République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri de Tizi-Ouzou
Faculté de Génie Electrique et de l'Informatique
Département informatique

Mémoire de fin d'étude de master

Domaine: Mathématique et informatique

Filière: Informatique

Spécialité : Conduite de projets informatiques

Thème

IMPLÉMENTATION D'UNE APPROCHE D'APPRENTISSAGE AUTOMATIQUE POUR LA PRÉDICTION DU SCORE À PRIORI DE DOCUMENT

Présenté par :

Mr : KHELOUI Amayas
Mr : KHERIS Lounes

Devant les jurys composés de :

Président : Mr CHEBOUBA Lokmane

Examinateur : Mr SAIDANI Fayçal

Proposé et encadré par :

Mr HAMMACHE Arezki

Promotion: 2019/2020

Remerciements

Tout d'abord on remercie « DIEU » pour nous avoir donné la force, capacité, volonté et courage afin de mener à bien et à terme ce travail.

Nous adressons également nos sincères remerciements à :

Notre promoteur Mr HAMMACHE Arezki pour son encadrement bien veillant, ses conseils, son suivi et ses remarques constructives tout au long de la rédaction et de l'élaboration de ce mémoire.

Aux membres du jurys pour avoir accepter de bien vouloir lire, examiner, évaluer et corriger notre travail.

Aux parents, familles et amis qui nous ont toujours été là pour nous, nous avoir soutenues et encouragés tout au long de nos années d'études.

Sommaire

Chapitre 2 : La pertinence à priori de document

Chapitre3 : Implémentation en mise en œuvre de l'approche

Liste des tableaux

Liste des Figures :

Introduction générale

Introduction générale

La recherche d'information est un domaine historiquement lié aux sciences de l'information et à la bibliothéconomie. Elle traite l'information dans la manière de l'organiser et de la façon de la sélectionner, elle peut être définie comme une activité qui dans le but de répondre à une question vise à localiser et à traiter une ou plusieurs informations au sein d'un environnement documentaire complexe.

Le traitement de cet environnement ne peut être effectué manuellement et donc l'objectif de la recherche d'information est d'extraire les informations pertinentes vis-à-vis d'une requête pour un utilisateur donné à travers l'utilisation d'un ensemble de programmes informatiques appelés systèmes de recherche d'information.

Un SRI peut guider l'utilisateur vers une bonne formulation de ses besoins. La solution proposée dans le but de réduire la distance entre la pertinence système et la pertinence utilisateur est la combinaison du score initial de la probabilité de pertinence document/requête avec le score à priori du document

Dans ce mémoire on s'intéresse particulièrement à l'extension d'un modèle de recherche en utilisant le score à priori de document, pour cela il faudra calculer les caractéristiques pris en compte pour le calcul de pertinence à priori de documents, puis combiner ce score avec le score de vraisemblance Document/Requête, pour enfin présenter les résultats obtenus par notre approche.

Pour atteindre cet objectif, nous avons structuré ce mémoire en trois chapitres :

Le premier chapitre intitulé « La recherche d'information » présente globalement la discipline de la recherche d'information.

Le deuxième chapitre intitulé « La pertinence à priori de documents » présente les caractéristiques prises en compte pour effectuer un classement à priori de documents ainsi que la méthode de combinaison de scores (à priori de document et score de vraisemblance Document/Requête).

Le dernier chapitre est consacré à la présentation de notre approche d'extension d'un modèle de recherche d'information en utilisant le score à priori de documents, ainsi que quelques résultats expérimentaux obtenus.

Chapitre 1 : La recherche d’information

I.1 Introduction

La Recherche d'Information (RI) peut être définie comme une activité dont la finalité est de localiser et de délivrer un ensemble de documents à un utilisateur en fonction de son besoin en informations. Le défit est de pouvoir, parmi le volume important de documents disponibles, trouver ceux qui correspondent au mieux à l'attente de l'utilisateur. L'opérationnalisation de la RI est réalisée par des outils informatiques appelés Systèmes de Recherche d'Information (SRI), ces systèmes ont pour but de mettre en correspondance une représentation du besoin de l'utilisateur (requête) avec une représentation du contenu des documents (fiche ou enregistrement) au moyen d'une fonction de comparaison (ou de correspondance). L'essor du web a remis la RI face à de nouveaux défis d'accès à l'information, il s'agit cette fois de retrouver une information pertinente dans un espace diversifié et de taille considérable.

Ce chapitre a pour but de présenter le domaine de la RI. Dans la première partie, 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 ; ainsi que, les modèles de RI. Dans la seconde partie est présentée l'évaluation des processus de systèmes de recherche d'information.

I.2 La recherche d'information

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. L'informatique a permis le développement d'outils pour traiter l'information et établir la représentation des documents au moment de leur indexation, ainsi que pour rechercher l'information. On peut aujourd'hui dire que la recherche d'information est un champ transdisciplinaire qui peut être étudié par plusieurs disciplines utilisant des approches qui devraient permettre de trouver des solutions pour améliorer son efficacité.

I.2.1 Définitions

- *Définition 1* : La recherche d'information est une activité dont la finalité est de localiser et de délivrer des granules documentaires à un utilisateur en fonction de son besoin en informations **[1].**

- *Définition 2* : 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 **[2].**
- *Définition 3* : La recherche d'information est une discipline de recherche qui intègre des modèles et des techniques dont le but est de faciliter l'accès à l'information pertinente pour un utilisateur ayant un besoin en information **[3].**

Toutes ces définition partagent l'idée que la RI à pour objet d'extraire d'un document ou d'un ensemble de documents les informations pertinentes qui reflètent un besoin d'information.

I.2.2 Concepts de base de la RI

Le rôle d'un Système de Recherche d'Information (SRI) est de mettre en œuvre des techniques et des moyens permettant de retourner les documents pertinents d'une collection en réponse à un besoin en information d'un utilisateur, exprimée par un langage de requêtes qui peut être le langage naturel, une liste de mots clés ou un langage booléen .
Afin d'atteindre cet objectif, un processus d'indexation des documents de la collection est effectué. Il permet de construire une représentation synthétique des documents, appelée 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. Puis, le système établit une correspondance entre la représentation de la requête et la représentation des documents (index) pour sélectionner et présenter les documents qui répondent le mieux au besoin de l'utilisateur (les documents pertinents). Le SRI s'appuie sur des modèles de RI pour établir cette correspondance entre les documents et la requête. L'architecture générale d'un SRI illustrée par la Figure I.1 fait ressortir des éléments constitutifs tels que : le document, le besoin en information, la requête et la pertinence, ainsi que trois principales fonctionnalités : l'indexation, la recherche et la reformulation de la requête. Dans ce qui suit, nous détaillons ces éléments et ces processus.

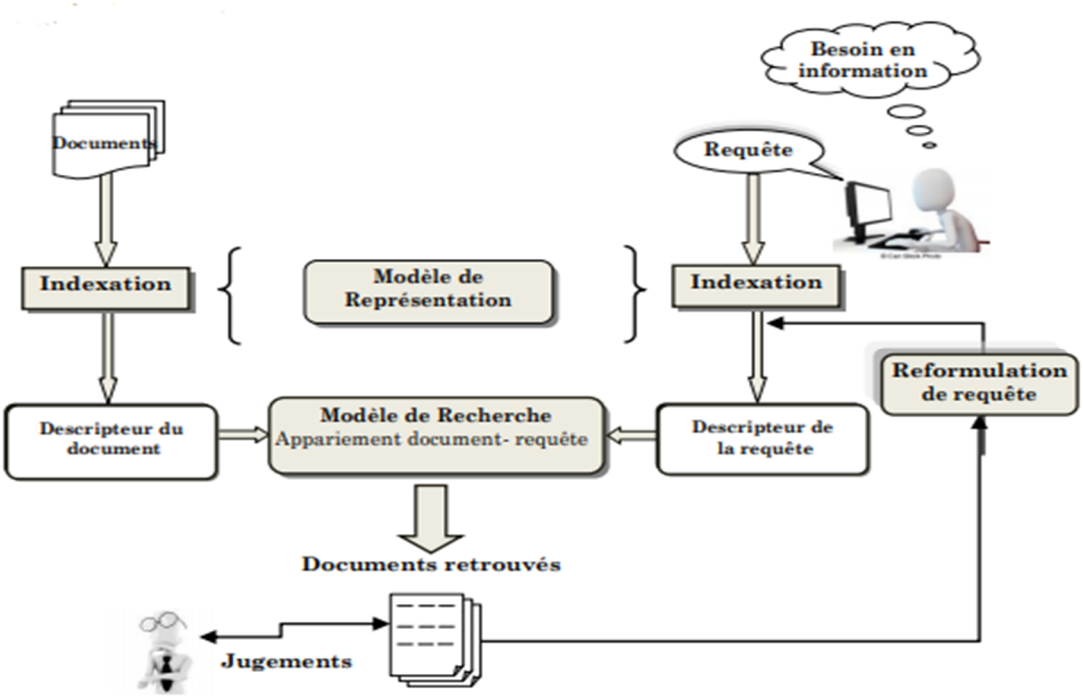

Figure I-1 : Architecture générale d'un SRI.

Collection de documents : la collection de documents (ou fond documentaire) constitue l'ensemble des informations exploitables et accessibles. 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 **[4].**

Document : Un document est un élément essentiel dans un SRI. Dans son acceptation courante, l'une des définitions possibles du terme document est de le considérer comme un support physique de l'information, qui peut être du texte, une page web, une image, une séquence vidéo, etc. Dans le cas d'un document texte on peut le représenter selon trois vues :

- **La vue sémantique (ou contenu)** : elle se concentre sur l'information véhiculée dans le document.
- **La vue logique** : elle définit la structure logique du document (structuration en chapitres, sections).
- **La vue présentation** : elle consiste en la présentation sur un médium à deux dimensions (alignement de paragraphes, indentation, en-têtes et pieds de pages, etc.). **[5]**

Besoin d'informations : la notion de besoin en information en recherche d'informations est souvent assimilée au besoin de l'utilisateur. Trois types de besoin utilisateur ont été définis :

- **Besoin vérificatif** : l'utilisateur cherche à vérifier le texte avec les données connues qu'il possède déjà. Il recherche donc une donnée particulière, et sait même souvent comment y accéder. La recherche d'un article sur Internet à partir d'une adresse connue serait un exemple d'un tel besoin. Un autre exemple serait de chercher la date de publication d'un ouvrage dont la référence est connue. Un besoin de type vérificatif est dit stable, c'est-à dire qu'il ne change pas au cours de la recherche.
- **Besoin thématique connu** : l'utilisateur cherche à clarifier, à revoir ou à trouver de nouvelles informations dans un sujet et un domaine connu. Un besoin de ce type peut être stable ou variable ; il est très possible en effet que le besoin de l'utilisateur s'affine au cours de la recherche. Le besoin peut aussi s'exprimer de façon incomplète, c'est-à-dire que l'utilisateur n'énonce pas nécessairement tout ce qu'il sait dans sa requête mais seulement un sous-ensemble. C'est ce qu'on appelle dans la littérature le label.
- **Besoin thématique inconnu** : cette fois, l'utilisateur cherche de nouveaux concepts ou de nouvelles relations en dehors des sujets ou des domaines qui lui sont familiers. Le besoin est intrinsèquement variable et est toujours exprimé de façon incomplète.

Requête : la requête constitue l'expression du besoin en information de l'utilisateur. Elle représente l'interface entre le SRI et l'utilisateur. Divers types de langages d'interrogation sont proposés dans la littérature. Une requête est un ensemble de mots clés, mais elle peut être exprimée en langage naturel, booléen ou graphique **[4].**

Pertinence : La pertinence est une notion fondamentale et cruciale dans le domaine de la RI. Cependant, la définition de cette notion complexe n'est pas simple, car elle fait intervenir plusieurs notions. Basiquement, elle peut être définie comme la correspondance entre un document et une requête ou encore comme une mesure d'informativité du document à la requête. Essentiellement, deux types de pertinence sont définis : la pertinence système et la pertinence utilisateur **[5].**

- **La pertinence Système** est souvent présentée par un score attribué par le SRI afin dévaluer l'adéquation du contenu des documents vis-à-vis de celui de la requête. Ce type de pertinence est objectif et déterministe.

- **Pertinence utilisateur** quant à elle, se traduit par les jugements de pertinence de l'utilisateur sur les documents fournis par le SRI en réponse à une requête. La pertinence utilisateur est subjective, car pour un même document retourné en réponse à une même requête, il peut être jugé différemment par deux utilisateurs distincts (qui ont des centres d'intérêt différents). De plus, cette pertinence est évolutive, un document jugé non pertinent à l'instant t pour une requête peut être jugé pertinent à l'instant t+1, car la connaissance de l'utilisateur sur le sujet a évolué **[5].**

I.3 Processus d'indexation

Pour que la recherche d'information se réalise avec des coûts acceptables, il convient d'effectuer une opération fondamentale sur les documents de la collection. Cette opération est nommée indexation **[6][7]** Elle consiste à associer à chaque document une liste de mots clés appelée aussi descripteur, susceptible de représenter au mieux le contenu sémantique des documents. La finalité de l'indexation est donc de produire une représentation synthétique des documents, formé de termes, ces termes peuvent être extraits de trois manières :

- ***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 **[8]**.Néanmoins, cette indexation présente un certain nombre d'inconvénients 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 différents moments.
- ***Semi-automatique*** : la tâche d'indexation est réalisée ici conjointement par un programme informatique et un spécialiste du domaine **[9]**. Le choix final des descripteurs revient à l'indexeur humain. Dans ce type d'indexation un langage d'indexation contrôlé est généralement utilisé.
- ***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.

Bien que l'indexation se base sur des techniques relativement établies, il peut y avoir plusieurs indexations différentes d'un même texte, aussi valables les unes que les autres, en fonction de l'usage qui doit en être fait et du public auquel elles s'adressent.

Les différentes étapes de l'indexation sont schématisées comme suit :

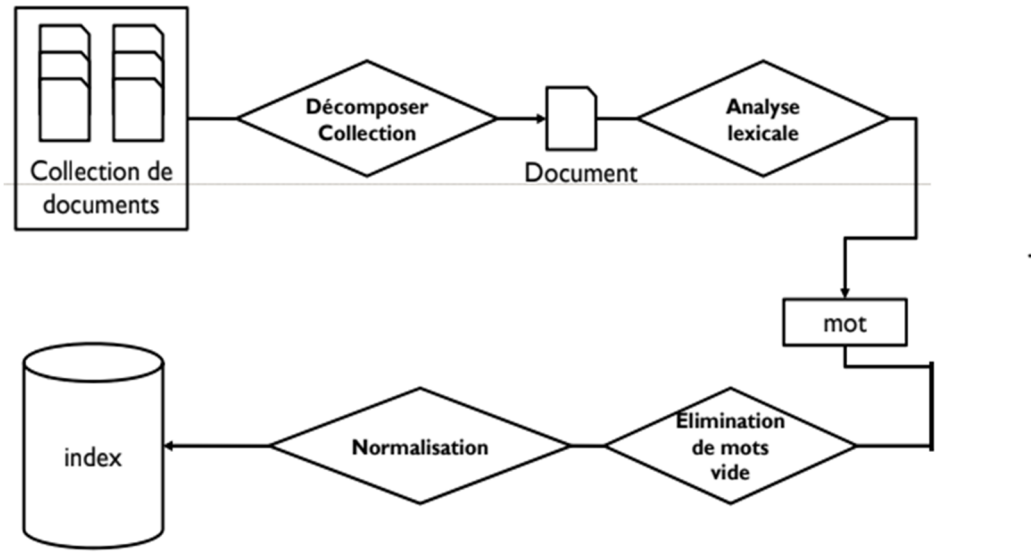

Figure I-2 : Indexation d'un document.

I.3.1 L'analyse lexicale

Elle permet de convertir un texte de document en une liste de termes. Un terme est un groupe de caractères constituant un mot significatif **[10]**. L'analyse lexicale permet de reconnaître les espaces de séparation des mots, les chiffres, les ponctuations, etc.

I.3.2 L'élimination des mots vides

Les mots vides (article, proposition, conjonction, etc.) sont des mots non significatifs dans un document, 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 ou stop-list),
- L'élimination des mots ayant une fréquence qui dépasse un certain seuil dans la collection.

L'élimination des mots vides réduit la taille de l'index, ce qui améliore le temps de réponse du système. Cependant, elle peut réduire le taux de rappel, en réponse à des requêtes bien spécifiques (par exemple, la requête be or not be).

I.3.3 La normalisation

La normalisation consiste à représenter les différentes 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 utilisées : la table de correspondance, l'élimination des affixes (l'algorithme de Porter), la troncature, l'utilisation des N-grammes **[11].**

L'inconvénient majeur de cette opération est qu'elle supprime dans certains cas la sémantique des termes originaux, c'est le cas par exemple des termes derivate/derive, activate/active, normalisés par l'algorithme de Porter.

I.3.4 Le choix de descripteurs

Elle consiste à déterminer le type d'unités élémentaires pour représenter les documents. On parle aussi de descripteur. L'objectif est d'avoir une représentation des documents permettant une moindre perte d'information sémantique possible. On distingue plusieurs types de descripteurs **[12].**

- **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 de bi-grammes et trigrammes dans la recherche d'information.
- **Les mots composés** : groupes de mots ou expression (phrase en anglais) sont souvent plus riches sémantiquement que les mots qui les composent pris séparément. Par exemple, le mot composé "imprimante laser" est plus précis que "imprimante" et "laser" pris isolément. Cet argument a conduit à leur large utilisation en RI.
- **Les concepts** : qui sont des expressions pris généralement d'une structure conceptuelle, tels que les thésaurus ou les ontologies.

I.3.5 La création de l'index

Au terme du processus d'indexation, un ensemble de structure de données sont crées. Ces dernières permettent un accès efficace à la représentation des documents. Le fichier inverse est la structure de données la plus utilisée **[6][13]** il enregistre pour chaque descripteur les identificateurs des documents qui le contiennent et sa fréquence dans chacun de ces documents.

Généralement, les structures de données sont compressées avant d'être enregistrées sur le disque, ce qui permet de réduire la taille de l'index. Parmi les méthodes de compression utilisées on peut citer la méthode Elias Gamma **[14]** qui opère au niveau bit requérant ainsi beaucoup d'opérations pour la compression et la décompression **[15].**

I.4 Appariement document requête

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 le SRI compare les deux représentations. Le résultat de cette comparaison se 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. Cette fonction d'appariement est notée *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.5 Les modèles de recherche d'information

L'étape d'appariement, décrite précédemment, repose sur des modèles de RI dont l'objectif est de fournir une formalisation du processus de RI et un cadre théorique pour la modélisation de la mesure de pertinence. Il existe un grand nombre de modèles de RI textuelle développé dans la littérature. Ils reposent sur l'utilisation de la logique, l'algèbre, la théorie de la probabilité et les statistiques. Comme le montre la figure I.3, on peut distinguer trois grandes classes de modèles, regroupés selon les fondements mathématiques sur lesquels ils se basent **[16].**

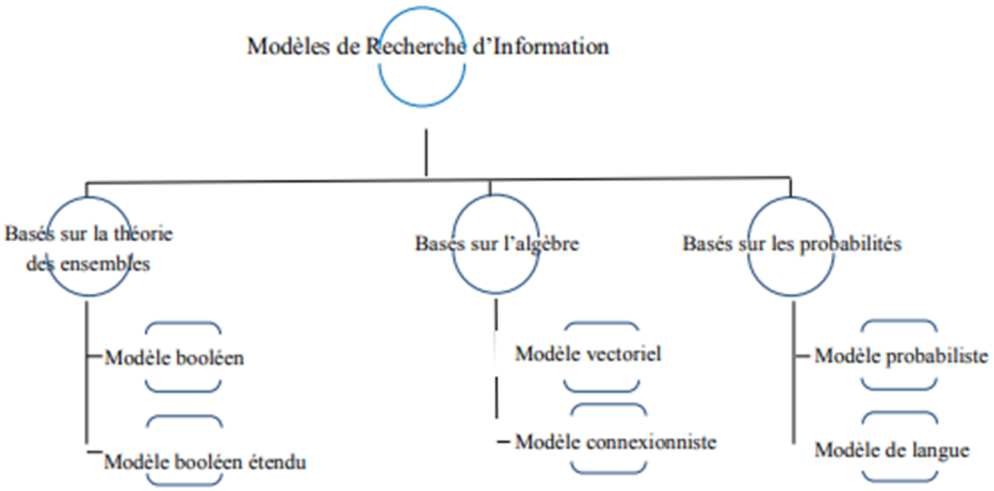

Figure I-3 : Taxonomie des modèles de RI [Baeza-Yates et al.1999].

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 = {ti}, i ∈ {1, ..., n} est constitué de n mots ou racines de mots qui apparaissent dans les documents. Le 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

Le modèle booléen est basé sur la théorie des ensembles. Dans ce modèle, les documents et les requêtes sont représentés par des ensembles de mots clés. Chaque document est représenté par une conjonction logique des termes non pondérés qui constitue l'index du document. Un exemple de représentation d'un document est comme suit : $d = t1 \wedge t2 \wedge t3... \wedge t_n$.

Une requête est une expression booléenne dont les termes sont reliés par des opérateurs logiques (OR, AND, NOT) permettant d'effectuer des opérations d'union, d'intersection et de différence entre les ensembles de résultats associés à chaque terme. Un exemple de représentation d'une requête est comme suit : $q = (t1 \wedge t2) \vee (t3 \wedge t4)$. La fonction de

correspondance est basée sur l'hypothèse de présence/absence des termes de la requête dans le document et vérifie si l'index de chaque document dj implique l'expression logique de la requête q. Le résultat de cette fonction est donc binaire est décrit comme suit : $RSV\ (q, d) = \{0,1\}$

I.5.2 Le modèle vectoriel

Le modèle vectoriel de base a été introduit par Salton **[17]** concrétisé dans le cadre du système SMART. Ce modèle se base sur une formalisation géométrique. En effet, les documents et les requêtes sont représentés dans un même espace, défini par un ensemble de dimensions, chaque dimension représente un terme d'indexation. Les requêtes et les documents sont alors représentés par des vecteurs, dont les composantes représentent le poids du terme d'indexation considéré dans le document (la requête). Formellement, si on a un espace T de termes d'indexation de dimension n, $T = \{t_1,t_2,\ldots,t_j,\ldots,t_n\}$. Un document d_i est représenté par un vecteur $d_i\ (w_{i1},w_{i2},\ldots,w_{ij},\ldots,w_{in})$. Une requête q par un vecteur $q(w_{q1},w_{q2},\ldots,w_{qj},\ldots,w_{qn})$.Où $W_{ij}(resp.W_{qj})$ représente le poids du terme t_j dans le document d_i (respectivement dans la requête q).

Le modèle vectoriel offre des moyens pour la prise en compte du poids de terme dans le document. Dans la littérature, plusieurs schémas de pondération ont été proposés. La majorité de ces schémas prennent en compte la pondération locale et la pondération globale. La pondération locale permet de mesurer l'importance du terme dans le document. Elle prend en compte les informations locales du terme qui ne dépendent que du document. Elle correspond en général à une fonction de la fréquence d'occurrence du terme dans le document (noté *tf* pour term frequency), exprimée ainsi :

$$Tf_{ij} = 1 + log(f(t_i,d_j)) \qquad \text{(I.1)}$$

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 les 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 (i.e. peu utile pour la discrimination). Un facteur de pondération globale est alors introduit. Ce facteur nommé *idf* (inverted document frequency), dépend d'une manière inverse de la fréquence en document du terme et exprimé comme suit :

$$Idf = log(N/n_i) \qquad \text{(I.2)}$$

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

Les fonctions de pondération combinant la pondération locale et globale sont référencées sous le nom de la mesure *tf* × *idf*. Cette mesure donne une bonne approximation de l'importance du terme dans les collections de documents de taille homogène. Cependant, un facteur important est ignoré, la taille du document. En effet, la mesure (*tf* × *idf*) ainsi définie favorise les documents longs, car ils ont tendance à répéter le même terme, ce qui accroit leur fréquence, par conséquent augmentent 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 pondération, comme facteur de normalisation **[18] [19].**

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**[20]** dont les plus courantes sont décrites dans **le tableau 1** ci-dessous.

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	$RSV(q,d_i)=\frac{\sum_{j=1}^{\lvert T\rvert} w_{qj}.w_{ij}}{\sum_{j=1}^{\lvert T\rvert} {w_{qj}}^2+\sum_{j=1}^{\lvert T\rvert} {w_{ij}}^2-\sum_{j=1}^{\lvert T\rvert} w_{qj}.w_{ij}}$

Tableau I-1 :Les mesures de similarités utilisées dans le modèle vectoriel

Le modèle vectoriel caractérisé par sa prise en compte du poids des termes dans les documents, permet de retrouver des documents qui répondent partiellement à une requête. De plus, ce modèle offre un moyen facile pour classer les résultats d'une recherche, qui est basée sur la similarité potentielle entre documents et requête. L'inconvénient majeur de modèle vectoriel est qu'il repose sur l'hypothèse de l'indépendance des termes d'indexation, or ces termes dans les documents sont souvent sémantiquement liés. Plusieurs variantes du modèle

vectoriel ont été proposées, pour remédier à cette limitation, c'est-à-dire prendre en compte la dépendance entre termes d'indexation. Parmi elles, on trouve, le modèle vectoriel généralisé, le modèle LSI (Latent Semantic Indexing) et le modèle connexionniste **[5].**

I.5.3 Les modèles probabilistes

I.5.3.1 Le modèle probabiliste de base

Le modèle probabiliste est fondé sur la théorie des probabilités. Il trie les documents selon leur probabilité de pertinence vis-à-vis d'une requête. La fonction de classement (tri) de ce modèle est exprimée ainsi :

$$RSV(q,d)=\frac{P(Per|q,di)}{P(NPer|q,di)} \quad (I.3)$$

L'idée de base de cette fonction est de sélectionner les documents ayant à la fois une forte probabilité d'être pertinents et une faible probabilité d'être non pertinents à la requête.

Où $P(Per\ |q,d_i)$ et $P(NPer\ |q,d_i)$: la probabilité qu'un document d soit pertinent (*Per*) vis-à-vis de la requête q) respectivement non pertinent *(NPer)).*

En appliquant la formule de bayes pour les deux probabilités on obtient :

$$P(Per|q,d_i)= \frac{P(Per|q).\ P(di|Per,q)}{P(d_i)} \quad (I.4)$$

$$P(NPer|q,d_i)=\frac{P(NPer|q).\ P(di|NPer,q)}{P(d_i)} \quad (I.5)$$

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 ;

$P(d_i|NPer, q)$ indique la probabilité que d_i fait partie des documents non pertinents pour la requête *q* ;

P(Per|q) et P(NPer|q) indiquent respectivement la probabilité de pertinence et de non-pertinence d'un document quelconque (avec *(Per|q) + P(NPer|q) = 1)* qui sont fixes. Après remplacement dans la fonction de tri, on aura la formule suivante :

Si on suppose que les termes d'indexation sont indépendants, alors on peut estimer les deux probabilités ainsi :

$$P(d_i|Per,q) = \prod_{tj\in di} P(\boldsymbol{tj}|Per,q) \times \prod_{tj\notin di} 1 - P(\boldsymbol{tj}|Per,q) \quad (I.6)$$

$$P(d_i|NPer,q) = \prod_{tj\in di} P(\boldsymbol{tj}|NPer,q) \times \prod_{tj\notin di} 1 - P(\boldsymbol{tj}|NPer,q) \quad (I.7)$$

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

En posant $P_i = P(\boldsymbol{t_j}|Per,q)$, $Q_i = P(\boldsymbol{t_j}|NPer,q)$ et $P_i = q_i$ pour les termes qui 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 :

$$RSV(d_i,q) = \sum_{ti\in q} log\left[\frac{Pi(1-qi)}{qi(1-Pi)}\right] \quad (I.8)$$

Afin de classer les documents avec cette formule, il faut estimer les valeurs des deux probabilités. En l'absence de collection (documents) 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).

I.5.3.2 Le modèle de langue

Les modèles statistiques de langue sont exploités avec beaucoup de succès dans divers domaines : la reconnaissance de la parole **[21]**, la traduction automatique **[22][23]**, la recherche d'information **.**etc.

L'utilisation des modèles de langue en RI remonte à 1998. Le principe de ce modèle consiste à construire un modèle de langue pour chaque document, soit M_d , puis de calculer la probabilité qu'une requête q puisse être générée par le modèle de langue du document, soit $P(q|M_d)$.

Le modèle de langue utilisé est souvent le modèle uni-gramme, la probabilité $P(q|M_d)$ est alors exprimée ainsi :

$$P(q|M_d) = \prod_{t\in q} P(t|Md) \quad (I.9)$$

$P(t|M_d)$ peut être estimé en se basant sur l'estimation maximale de vraisemblance (maximum likelihood estimation). Elle est donnée par :

$$P(t,Md) = \frac{tf(t,d)}{|d|} \quad (I.10)$$

Où $\boldsymbol{tf(t,d)}$ est la fréquence du terme $\boldsymbol{t_i}$ dans le document d.

Pour remédier au problème posé par les mots de la requête absents dans le document, qui ont pour effet d'avoir la probabilité $P(q|M_d)$ nulle ; des techniques de lissage (smoothing) sont utilisées, dont le lissage de Laplace (ajouter-un), le lissage de Good-Turing, le lissage Backoff, le lissage par interpolation, etc. Leur principe consiste à assigner des probabilités non nulles aux termes, qui n'apparaissent pas dans les documents.

I.6 La reformulation de requêtes

Dans les SRI, la requête initiale seule est souvent insuffisante pour permettre la sélection de documents répondant au besoin de l'utilisateur. De ce fait, plusieurs techniques ont été proposées pour améliorer les performances des SRI. Ces méthodes apportent des solutions aux deux principales questions :

1. Comment peut-on retrouver plus de documents pertinents vis-à-vis d'une requête donnée?

2. Comment peut-on mieux exprimer la requête de l'utilisateur de manière à mieux répondre à son besoin?

La figure (I.4), présente les principales techniques d'amélioration des SRI par reformulation de la requête initiale en y ajoutant de nouveaux termes. La reformulation peut se faire par expansion automatique de la requête, par combinaison de différentes présentations de la requête ou par réinjection de pertinence. Nous présentons dans ce qui suit ces trois principales techniques **[4].**

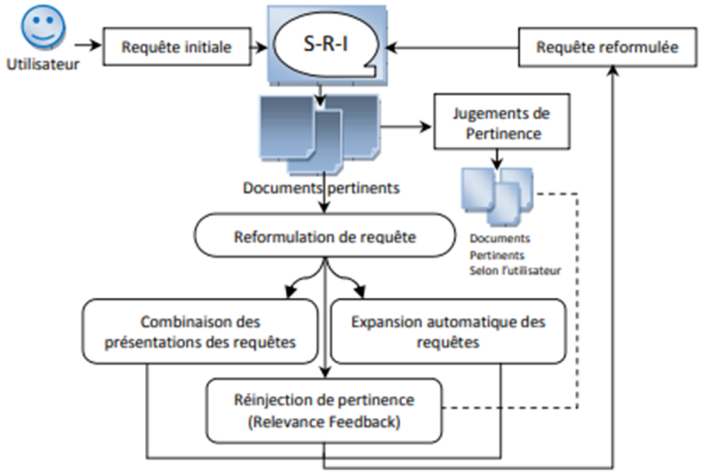

Figure I-4 : Techniques d'améliorations des SRI par reformulation de requete.

I.6.1 Expansion automatique des requêtes

L'expansion directe de la requête consiste à rajouter à la requête initiale des termes issus de ressources linguistiques existantes ou bien de ressources construites à partir des collections. Plus précisément, un niveau des ressources linguistiques, le but est d'utiliser un vocabulaire contrôlé issu de ressources externes. On peut alors utiliser des ontologies linguistiques (citons par exemple Wordnet). On peut également ajouter à la requête des variantes morphologiques des termes employés par l'utilisateur. Le but de ce mécanisme est d'assurer la restitution des documents indexés par des variantes des termes composant la requête. Les associations établies manuellement traduisent généralement des relations de synonymie et de hiérarchie. Les thésaurus construits manuellement sont un moyen efficace pour l'expansion de requête. Cependant, leur construction et la maintenance des informations sémantiques qu'ils contiennent sont coûteuses en temps et nécessitent le recours à des experts des domaines considérés. Pour cette raison, ils restent peu utilisés par les SRI. En ce qui concerne la seconde catégorie de ressources, elles sont construites en s'appuyant sur une analyse statistique des collections. Il s'agit de chercher des associations de termes afin d'ajouter des termes voisins à la requête. Il existe aussi d'autres méthodes entièrement automatiques telles que le calcul des liens contextuels entre termes et la classification automatique de documents. Les associations créées automatiquement sont généralement basées sur la cooccurrence des termes dans les documents. Les liens inter-termes renforcent la notion de pertinence des documents par rapport aux requêtes **[4].**

I.6.2 Combinaison des présentations des requêtes

Plusieurs approches de RI utilisent une seule représentation de requête comparée à plusieurs représentations de document (algorithmes multiples de recherche). Il a été montré dans **[24]**qu'une recherche plus efficace peut être atteinte en exploitant des représentations multiples de requêtes ou des algorithmes de recherche différents ou encore en utilisant différentes techniques de réinjection. Une combinaison des représentations de requêtes peut augmenter le rappel d'une requête, tandis que la combinaison des algorithmes de recherche peut augmenter la précision. La base théorique de la combinaison des évidences a été présentée par Ingwersen **[25].** Il a en particulier montré que des représentations multiples d'un même objet, par exemple une requête, permettent une meilleure perception de l'objet qu'une seule bonne représentation. Cependant, il est important que chacune des sources d'évidences utilisées fournisse non seulement un point de vue différent sur l'objet, mais que ces points de vue aient différentes bases cognitives. Les représentations multiples d'une requête peuvent

donner différentes interprétations du besoin en information. Une des approches de combinaison de multiples représentations de requêtes est proposée dans **[26].** Elle consiste à calculer les scores des documents directement depuis la fonction d'appariement document-requête en utilisant le même système de recherche mais différentes versions de la requête. Ensuite, les résultats obtenus par chacune des versions sont combinés pour avoir une seule liste finale. Ces versions sont issues soit des expressions d'une même requête par des chercheurs différents, soit des présentations d'une même requête dans des langages différents. Tamine et al, proposent dans **[27]** une technique de recherche d'information basée sur les algorithmes génétiques, plus précisément, elle propose d'utiliser une population de requêtes qui évolue à chaque étape de la recherche et tente de récupérer le maximum de documents pertinents **[4].**

I.6.3 Réinjection de pertinence

Ces méthodes impliquent que l'utilisateur doit sélectionner les documents qu'il considère pertinents à partir des résultats issus de sa requête initiale. Ce jugement de pertinence de l'utilisateur est ensuite exploité pour reformuler la requête initiale en modifiant le poids des termes qu'elle contient et/ou en ajoutant de nouveaux termes considérés utiles pour retrouver des documents pertinents. La technique de réinjection de pertinence a été mise en place à l'origine dans le modèle vectoriel. Rocchio**[28]** a proposé le modèle de reformulation de requête suivant **:**

$$Q_N = \alpha.Q_0 + \beta.\frac{1}{|R|}\sum_{r\epsilon R} r - \frac{1}{|\acute{R}|}\sum_{r\acute{\epsilon}\acute{R}} \acute{r} \qquad \text{(I.11)}$$

Q_N est le vecteur de la nouvelle requête (reformulée);

Q_0 est le vecteur de la requête originale;

R est l'ensemble des vecteurs *r* des documents jugés pertinents par l'utilisateur;

$\acute{\boldsymbol{R}}$ est l'ensemble des vecteurs $\acute{r}$ des documents jugés non pertinents par l'utilisateur;

$\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}$ sont les paramètres de la reformulation. On peut 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, comme suit :

$$W_{qj} = log\left[\frac{r+0.5/(\acute{r}-r+\acute{0}.5)}{(dfj-\acute{r}+0.5)/(n-dfj-r+r+\acute{0}.5)}\right] \quad \text{(I.12)}$$

Où : $\boldsymbol{r}$ représente le nombre de documents pertinents ;

$\boldsymbol{\acute{r}}$ est le nombre de documents pertinents contenant le terme q_j;

$\boldsymbol{df_j}$ est le nombre de documents contenant le terme q_j;

$\boldsymbol{n}$ est le nombre total de documents dans la collection.

I.7 Evaluation des performances d'un système d'information

L'évaluation d'un SRI constitue une étape importante dans l'élaboration d'un modèle de RI, puisqu'elle permet de paramétrer le modèle, et de fournir des éléments de comparaison entre modèles. L'évaluation nécessite alors, la définition d'un ensemble de mesures et méthodes d'évaluation et collections de test, assurant l'objectivité de l'évaluation. L'évaluation d'un SRI, peut être appréhendée selon deux aspects : un aspect efficience et un aspect efficacité. L'aspect efficience dépend de l'évaluation cognitive de l'utilisateur, tels que la facilité d'utilisation du système, rapidité d'accès, temps de réponse à une requête, présentation des résultats, etc. L'aspect efficacité concerne la capacité du système à sélectionner le maximum de documents pertinents et un minimum de documents non pertinents**].**

I.7.1 Les Collections de test

Une collection (ou corpus) de test constitue le moyen d'évaluation des SRI. Elle est généralement composée d'un ensemble de documents, d'un ensemble de requêtes et des jugements de pertinence associés à ces requêtes. L'évaluation d'un SRI consiste à comparer les résultats retournés par ce dernier par rapport aux jugements de pertinence. Des mesures d'évaluation, décrites dans la section suivante, sont utilisées pour effectuer cette comparaison. Les collections de test sont le résultat de projets d'évaluation qui se sont multipliés depuis les années 1970, on peut citer la collection CACM1, la collection CISI2, la campagne CLEF3 et la campagne TREC4.

La campagne TREC constitue à ce jour la campagne de référence dans le cadre de l'évaluation des systèmes de recherche d'information et cela depuis son lancement en 1992 **[29][30]** L'objectif de cette campagne est de proposer une plate-forme qui réunit des collections de test, des tâches spécifiques et des protocoles d'évaluation pour chaque tâche afin de mesurer les différentes stratégies de recherche**[31].**

Les tâches proposées se sont diversifiées d'une campagne à une autre (à raison d'une par an) ; parmi les tâches proposées dans TREC 2012 on peut citer : recherche d'information sur le

web, recherche d'information médical, recherche d'information dans les micros blogs, recherche d'information contextuelle, etc. La taille des collections augmente au fil des années, passant de 2 Go dans TREC 1 à 25 To dans TREC 2011. Chaque collection est composée d'un certain nombre de documents, allant de quelques milliers à plusieurs millions. Les documents sont codés à l'aide de SGML dans un format spécifique TREC. La Figure I.5 illustre un exemple d'un document TREC.

```
<DOC>
<DOCNO> AP880212-0004 </DOCNO>
<FILEID>AP-NR-02-12-88 1637EST</FILEID>
<FIRST>r w AM-PeanutSupports     02-12 0155</FIRST>
<SECOND>AM-Peanut Supports,150</SECOND>
<HEAD>Peanut Price Supports Will Go Higher This Year</HEAD>
<DATELINE>WASHINGTON (AP) </DATELINE>
<TEXT>
   Price supports for peanuts grown under 1988
quotas will be $615.27 per ton, an increase of $7.80 from last
year, the Agriculture Department said Friday.
   Deputy Secretary Peter C. Myers said the increase was required
by a formula in the law which takes rising production costs into
consideration.
   The annual quota is set at a level equal to the estimated
quantity of peanuts that will be needed for domestic edible uses,
seed and related purposes.
   Production of non-quota peanuts, which can be grown for peanut
oil and meal, and for export, will be supported at $149.75 per ton,
unchanged from last year, Myers said.
   In setting the support for non-quota peanuts, officials are
required to consider certain factors, including the demand for oil
and meal, the expected prices for other vegetable oils and meals,
and the foreign demand for peanuts.
</TEXT>
</DOC>
```

Figure I-5 : Exemple d'un document TREC.

Chaque collection TREC a généralement 50 à 100 requêtes correspondantes. Une requête TREC est structurée comme suit: un identifiant de requête unique TREC, un titre, une description plus détaillée du besoin en information et une rubrique qui explique dans quelles circonstances un document doit être jugé pertinent ou non pertinent pour une requête. Un exemple d'une requête TREC est montré dans la figure I.6.

```
<top>

<head> Tipster Topic Description
<num> Number:  066
<dom> Domain:  Science and Technology
<title> Natural Language Processing
<desc> Description:
Document will identify a type of natural language processing technology which
is being developed or marketed in the U.S.
<smry> Summary:
Document will identify a type of natural language processing technology which
is being developed or marketed in the U.S.
<narr> Narrative:
A relevant document will identify a company or institution developing or
marketing a natural language processing technology, identify the technology,
and identify one or more features of the company's product.
<con> Concept(s):
1.  natural language processing
2.  translation, language, dictionary, font
3.  software applications
<fac> Factor(s):
<nat>  Nationality:  U.S.
</fac>
<def> Definition(s):

</top>
```

Figure I-6 : Exemple d'une requête TREC.

I.7.2 Mesures d'évaluation du SRI

Le principal objectif d'un système de recherche d'information est de restituer à l'utilisateur tous les documents pertinents et de rejeter tous les documents non pertinents. Cet objectif est évalué à l'aide de différentes mesures d'évaluation **[32].** On présente ci-dessous les plus utilisées.

- **La précision** : est le rapport du nombre de documents pertinents restitués par le système (*SP*) sur le nombre total de documents restitués *(R)*, exprimée ainsi :

$$Précision = \frac{SP}{R} \quad \text{(I.13)}$$

- **Le rappel** : est le rapport du nombre de documents pertinents restitués (*SP*) sur le nombre total de documents pertinents *(P)*, exprimé ainsi :

$$\text{Rappel} = \frac{SP}{P} \quad \text{(I.14)}$$

- **Le bruit** : la mesure d'évaluation bruit est une notion complémentaire à la précision, elle est définie par $B = 1 - P$ où P est la précision du SRI.

- **Le silence** : la mesure d'évaluation silence est une notion complémentaire au rappel, elle est définie par *S = 1 - R* où *R* est le rappel du SRI.
- **La précision moyenne non interpolée (MAP) :**

  La précision moyenne non interpolée (Mean Average Precision) est calculée en deux étapes. D'abord on calcule la précision moyenne pour une requête donnée (AP_q) ainsi pour chaque document pertinent retrouvé on calcule sa précision ($pr(d_i)$) qui est égale au nombre de documents pertinents retrouvés sur le rang de ce document ; pour les documents retrouvés non pertinents leur précision est égale à zéro.

  La précision moyenne pour une requête donnée est alors obtenue en calculant la moyenne des précisions des documents pertinents, exprimée ainsi :

$$\boldsymbol{AP_q = \frac{1}{N} \sum_{i=1}^{N} (pr(d_i))} \quad \text{(I.15)}$$

  Avec

$$pr(d_i) = \begin{cases} \frac{r_{ni}}{n_i} & \text{Si } d_{ij} \text{ est retrouvé} \\ 0 & \text{Sinon} \end{cases}$$

  Où n_i dénote le rang du document d_i qui a été retrouvé et qui est pertinent pour la requête, r_{ni} le nombre de documents pertinents retrouvé au rang n_i et N est le nombre total de documents pertinents pour la requête q.

  Dans la seconde étape, on calcule la précision moyenne pour un ensemble de requêtes, en effectuant la moyenne des précisions moyennes de chaque requête, elle est exprimée ainsi :

$$\text{MAP} = \frac{1}{M} \sum_{j=1}^{M} \boldsymbol{AP_{q_j}} \quad \text{(I.16)}$$

  Où AP_{q_j} dénote la précision moyenne pour la requête « j » et M représente le nombre de requêtes considérées.

- **La précision à N document :** C'est la proportion des documents les plus pertinents retournés DPR au rang N, alors la précision est exprimée ainsi :

$$\mathbf{P@N} = \frac{\mathbf{DPR}}{\mathbf{N}} \quad \text{(I.17)}$$

- **Courbe de Rappel-Précision** : un système idéal devrait retourner tous les documents pertinents et que les documents pertinents ; c'est à dire un taux de précision et de rappel égal à 100%. Cette situation ne se produit pas dans un système réel car le taux de précision et de rappel sont antagonistes. En effet, Lorsque la précision augmente, le rappel diminue et inversement. Ainsi, pour mesurer les performances d'un système il faut utiliser les deux mesures conjointement. Cela est réalisé en calculant la paire des mesures (taux de rappel, taux de précision) à chaque document restitué. Nous considérons par exemple une requête pour laquelle il existe cinq (5) documents pertinents dans le corpus. Le Tableau I.2 illustre le calcul de la précision et de rappel pour les dix (10) premiers documents retournés par un SRI. La lettre (P) précise que le document est pertinent.

Rang document	Pertinence	Rappel	Précision
Doc 1	P	1/10	1
Doc 2	P	2/10	1
Doc 3		2/10	2/3
Doc 4		2/10	2/4
Doc 5	P	3/10	3/5
Doc 6	P	4/10	4/6
Doc 7		4/10	4/7
Doc8	P	5/10	5/8
Doc 9		5/10	5/9
Doc 10		5/10	5/10

Tableau I-2 :Exemple de calcul de rappel et de précision pour une requete.

La figure I.7 illustre la courbe de rappel et précision correspondante aux résultats du Tableau I.2.

Pour rendre la courbe lisible on ne garde que la précision calculée à chaque point de rappel (c'est à dire à chaque document pertinent restitué).

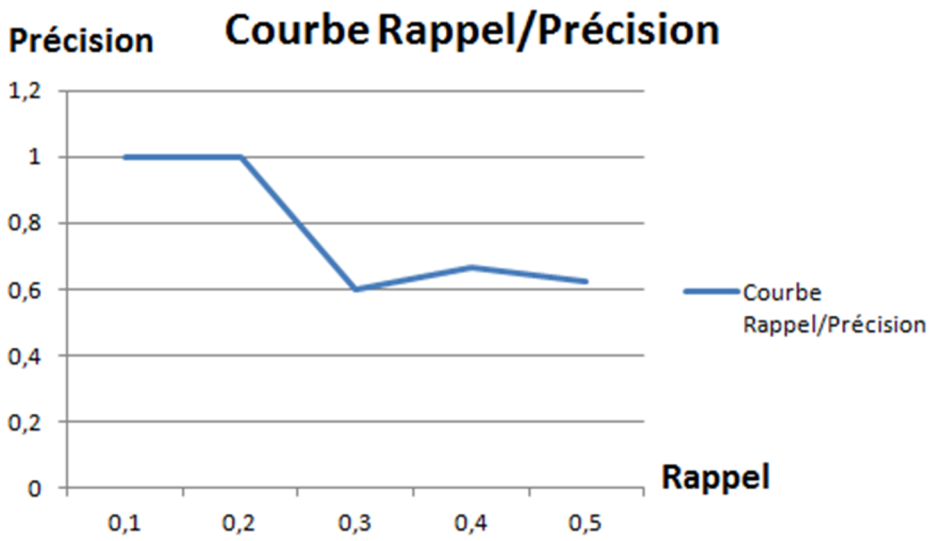

Figure I-7 : Courbe de Rappel et Précision.

La courbe ci-dessus permet d'évaluer les performances du système pour la requête considérée. Afin d'évaluer le système pour un ensemble de requêtes, on calcul la moyenne des précisions à chaque niveau de rappel. Comme les niveaux de rappel ne sont pas unifiés 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 sont calculées par une 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. Cette interpolation est encore discutable, mais présente un intérêt dans l'évaluation de systèmes de recherche d'information. Elle permet entre autre de construire des courbes décroissantes plus simple à comparer **[6].**

I.8 Conclusion

Dans ce chapitre nous avons présenté les principales notions et concepts de la recherche d'information. 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. Ensuite, nous avons étudié les différents modèles de la RI. Enfin, la reformulation de requête. Le chapitre suivant portera sur la pertinence à priori de document ainsi qu'une méthode de combinaison des scores.

Chapitre 2 : La probabilité de pertinence à priori de documents

II.1 Introduction

Les systèmes de recherche d'information exploitent dans leur majorité deux classes de sources d'évidence pour trier les documents répondant à une requête. La première, la plus exploitée, est dépendante de la requête, elle concerne toutes les caractéristiques relatives à la distribution des termes de la requête dans le document et dans la collection (tf-idf). La seconde classe concerne des facteurs indépendants de la requête, elle mesure une sorte de qualité ou d'importance a priori du document. Parmi ces facteurs, on en distingue la longueur du document, le type d'URL du document, la présence d'URL dans le document, ses auteurs, etc.

Dans ce chapitre, nous nous organiserons comme suit : dans un premier lieu nous allons définir la pertinence à priori des documents , définir les caractéristiques qui permettent de classer les documents selon leur pertinence à priori, présenter le modèle de langue pour enfin , présenter les différentes méthodes de combinaison de scores à priori et dépendant de la requête.

II.2 Définition de la pertinence à priori de documents

La pertinence à priori des documents est la probabilité qu'un document soit plus pertinent qu'un autre car ils différent dans certaines propriétés indépendantes de la requête, ces propriétés peuvent être la taille de document, le nombre de liens entrants, etc. Si la probabilité a priori de pertinence d'un document n'est pas conditionnée par l'une de ces propriétés alors cette probabilité représente la probabilité de prélever un document de la collection. Par conséquent, tous les documents dans la collection ont la même probabilité d'être sélectionnés, et donc la probabilité a priori de pertinence de document peut être ignorée lors du classement des documents. Par contre, si la probabilité a priori est conditionnée par l'une de ces caractéristiques alors les documents de la collection n'ont pas la même probabilité a priori, et donc les documents ne sont pas équiprobables. Par exemple, si la caractéristique utilisée est le score de popularité de document alors un document populaire est plus probable d'être pertinent qu'un document moins populaire.

II.3 Les caractéristiques utilisées pour le classement à priori des documents

Plusieurs caractéristiques ont étés utilisées pour estimer la probabilité a priori d'un document, on peut citer : la longueur du document **[33][34**],la structures des liens**[33][36]**, le facteur temps **[37][38]**et le rapport information/bruit**[39].**

L'intuition derrière l'utilisation de ces caractéristiques est que : un document est plus probable d'être pertinent car : il est plus long, il est plus populaire, il est plus récent, ou contient plus d'informations que de bruit. Nous présentons ci-dessous quelques travaux utilisant ces caractéristiques.

II.3.1 La structure de liens

L'idée derrière l'utilisation de la structure des liens est que les documents populaires ou les plus cités tendent à être plus pertinents. La méthode simple d'utilisation de la structure des liens est l'usage du nombre de liens entrants. La probabilité de pertinence a priori est alors exprimée comme suit :

$$P(d) = \frac{n(l,d)}{\sum_{di \in C} n(l,di)} \quad \text{(II.1)}$$

Où (l, d) est le nombre de liens entrants dans le document.

C : est la collection de documents.

D'autres facteurs plus sophistiqués on été utilisés comme : le Hits **[40]** , le PageRank **[41]** qui est à l'origine du moteur de recherche Google. Le principe de cet algorithme consiste à ordonner les pages web selon leur popularité, en se basant sur l'hypothèse suivante : « une page est populaire (importante) quand elle est beaucoup citée ou citée par une page très populaire». L'estimation de cette popularité est formalisée comme suit :

$$PR(p) = (1 - d) \times \frac{1}{T} + d \times \sum_{i=1}^{k} \frac{PR(pi)}{C(pi)} \quad \text{(II.2)}$$

Où :

PR(p) : est le PageRank de la page p.

T : est le nombre total de pages sur le web (indexées) ;

d : est un paramètre fixé à 0.85 ;

$C(p_i)$: est le nombre de liens sortant de la page pi , et k est le nombre de pages qui pointent la page p.

De nombreux travaux ont montré que l'incorporation du PageRank dans le classement des pages web, améliore les performances de recherche sur de grandes collections de type web **[42][43][44].**

II.3.2 La taille de document

L'intuition de l'utilisation d'une telle caractéristique est qu'un document plus long tend à contenir plus d'informations et par conséquent il est plus probable d'être pertinent. Les résultats obtenus avec l'utilisation de cette caractéristique ont été mixtes et cela selon la collection utilisée **[33].**

La probabilité de pertinence a priori est proportionnelle à la taille du document, elle est exprimée ainsi:

$$P(d) = \frac{|d|}{|C|} \quad \text{(II.3)}$$

Où $|d|$ est la taille de document et $|C|$ est la taille de la collection.

Parapar et al **[34]** ont proposé d'estimer cette probabilité *P(d)* en utilisant la taille compressée d'un document. La formule utilisée dans ce dernier travail est exprimée ainsi :

$$P(d) = \frac{comp(d)}{\sum_{di\,\epsilon\,C} comp(di)} \quad \text{(II.4)}$$

Où *comp(d*) est la taille en octets du document ***d*** compressé (zippé) divisée sur la taille originale en octets du document. Ce nouveau facteur a été évalué et comparé au facteur taille originale de document en utilisant quatre collections TREC. Les résultats présentés montrent que la taille compressée d'un document obtient des améliorations de précision moyenne (MAP) allant de +0,4% à +3,1% par rapport à l'utilisation de la taille originale de document.

II.3.3 La date de création de document

La caractéristique de la date de création de document est aussi utilisée sous l'intuition suivante : « Les documents récents tendent à être plus pertinents que les documents anciens », pour estimer la probabilité a priori d'un document, Li et Croft **[35]**ont proposé un modèle de langue qui permet d'intégrer la notion de « temps » dans l'évaluation de pertinence d'un document vis-à vis d'une requête, où ils assignent une plus grande probabilité de pertinence pour les documents ayant une date de création récente. Ainsi, ils expriment La probabilité de pertinence a priori d'un document sachant sa date de création, comme une distribution exponentielle, exprimée ainsi:

$$P(d|Td) = \lambda\, e^{-\lambda(T_C - T_d)} \quad \text{(II.5)}$$

Où T_C est la date la plus récente dans toute la collection (exprimée en mois) et T_d est la date de création du document.

Les évaluations réalisées ont montré que l'incorporation de la notion de temps en utilisant la distribution exponentielle est bénéfique pour la RI**.**

II.3.4 Le rapport information/bruit

L'intuition de l'utilisation d'une telle caractéristique est qu'un document contenant le plus de tokens après le prétraitement est susceptible d'être plus pertinent qu'un document qui en a moins.

Il est défini comme le rapport entre la taille de document après prétraitement (élimination des mots vides et des balises HTML) et la taille de document sans prétraitement **[39].**

La probabilité de pertinence a priori est exprimée comme suit :

$$P(d) = \frac{I_{token}}{I_{document}} \qquad \text{(II.6)}$$

Où I_{token} est la taille de document après le prétraitement et $I_{document}$ est la taille de document avant le prétraitement. Ainsi, un document avec moins de mots vides et peu de balises HTML produit un haut rapport information/ bruit, ce qui signifie que le document est de « bonne » qualité.

II.3.5 Type d'URL de document

Kraaij et al **[33]** ont utilisé la forme (type) de l'URL pour estimer la probabilité qu'une page soit une page d'entrée. Elle est définie ainsi :

$$P(d) = P(PE \mid url_{type}\ (d)) = \frac{c(PE,t_i)}{c(t_i)} \qquad \text{(II.7)}$$

Où url_{type} est le type de l'URL de document d , $c(PE, t_i)$ est le nombre de documents du type d'URL « t_i » qui sont des pages d'entrées « PE » pour un site web, il est obtenu à partir des évaluations de pertinence et $c(t_i)$ est le nombre de documents de type d'URL « t_i ». Quatre types de catégories d'URL ont été définis : Racine, sous-racine, chemin (répertoire) et Fichier. Sur la base de ces quatre types d'URL, ils ont mené des expérimentations sur la collection web WT10g (collection utilisée dans TREC 2001) pour estimer la probabilité qu'une page soit une page d'entrée sachant son type d'URL. Ils ont constaté que cette source d'information est un bon indicateur pour prévoir la pertinence d'une page.

II.4 Le modèle de langue et la pertinence à priori de documents

Le modèle de langue permet de prendre en compte des fonctionnalités indépendantes des requêtes, c'est-à-dire simplement liées à un document sans avoir de requête. Ensuite, il suit une dérivation du modèle de récupération ML où la probabilité de pertinence p (r | Q, D), étant donné une requête et un document est estimée indirectement en invoquant la règle de Bayes.

Supposons que les variables aléatoires D et Q désignent respectivement un document et une requête. Soit la variable aléatoire binaire R pour la pertinence r, p (r) = p (R = 1) et la non-pertinence $\bar{r}$, p ($\bar{r}$) = p (R = 0).

$$P(r|Q,D) = \frac{p(D,Q)p(r)}{p(D,Q)} \quad \text{(II.8)}$$

$$= p(Q|D,r)p(D|r)\frac{p(r)}{p(D,Q)} \quad \text{(II.9)}$$

$$= p(Q|D,r)\,p(r|D)\frac{p(D)}{p(D,Q)} \quad \text{(II.10)}$$

En supposant l'indépendance entre les requêtes et les documents p (D, Q) = p (D) p (Q), et étant donné que p (Q) n'affecte pas le classement (il est indépendant du document), l'équation devient

$$P(r|Q,D) = \frac{p(Q|D,r)\,p(r|D)}{p(Q)} =_{rank} p(Q|D,r)\,p(r|D) \quad \text{(II.11)}$$

Où p (Q | D, r) est la probabilité de la requête pour le document D et p (r | D) est la probabilité de pertinence à priori du document. Dans l'équation (II.11), nous supposons que la probabilité de pertinence à priori de document est beaucoup plus importante que la probabilité de pertinence de la requête pour le même document pour obtenir une formule finale dépendante de p (r | D). La dérivation présentée précédemment a pris une hypothèse plus raisonnable, Q et D sont indépendants sous r, et à partir du rapport de cotes de pertinence, le score de pertinence final dépend de p (r | D) / (1 - p (r | D).

La requête est décomposée en ses termes Q = {q1, q2,. . . , qn} et supposons que, la pertinence r et le document D sont indépendants les uns des autres et sont générés par une distribution multinomiale.

$$\text{P(Q}|D,r) = \prod_{i=1}^{n} p(q_i|D,r) \quad \text{(II.12)}$$

Afin d'exclure les probabilités nulles pour les termes non vus dans un document, cette estimation doit être lissée.

La plupart des méthodes de lissage utilisent deux distributions, l'une pour les mots apparaissant dans le document (ps) et l'autre pour les mots invisibles (pu).

La technique de lissage la plus populaire et efficace est le lissage préalable de **Dirichlet**:

$$P_{Dir}(q_i|D) = \frac{\mathrm{tf}(\mathrm{t_i},\mathrm{D}) + \mu \mathrm{P_{ML}}(ti|C)}{|\mathrm{D}| + \mu} \quad \text{(II.13)}$$

Où μ est un paramètre appelé pseudo-fréquence.

$tf(t_i, D)$ est la fréquence du terme t_i dans le document D.

$P_{ML}(ti|C)$ est la probabilité qu'un terme t_i soit généré par la collection du modèle de langue P_{ML}.

II.5 Combinaison du score à priori et du score Document/Requête

Pour combiner le score à priori de document avec le score initial (document/ requête) p (Q | D, r) en utilisant la somme logarithmique standard présentée dans la formule (II.14).

Si nous suivons une dérivation de somme logarithmique de l'équation (II.11) alors, la manière standard de combiner le document a priori avec la requête afin de produire un score de document serait:

$$Score\ (D, Q) = log\ p(Q|D, r) + log\ p(r|D) \quad \text{(II.14)}$$

Tel que :

p(Q|D, r) : est la probabilité de pertinence de la requête Q pour chaque document D.

p(r|D) : est la probabilité de pertinence à priori du document D.

II.6 Conclusion

Dans ce chapitre nous avons défini la pertinence à priori de documents, ainsi que les caractéristiques utilisées pour effectuer ce classement .Ensuite, nous avons évoqué la pertinence à priori de documents dans le modèle de langue. Pour enfin présenter le modèle de combinaison de la probabilité à priori de document et de la requête en utilisant la somme logarithmique.

Dans le prochain chapitre nous allons mettre en œuvre une approche qui nous permettra d'étendre le modèle de recherche avec le score à priori de documents et constater une possible amélioration de notre système de recherche possible amélioration de notre système de recherche.

Chapitre 3 : Implémentation et mise en œuvre de l'approche

III.1 Introduction

La probabilité de pertinence à priori de documents est parmi les méthodes utilisée pour améliorer les résultats de la recherche d'information. Notre travail consiste à implémenter et tester une nouvelle méthode d'estimation de la probabilité à priori des documents basée sur des caractéristiques du document. Ce score de pertinence à priori est ensuite combiné avec le score de comparaison document/requête.

Ce chapitre est organisé comme suit, nous allons tout d'abord définir notre approche de calcul de probabilité de pertinence à priori. Par la suite nous allons introduire l'environnement de développement de notre approche en précisant les outils et le langage utilisés pour sa mise en œuvre. Enfin nous présentons les résultats obtenus sur la collection de test TREC AP88.

III.2 Description de l'approche

Notre approche consiste tout d'abord à calculer la probabilité de pertinence à priori de document afin de la combiner avec le score initial (correspondance document/requête). Pour effectuer ce calcul nous nous sommes basés sur les caractéristiques du document qui seront présentées dans la partie suivante. Une fois ces caractéristiques identifiées nous effectuons un apprentissage automatique en utilisant la régression linéaire avec la fonction logistique (sigmoïdale) afin d'obtenir la probabilité de pertinence à priori de documents. Une fois cette probabilité est obtenue nous la combinons avec le score initial.

III.2.1 Les caractéristiques utilisées pour le calcul de la probabilité de pertinence à priori de document :

Pour effectuer le calcul de pertinence à priori des documents, nous devons tout d'abord spécifier les caractéristiques à prendre en compte. Dans notre approche nous avons utilisé les caractéristiques suivantes :

III.2.1.1 La longueur de document :

La longueur du document est le nombre de termes que contient le document, Elle peut être un très bon exemple de jugement de pertinence à priori de documents. Elle peut être utile en cas de requête vague de l'utilisateur, le système retournera donc les documents classés par ordre de longueur au lieu d'un classement aléatoire.

Elle est calculée comme suit :

$$\text{Prior}_{\text{longueur doc}}(D) = |D| \qquad \text{(III.1)}$$

III.2.1.2 Nombre de termes uniques :

L'intuition derrière cette caractéristique est qu'un document qui n'a pas de répétition de mots (le moins de bruit) est mieux classé qu'un document qui contient des mots qui se répètent plusieurs fois. Le nombre de termes uniques signifie le nombre de termes qui apparaissent dans le document en ignorant la redondance.

Elle est formalisée comme suit :

$$\text{Prior}_{\text{termunique}}(D)=|D_t| \quad \text{(III.2)}$$

Où D_t est le nombre de termes dans le document D.

III.2.1.3 Moyenne IDF :

L'intuition derrière cette caractéristique est de classer les documents par rapport à l'importance des termes qu'il contient, un document qui contient des termes rares est mieux classé qu'un document contenant des termes courants.

Ce facteur est calculé comme suit :

$$\text{Prior}_{\text{MoyIDF}}(D)= \frac{\sum_{t_i \in D} \text{IDF}(t_i)}{|D_t|} \quad \text{(III.3)}$$

Tel que :

$$\text{IDF(t)}= \log\left(\frac{\text{nbDoc}}{\text{nb}Doc(t)}\right) \quad \text{(III.4)}$$

Où nbDoc est le nombre de documents dans la collection et nbDoc(t) est le nombre de documents contenant un terme.

III.2.1.4 Écart type IDF :

L'intuition derrière cette caractéristique est que plus $Prior_{\text{Écart type IDF}}(D)$ est grand, la probabilité de pertinence à priori de document est minime.

Il est calculé de la manière suivante :

$$Prior_{\text{Écart type IDF}}(D)= \sqrt{\frac{\sum(\text{IDF(t)}-\text{moyenneIDF(D)})^2}{|D|}} \quad \text{(III.5)}$$

Où IDF(t) est donné dans la formule (III.4).

III.2.1.5 Écart type TF :

L'intuition derrière cette caractéristique est que plus $Prior_{\text{Écart type TF}}(D)$ est grand, la probabilité de pertinence à priori de document diminue.

Il est calculé de la manière suivante :

$$Prior_{\text{Écart type TF}}(D)=\sqrt{\frac{\sum(\text{TF(t)}-\text{moyenneTF(D)})^2}{|\text{D}|}} \quad \text{(III.7)}$$

Où TF(t) est la fréquence d'un terme dans le document D.

III.2.1.6 Rapport TFmin/TFmax

Le rapport TFmin/TFmax permet de calculer l'homogénéité des fréquences des termes du document. Plus ce rapport est minime plus la probabilité de pertinence du document augmente.

Il est calculé comme suit :

$$\text{Prior}_{\text{rapport}}(\text{D})=\frac{\text{TFmin(D)}}{\text{TFmax(D)}} \quad \text{(III.9)}$$

Où TFmin(D)et TFmax(D) sont respectivement la fréquence minimale et maximale d'un terme dans un document D.

III.2.1.7 Entropie

L'intuition derrière cette caractéristique est d'estimer la cohérence de documents, les documents avec une valeur de l'entropie assez importante ont tendance à être plus cohérents et donc une forte probabilité de pertinence à priori du document.

Elle est calculée de la manière suivante :

$$Prior_{\text{Entropie}}(D)=-\sum_{t_i \in D}\left(\frac{\text{TF(ti,D)}}{|\text{D}|}\right)\log\left(\frac{\text{TF(ti,D)}}{|\text{D}|}\right) \quad \text{(III.9)}$$

Où TF(ti, D) est la fréquence d'un terme t_i dans le document D.

III.2.2 L'apprentissage de la probabilité de pertinence à priori de documents :

Une fois les caractéristiques des documents sont obtenues, nous entamons la phase d'apprentissage qui nous permettra de prédire la probabilité de pertinence des documents en utilisant la régression linéaire.

L'apprentissage de la probabilité de pertinence à priori des documents se fera avec un échantillon de documents de la collection, où la pertinence à déjà été jugée (Figure III.1) afin de récuperer les coefficients de corrélation entre les caractérisiques de document et la probabilité de pertinence de document.

La régression linéaire se fera avec la fonction logistique présentée ci-dessous :

$$Y = \frac{1}{1+e^{-wX}} \qquad \text{(III.11)}$$

Tel que :

Y : La probabilité de pertinence à priori de document.

w : Les coefficients de corrélation entre les variables X et la pertinence Y.

X : Les variables explicatives (les caractéristiques décrites dans la section précédente).

Cette fonction vise à expliquer une variable d'intérêt binaire (c'est-à-dire de type « oui / non », « vrai / faux »ou « 0 / 1»). Les variables explicatives qui seront introduites dans le modèle sont quantitatives.

Nous avons utilisé les Qrels de requêtes numérotées de 51 à 100 comme donnés d'apprentissage. Ci-dessous un extrait du fichier Qrels :

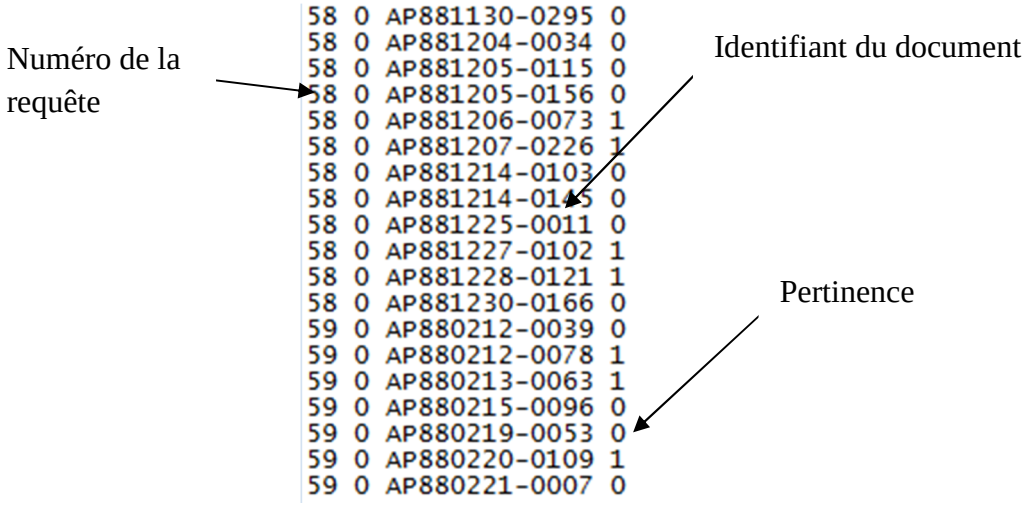

Figure III-1 : Extrait du fichier des Qrels

Dans notre cas les variables explicatives sont pour chaque document : sa longueur, le nombre de termes uniques, l'écart type TF, l'écart type IDF, la moyenne IDF, le rapport TFmin/TFmax et l'entropie. Ces variables expliqueront la pertinence des documents.
Une fois la fonction logistique est apprise sur le jeu de donnés, nous utiliserons ses résultats pour estimer la pertinence à priori d'un document « D » noté comme suit : Prior (D).

III.2.3 La combinaison des scores :

Afin de combiner les scores à priori des documents et les scores initiaux nous avons utilisé la somme logarithmique citée dans le chapitre précédent, elle est exprimée comme suit:

$$Score\ Final\ (D,Q)\ =\ Score\ Initial\ (D,Q)\ +\ \lambda\ Prior(D) \qquad (\text{III.}12)$$

Nous avons varié la valeur de λ de 0.5 à 5 avec un pas de 0.5 et la valeur qui a donné le meilleur résultat est λ=3.

III.2.4 Les outils et langages utilisés :

Pour la mise en œuvre de notre approche nous avons utilisé des outils et des langages de programmation que nous présentons ci-dessous :

III.2.4.1 Le langage de programmation Java :

La technologie **Java** définit à la fois un langage de programmation et une plateforme informatique. Crée par l'entreprise Sun Microsystems, et reprise depuis par la société Oracle, la technologie Java est indissociable du domaine de l'informatique et du Web.

La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitations tels que Unix, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications.

Le langage Java reprend en grande partie la syntaxe du langage C++ , Néanmoins Java à été épuré des concepts les plus subtils du C++ et à la fois les plus déroutants , tels que les pointeurs et références , ou l'héritage multiple contourné par l'implémentation des interfaces.

Java a donné naissance à un système d'exploitation (JavaOS) , à des environnements de développement (eclipse/JDK) , des machines virtuelles (MSJVM (en),JRE) applicatives multiplateforme (JVM) , une déclinaison pour les périphériques mobiles/embarqués (J2ME),une bibliothèque de conception d'interfaces graphiques (AWT/Swing) , des

applications lourdes (Jude,Oracle,SQL Worksheet ,etc.), des techniques web (servlets, applets) et une déclinaison pour l'entreprise (J2EE).

III.2.4.2 NetBeans :

NetBeans est un environnement de développement intégré, placé en open source par Sun en juin 2000 sous licence CDDI. (Common Devloppement and Distribution License) et GPLv2. En plus de Java, Netbeans permet la prise en charge native de divers langage tels le C, C++, JavaScript, le XML, le PHP et le HTML, où d'autres (dont Python et Ruby) par l'ajout de greffons. Il offre toutes les facilités d'un IDE moderne (éditeur avec coloration syntaxique, projets multi-langages, refactoring, éditeur graphique d'interfaces et de pages Web).

NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing(Java)). L'IDE NetBeans s'appuie sur cette plateforme.

Il supporte principalement les langages suivants :

Java, Javadoc, JavaScript12, C.

Les plates-formes supportées sont :

Microsoft Windows, Linux, Mac OS ; Solaris10.

Nous vous présentons ci-dessous l'interface qu'on a générée avec plate-forme NetBeans :

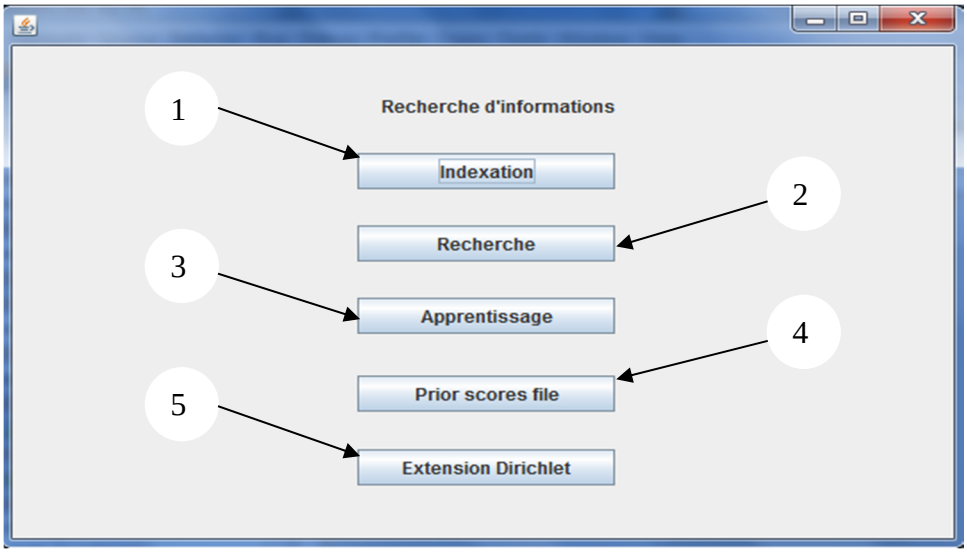

Figure III-2 : Interface Netbeans de notre approche

Tel que :

Bouton 1 : Permet d'effectuer l'indexation de la collection AP88.

Bouton 2 : Permet d'effectuer la recherche avec le modèle de langue Dirichlet (Formule II.13).

Bouton 3 : Permet le calcul des caractéristiques pour un échantillon de test Qrels AP88 afin d'effectuer l'apprentissage.

Bouton 4 : Permet de calculer le score à priori des documents, et les stocker dans un fichier.

Bouton 5 : Permet d'effectuer la recherche avec l'extension du modèle Dirichlet en se basant sur la formule(III.12).

III.2.4.3 Terrier :

Terrier est une plate-forme dédiée à la recherche d'information. Elle implémente les différents modules intervenant dans le processus de RI classique et offre en plus un cadre pour l'évaluation des résultats de recherche pour différentes applications. Terrier a été largement éprouvé**.** Le choix de cette plate-forme est dû aussi à sa capacité à traiter de grandes collections de documents telles que les collections TREC.

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

La phase de recherche comprend le Manager, un module qui interagit avec l'application, réalise la mise en correspondance à travers les calculs des pondérations (selon le schéma de pondération (Weighting Model) choisi : PL2, BM25, Dirichlet LM etc.) ainsi que les scores des documents. Le résultat renvoyé à l'utilisateur, est la liste des documents jugés pertinents et leurs scores respectifs, dans notre cas on a choisi d'étendre la classe Dirichlet LM.

III.2.4.4 RStudio :

R est un langage de traitement et d'analyse de données de plus en plus répandu, notamment grâce à sa puissance et au fait qu'il est libre, gratuit et multiplateforme.

RStudio est un outil apparu récemment et qui vient combler un manque dans la collection des outils associés à R : il s'agit d'un environnement de développement intégré (*IDE* en anglais) fonctionnel, libre, gratuit et multiplateforme.

Un IDE n'est pas une interface graphique au sens de SPSS ou Modalisa, qui permettrait d'utiliser le logiciel à travers des menus et des boîtes de dialogue : il s'agit d'un environnement facilitant la saisie, l'exécution de code, la visualisation des résultats, etc.

RStudio est multiplateforme, vous pouvez donc le télécharger et le faire fonctionner aussi bien sous Windows, Mac OS X ou Linux :

Son interface se présente sous la forme d'une unique fenêtre découpée en quatre zones que l'on peut redimensionner, masquer ou maximiser selon ses préférences.

Dans notre cas nous avons utilisé ce logiciel afin d'apprendre la probabilité de scores à priori pour un échantillon de 8080 documents (extrait présenté dans la Figure III.1) et récupérer les coefficients de corrélation des variables explicatives citées dans (III.2.1).

III.3 Architecture de notre approche :

Nous présentons ci-dessous l'architecture de notre approche :

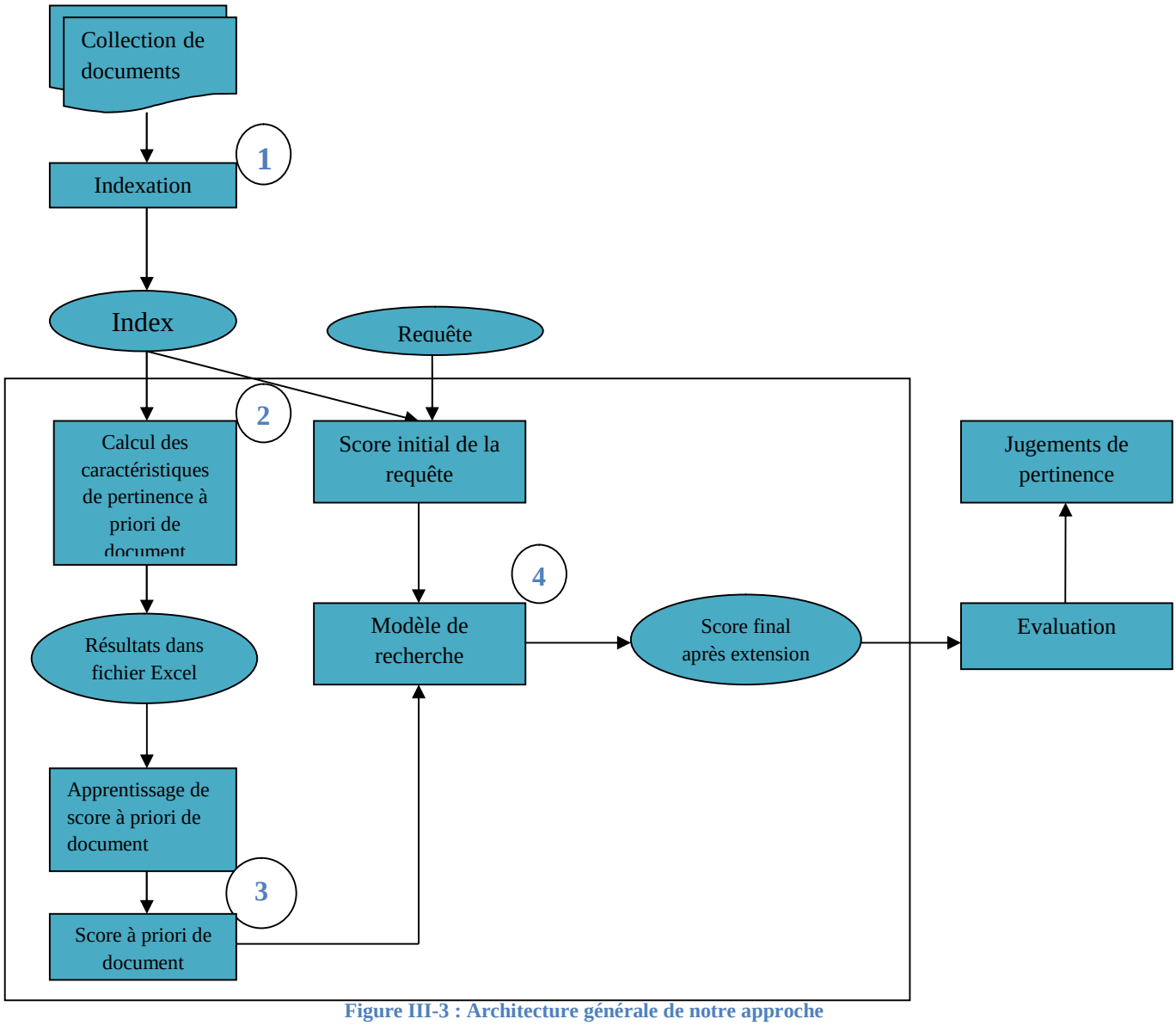

Figure III-3 : Architecture générale de notre approche

III.3.1 Indexation des documents

La première étape à faire avant la recherche est l'indexation de la collection en utilisant la classe TRECIndexing de la plateforme Terrier.

III.3.2 Calcul des caractéristiques utilisées pour le calcul de la probabilité de pertinence à priori des documents

Afin de calculer la probabilité à priori de pertinence de documents nous avons tout d'abord, pris un échantillon de Qrels (jugement de pertinence) afin de calculer les caractéristiques pour chaque document de cet échantillon de 8080 documents pour enfin classer les résultats ainsi que le jugement de pertinence dans un fichier Excel qui sera chargé dans le logiciel RStudio afin d'effectuer une régression linéaire (figure ci-dessous) :

	docLength	termuni	MoyenneIDF	EcartTypeIDF	EcartTypeTF	Rapp	entr	pertinence
1	246	157	2.976735	1.713730	1.3697082	0.07692308	4.832545	1
2	135	86	3.478316	2.283547	1.1567868	0.14285714	4.259413	0
3	272	190	3.810235	2.333066	0.9857151	0.12500000	5.083984	1
4	537	315	2.939048	1.517538	1.9202702	0.03703704	5.453510	0
5	251	178	3.346908	1.867792	1.0628079	0.10000000	5.007280	0
6	335	198	3.118314	1.641085	1.4842729	0.09090909	5.036312	0
7	169	106	3.268336	2.177996	1.4326272	0.09090909	4.418075	1
8	540	336	3.114644	1.631396	1.3867387	0.09090909	5.580365	1
9	232	145	3.313017	1.943166	1.1590722	0.12500000	4.784775	0
10	474	285	3.050225	1.785104	1.6771563	0.04545454	5.395081	0
11	435	269	3.161132	1.704615	1.6759410	0.05555556	5.316633	0
12	241	157	3.135855	1.973552	1.2442411	0.11111111	4.841396	0
13	253	137	3.149648	1.939627	1.8199643	0.10000000	4.604044	0
14	244	160	3.462004	2.007828	1.2346558	0.11111111	4.861842	0
15	192	116	3.139418	2.064152	1.6140731	0.09090909	4.477463	0
16	169	114	2.973938	1.834310	1.1411940	0.12500000	4.543653	0
17	463	282	3.342122	1.968776	1.4956512	0.09090909	5.387292	1
18	407	246	3.136144	1.833237	1.6694652	0.05555556	5.238889	1
19	290	191	3.403503	1.903473	1.3177918	0.07692308	5.029534	1
20	565	359	2.880105	1.414077	1.4509567	0.05555556	5.645137	0

Figure III-4 : Fichier Excel contenant les valeurs des caractéristiques utilisées pour le calcul du score à priori de pertinence de documents

III.3.3 Apprentissage et calcul de scores à priori de documents :

Une fois le fichier chargé nous allons effectuer une régression linéaire en utilisant la fonction logistique expliquée précédemment en utilisant les commandes de la figure suivante :

```
setwd("C://poiexcel")
donnes<-read.csv("writesheet.csv",header = T )
model<-glm(pertinence~docLength+termuni+MoyenneIDF+EcartTypeIDF+EcartTypeTF+Rapp+entr,family = "binomial"(link = 'logit'),data = donnes)
```

Figure III-5 : Commandes utilisées pour la mise en œuvre de la fonction logistique

Cette régression nous permettra de récupérer les coefficients de corrélation qui vont être utile pour le calcul du score à priori de document. Pour savoir quelles caractéristiques prendre en compte dans notre approche, on exécute la commande suivante :

anova (model, test="Chisq")

On aura donc le résultat présenté dans la Figure III.6, A partir de ce résultat il est désormais possible de distinguer les caractéristiques qui ont une influence considérable sur la probabilité de pertinence à priori de document, tel que plus la valeur du test **Pr (>Chi) (**un test qui permet de déterminer une association significative entre deux catégories) est inférieur à 0.01 (p-value), la corrélation entre la variable explicative et la pertinence est grande.

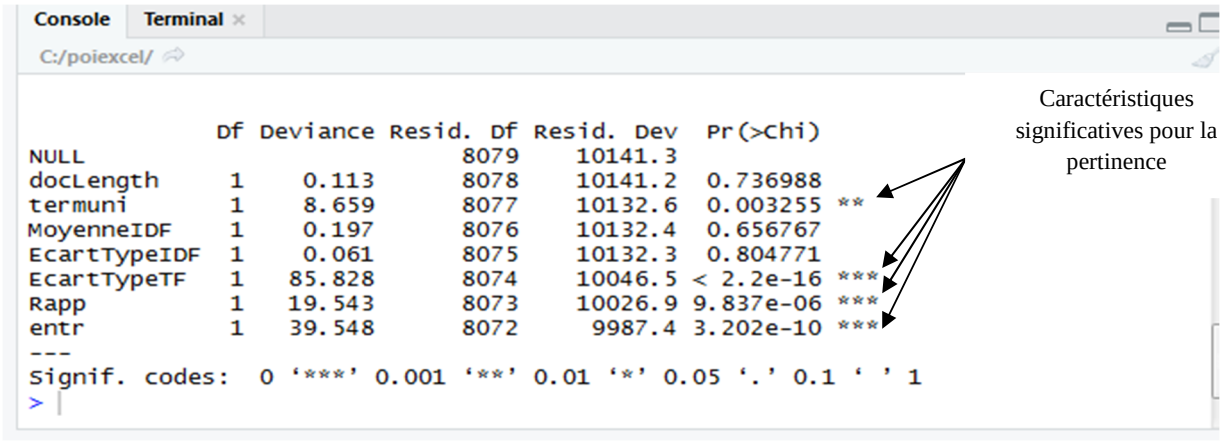

Figure III-6: Résultats de la corrélation entre les variables explicatives et la pertinence

Après avoir distingué les caractéristiques les plus influentes sur la pertinence on pourra donc choisir les coefficients à retenir pour le calcul du score à priori de document.

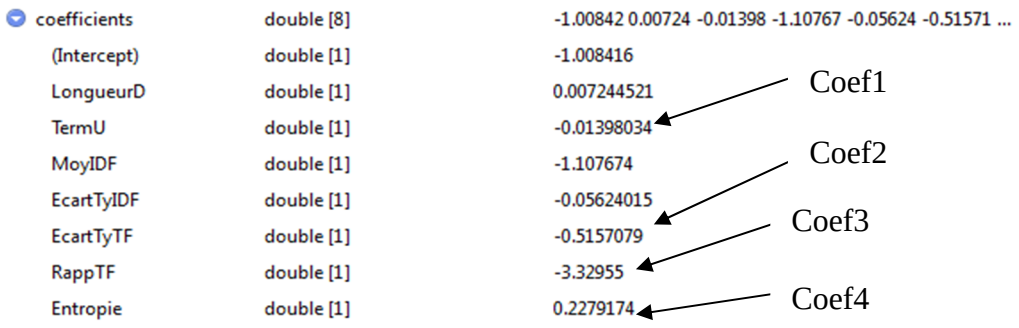

Figure III-7: Coefficients de corrélation entre les caractéristiques et la Pertinence

Après la récupération des coefficients de corrélation, nous calculons le score à priori de document en utilisant la formule suivante :

$$\text{Prior (D)} = \frac{1}{1+e^{-Z}} \qquad \text{(III.13)}$$

Tel que :

$$\text{Z(D)} = \text{coef1} \times \text{car1(D)} + \text{coef2} \times \text{car2(D)} + \ldots\ldots + coef_n \times car_n(D) \qquad \text{(III.14)}$$

A partir de la figure III.6 on constate que le nombre de termes uniques, l'écart type TF, le rapport TFmin/TFmax et l'entropie sont des variables très significatives pour Y qui est la pertinence, par conséquent nous avons considéré que ces caractéristiques dans le calcul du score de pertinence à priori. La figure suivante montre un extrait du fichier obtenu.

```
AP880903-0173=0.9424019031607664
AP880903-0172=0.9597758134183703
AP880903-0171=0.8886216445195803
AP880903-0170=0.9657471821567375
AP880825-0239=0.9622846798050319
AP880825-0238=0.908272810697752
AP880825-0237=0.9446172094991916
AP880825-0236=0.9770416569728929
AP880825-0235=0.8586555678981126
AP880825-0234=0.9599965084412845
AP880825-0233=0.9146509785500051
AP880825-0232=0.6274360385411388
AP880825-0231=0.9401424902285082
AP880825-0230=0.9681465058492944
AP880903-0169=0.9312424249667796
AP880903-0168=0.74791256967496
AP880903-0167=0.8415901796164752
AP880903-0166=0.7220253676052297
AP880903-0165=0.6936804256268455
AP880903-0164=0.9664172911409031
AP880903-0163=0.5471555630724665
AP880903-0162=0.3225868772987033
AP880903-0161=0.6964567457144384
AP880903-0160=0.008048372641645181
AP880825-0229=0.9282145903960114
AP880825-0228=0.9512393332726228
AP880825-0227=0.9828902626745522
AP880825-0226=0.8982126630805426
AP880825-0225=0.8950444676281828
AP880825-0224=0.8084842483752701
AP880825-0223=0.9697781853841538
AP880825-0222=0.9793471349616042
AP880825-0221=0.9584223158570822
```

Figure III-8 : Un extrait du fichier contenant des scores à priori de documents

III.3.4 Extension du modèle de recherche avec la score à priori de document :

Une fois la probabilité de pertinence à priori de document calculée on procédera à l'extension du modèle de recherche Dirichlet avec µ=2500 (par défaut) en utilisant la combinaison précédemment citée dans la formule (III.12).

III.4 Evaluation et résultats

Dans cette section nous présentons dans un premier temps la collection et les requêtes utilisées ainsi que les mesures d'évaluation adoptées. Ensuite nous présentons les résultats obtenus par notre approche comparativement au modèle de base (Dirichlet). Pour avoir une idée plus précise, nous avons analysé aussi les résultats requête par requête.

III.4.1 La collection de test et les requêtes utilisés :

Différentes collections de tests sont utilisées en recherche d'information. La collection que nous avons utilisé dans notre étude est : la collection **TREC AP88** (**A**ssociated **P**ress Newswire, 19**88**). Elle contient 79 919 documents.

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

Pour effectuer l'évaluation sous Windows il suffit d'accéder à l'emplacement de votre fichier Terrier, puis accéder au fichier bin et saisir les commandes suivantes :

- trec_terrier –e pour une évaluation simple.
- trec_terrier –e –p pour une évaluation requête par requête.

Afin d'évaluer les résultats, nous avons utilisé la mesure MAP, qui est la mesure la plus utilisée en recherche d'information et la précision à 1, 10, 15 et 20 documents.

III.4.2 Résultats obtenus avant et après l'extension des modèles de recherche :

Dans cette section nous présentons les résultats globaux et détaillés de l'évaluation du modèle de langue Dirichlet avant et après la mise en œuvre de notre approche.

Le tableau ci-dessous montre les résultats obtenus avec le modèle de base (Dirichlet) et notre extension du modèle de base, ainsi que les améliorations constatées.

Résultats de précision			
Précision	**Dirichlet_2500**	**Après notre approche**	**Taux d'amélioration**
MAP	0.2437	**0.2452**	**+0.61%**
P@1	0.4694	**0.4898**	**+4.34%**
P@10	**0.3347**	0.3306	-1.22%
P@15	0.3061	**0.3129**	**+2.22%**
P@20	0.2959	**0.2969**	**+0.33%**

Tableau III -1 : Résultats des deux modèles comparés

A partir du tableau (III.1) nous constatons que notre approche améliore le modèle de base (Dirichlet) notamment avec la mesure P@1 où un taux d'amélioration de +4.34% est constaté.

Pour avoir une idée plus précise de l'impact de notre extension nous avons analysé la précision requête par requête (voir figure III.8 et tableau III.2).

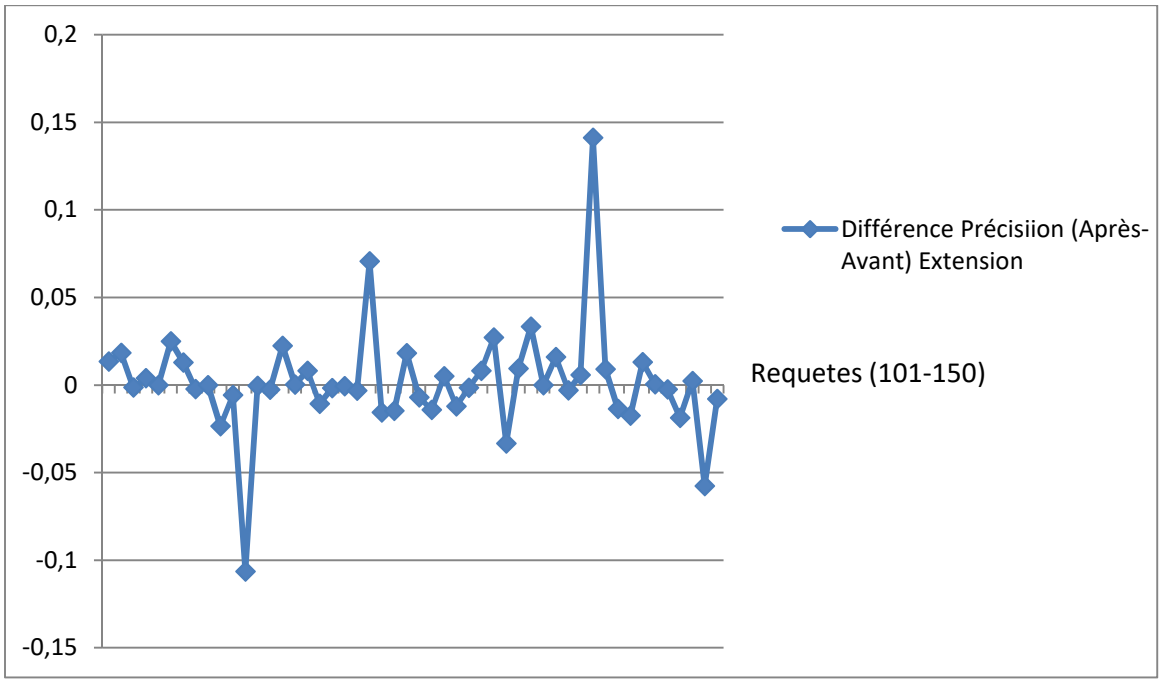

Figure III-1 Analyse requête par requête sur la collection AP88.

A partir de ce graphique nous avons noté que notre approche à amélioré le modèle de base sur 22 requêtes soit 44% du nombre total de requêtes.

Dans le tableau ci-dessous nous avons sélectionné les 22 requêtes améliorées :

Requête	MAP avant notre approche	MAP après notre approche	Taux d'amélioration
R101	0.1193	0.1328	11.31%
R102	0.3557	0.3740	5.14%
R104	0.2716	0.2755	1.43%
R106	0.3810	0.4059	6.53%
R107	0.4026	0.4155	3.20%
R115	0.2064	0.2288	10.85%
R116	0.0013	0.0015	1.15%
R117	0.2247	0.2329	3.46%
R122	0.3715	0.4421	19%
R125	0.2168	0.2349	8.34%
R128	0.0988	0.1039	5.16%
R131	0.0422	0.0504	19.43%
R132	0.7943	0.8217	3.44%
R134	0.6917	0.7011	1.35%
R135	0.4034	0.4368	8.27%
R137	0.1571	0.1731	10.18%
R139	0.0437	0.0494	13.04%
R140	0.1194	0.2606	118.25%
R141	0.0934	0.1024	9.64%
R144	0.0404	0.0534	32.18%
R145	0.1429	0.1433	0.27%
R148	0.0373	0.0396	6.17%

Tableau III-2 : Les requetes améliorées par notre approche avec le modèle de recherche Dirichlet

III.5 Conclusion :

Dans ce chapitre nous avons présenté en premier lieu notre approche de calcul de la pertinence à priori de documents. Cette pertinence est estimée via une méthode d'apprentissage basée sur un ensemble de caractéristiques : longueur de document, nombre de termes uniques dans le document, moyenne IDF, écart type TF, écart type IDF, Rapport TFmin/TFmax et l'entropie. Nous avons ensuite combiné cette pertinence avec la pertinence classique (Document/Requête).

Les évaluations effectuées sur une collection de test AP88 ont montré des améliorations notamment avec les mesures de haute précision.

Conclusion générale

Conclusion générale

Notre travail présenté dans le cadre de ce mémoire s'insère dans le domaine de la recherche d'information. Il porte sur l'extension d'une approche d'expansion de modèle de langue en utilisant la pertinence à priori de documents.

Pour mener à terme notre travail, nous avons donné un aperçu général sur la recherche d'information ainsi que le système de recherche d'information.

Nous avons ensuite défini la pertinence à priori de documents et quelques caractéristiques qui permettent de la calculer.

Pour mettre en œuvre notre approche qui est « l'extension du modèle de recherche en utilisant la probabilité de pertinence à priori de documents » nous avons utilisé la plateforme Terrier, le langage de programmation Java, l'environnement NetBeans et Rstudio qui nous a permit d'effectuer des calculs statistiques.

L'approche proposée a apporté une légère amélioration globale par rapport au modèle de recherche initial. En plus de ça, on a constaté des améliorations sur un certain nombre de requêtes. Ce qui est un bon indice.

Bibliographie

Bibliographie :

[1] Hernandez N. “Ontologie de domaine pour la modélisation du contexte en recherche d’information”, thèse de doctorat en informatique, Université Paul Sabatier. (2006)

[2] Boubekeur, F. “Contribution à la définition de modèles de recherche d'information flexibles basés sur les CP-Nets”. Thèse de doctorat en informatique, Université Paul Sabatier. 2008

[3] Daoud, M. “Accès personnalisé à l'information : approche basée sur l'utilisation d'un profil utilisateur sémantique dérivé d'une ontologie de domaines à travers l'historique des sessions de recherche”, thèse de doctorat en informatique, Université Paul Sabatier. (2009)

[4] Bouramoul, A, Thèse de doctorat : Recherche d’information contextuelle et sémantique sur le web.2011.

[5] Hammache, A. Thèse de doctorat : Recherche d’information :un modèle de langue combinant mots simples et mots composés, Université Mouloud Mammeri Tizi-Ouzou. 2013.

[6] Baeza-Yates, R., Ribeiro-Neto, B. A. Modern Information Retrieval. Pearson Education Ltd., Harlow, UK, 2nd edn, (2011).

[7] Robertson, S.E., Walker. S., Jones, S., Hancock-Beaulieu, M. and Gatford, M. Okapi at trec-3. TREC, pp. 109-126, 1994.

[8] Ren, F., Fan, L., Nie, J-Y. SAAK Approach: How to Acquire Knowledge in an Actual Application System. International Conference on Artificial Intelligence and Soft Computing, Honolulu, pp.136-140, 1999.

[9] Jacquemin, C., Daille, B., Royanté, J., and Polanco, X. In vitro evaluation of a program for machine-aided indexing. Inf. Process. Manage. 38, 6, pp. 765-792. 2002.

[10] Fox, C. Lexical analysis and stoplists, Frakes W B, Baeza-Yates R (eds) Prentice Hall, New jersey, pp. 102– 130. 1992.

[11] Adamson, G, Boreham, J. The use of an association measure based on character structure to identify semantically related pairs of words and document titles. Journal of Information Storage and Retrieval vol. 10, no. 7-8, pp. 253-260, 1974

[12] Baziz M., Indexation Conceptuelle Guidée Par Ontologie Pour La Recherche d'Information. Thèse de Doctorat en Informatique de l'Université Paul Sabatier de Toulouse, 2005

[13] Manning, D., Raghavan, P. And Schute, H. Introduction to Information Retrieval. Cambridge University Press, 2008.

[14] Witten, I., Moffat, A., and Bell, T. Managing Gigabytes: Compressing and Indexing Documents and Images, Van Nostrand Reinhold, New York, 1994.

[15] Williams, H., Zobel, J. Compressing Integers for Fast File Access. Computer Journal 42, pp. 193-201, 1999.

[16] Dominich, S. Mathematical Foundations of Information Retrieval. Kluwer Academic Publishers, Dordrecht, Boston, London, 2001.

[17]Salton, G. The Smart Retrieval System : Experiements in automatic document Processing. Prentice-Hall,1971.

[18] Robertson, S.E. , S. Walker. On relevance weights with little relevance information. Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 16–24, 1997.

[19] Singhal, A., Salton, G., Mitra, M., Buckley, C. Document length normalization. Information Processing and Management, 32(5), pp. 619–633, 1996.

[20] Van Rijsbergen, C. J. Information retrieval. London: Butterworth, 1979.

[21] .Jelinek, F. Statistical Methods for Speech Recognition. MIT Press, Cambridge, MA, 1998.

[22] Brown, P., Della Pietra, S., Della Pietra, V., and Mercer, R. The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics, 19(2), pp. 263-311, 1993.

[23] Manning, D., Schütze, H. Foundations of Statistical Natural Language Processing. MIT Press, 2000.

[24] J. H. Lee. “Combining the evidence of different relevance feedback methods for information retrieval”. Information Processing and Management, 34(6) :681-691, 1998.

[25] P. Ingwersen. “Polyrepresentation of information needs and semantic entities: elements of a cognitive theory for information retrieval interaction”. In Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval., pages 101-110, 1994.

[26] N. J. Belkin, P. Kantor, E. A. Fox, and J. A. Shaw. “Combining the evidence of multiple query representations for information retrieval”. In Information Processing and Management., pages 431-448, 1995.

[27] L. Tamine. “Optimisation de requêtes dans un système de recherche d’information approche basée sur l’exploitation de techniques avancées de l’algorithmique génétique”. pages 14-28, Décembre 2000

[28] .Liu, X. and Croft, W. B. Cluster-Based Retrieval Using Language Models. Proceedings of the 27th ACM SIGIR Conference on Research & Development on Information Retrieval, 186-193, 2004

[29] Voorhees, E. M. Using WordNet to disambiguate word senses for text retrieval. International Conference on Research and Development in Information Retrieval, pp. 171–180, 1993. **[31]**

[31] Voorhees, E.M. & Harman, D.K. TREC: Experiment and Evaluation in Information Retrieval. Digital Libraries and Electronic Publishing, MIT Press, 2005.

[32] Boughanem, M. Outils de validation en recherche d'information. La campagne d'évaluation TREC, 2003. http://inforsid2003.loria.fr/resumeConfRI.pdf, 2003.

[33] : .Kraaij, W., Westerveld, D., Hiemstra, D. The Importance of Prior Probabilities for Entry Page Search. Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 27– 34, 2002.

[34] : Parapar, J., E.Losada, D., Barreiro, A. «Compression-Based Document Length Prior for Language Model». In ACM International Conference on Research and Development in Information Retrieval, 2009.

[35] : .Liu, X. and Croft, W. B. Cluster-Based Retrieval Using Language Models. Proceedings of the 27th ACM SIGIR Conference on Research & Development on Information Retrieval, 186-193, 2004.

[36] : Hauff, C., Azzopardi, L. «Age dependent document priors in link structure analysis». In The 27th European Conference in Information Retrieval, 2005, p. 552–554.

[37] : Li, X., Croft, W.B. «Time-based language models». In Proceedings of the twelfth international conference on Information and knowledge management, 2003, p. 469–475.

[38] : Diaz, F., Jones, R. «Using temporal profiles of queries for precision prediction». In Proceedings of the 27th annual international conference on Research and development in information retrieval, 2004, p. 18–24.

[39] : Zhu, X.L., Gauch, S. «Incorporating quality metrics in centralized / distributed information retrieval on the world wide web», In Proceedings of the 23th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2000, p. 288– 295.

[40] : Cohn, D., Chang, H., «Learning to probabilistically identify authoritative documents». In Proceedings of the 17th International Conference on Machine Learning, 2000.

[41] : Brin, S., Page, L., «The anatomy of a large-scale hypertextual web search engine». In Proceedings of WWW7 (Brisbane, Australia, May 1998). http://www7.scu.edu.au/programme/fullpapers/1921/com1921.html

[42] : Upstill, T., Craswell, N., Hawking, D. «Predicting fame and fortune: PageRank or indegree? », In Proceedings of the Australasian Document Computing Symposium, 2003.

[43] : Craswell, N., Robertson S., Zaragoza, H. and Taylor, M. «Relevance weighting for query independent evidence», In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2005, p. 416–423.

[44] : Peng, J., Ounis I. «Combination of document priors in web information retrieval», In Proceedings of European conference on information Retrieval, 2007, p. 732–736.