REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire

De fin d'études

En vue de l'obtention d'un Master en informatique

Spécialité : Réseaux, Mobilité et Systèmes Embarqués

Thème

Implémentation et évaluation d'une extension du modèle MRF

Proposé et encadré par :

Mr HAMMACHE AREZKI

Réalisé par :

Mr BERRICHI AMIROUCHE

Mr OUHACHI RAFIK

Année universitaire : 2014/2015

Remerciements

Nous remercions le dieu le tout puissant de nous avoir donné assez de courage et de persévérance pour réaliser ce travail.

Nous exprimons notre profonde gratitude à notre cher promoteur, Monsieur A.HAMMACHE pour avoir dirigé ce travail, pour la documentation qu'il nous a fourni, pour ses critiques, ses conseils et ses orientations durant toute la période de notre travail.

Nous tenons à remercier les membres de jury pour avoir examiné le présent document et l'honneur d'accepter de juger et de porter leurs suggestions sur ce travail.

Enfin, nous tenons à remercier nos familles respectives ainsi que toute personne ayant contribué de près ou de loin à la rédaction du présent sujet.

Dédicaces

Je dédie ce modeste travail à :

Mes parents pour leur amour inestimable, leurs confiances,
Leurs soutiens, leurs sacrifices et leurs patiences
tout le long de ma vie.

A tous mes amis (es) de la promotion RMSE.
A mon chère amie et binôme, avec lequel j'ai eu
le plaisir de partager ce travail, et à tous
les membres de sa famille.

AMIROUCHE.

Dédicaces

Je dédie ce modeste travail à :

Mes parents pour leur amour inestimable, leurs confiances,
Leurs soutiens, leurs sacrifices et leurs patiences
tout le long de ma vie.

A tous mes amis (es) de la promotion RMSE.
A mon chère amie et binôme, avec lequel j'ai eu
le plaisir de partager ce travail, et à tous
les membres de sa famille.

RAFIK.

Sommaire

Sommaire

Sommaire

Liste des tableaux

Table des figures

Introduction générale

La Recherche d'information (RI) concerne les méthodes et mécanismes qui permettent la création et l'utilisation d'une base d'information. Elle propose des outils, appelés Systèmes de Recherche d'Information (SRI), dont l'objectif est de capitaliser un volume important d'informations et d'offrir des moyens permettant de localiser les informations pertinentes relatives au besoin d'un utilisateur exprimé à travers d'une requête.
Tout système de RI a un but précis : établir une correspondance entre l'information disponible et celle recherchée par l'utilisateur et ce par la mise en œuvre d'un mécanisme d'appariement entre la requête de l'utilisateur et les documents ou plus exactement entre la représentation des informations de la requête et la représentation des informations apparues dans la collection des documents.

L'efficacité d'un système de recherche d'information est souvent évaluée par deux métriques importantes qui sont le rappel et la précision, cette évaluation se fait souvent en utilisant les collections de test et les campagnes d'évaluation dont la plus connue est la campagne d'évaluation TREC.
Pour effectuer la correspondance entre documents et requêtes des modèles de RI sont utilisés.
Les modèles de RI se basent souvent sur la représentation en sac de mots des documents, c'est à dire les relations entre termes sont ignorées.
Cette simplicité de la représentation des documents facilite grandement les calculs. Cependant, elle introduit la perte d'une certaine sémantique des documents.
Afin d'outrepasser cette simplification des travaux ont été fait pour intégrer les relations entre termes. Parmi ces travaux on trouve le modèle MRF (Markov random field) [51].

L'objet de notre travail est de proposer une extension au modèle MRF en intégrant un nouveau facteur qui est la couverture spatiale d'un terme vis-à-vis d'un document.
Pour ce faire, nous avons structuré notre mémoire en trois chapitres :

Dans le ***premier chapitre***, nous présentons les concepts de base de la RI, notamment: le processus d'indexation et de recherche, les modèles de RI, la reformulation de la requête et les différentes mesures d'évaluation des SRI.

Le deuxième chapitre à pour but de présenter les différents facteurs pris en compte dans la pondération : facteurs classiques et les facteurs de la structure de document, la position des termes de la requête dans les documents et le facteur de proximité.

Le troisième chapitre est consacré à la description du modèle MRF et son extension basé sur le facteur de couverture spatiale. Puis, l'environnement de développement les outils utilisées sont présentés, et en fin les résultats d'évaluation de notre approche sont présentés.

Nous terminerons notre travail par une conclusion générale.

I.1Introduction

Face à l'accroissement rapide du volume documentaire stocké sous format numérique, est née la nécessité de mettre en place des systèmes et mécanismes facilitant l'accès aux informations contenues dans de tels volumes documentaires. Les systèmes mis en œuvre dans le cadre de la Recherche d'Information (RI), encore appelés Systèmes de Recherche d'Information (SRI), offrent des mécanismes et des techniques qui facilitent le stockage, l'organisation et l'accès aux informations souhaitées, contenues dans des collections de documents. Leur objectif principal étant de retrouver les documents pertinents susceptibles de répondre au mieux à un besoin en information d'un utilisateur, exprimé sous forme de requête.

Ce chapitre a pour but de présenter le domaine de la RI, 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, et en fin nous discutons de l'évaluation des systèmes de recherche d'information.

I.2La Recherche d'Information

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 [1].

I.3Les Systèmes de Recherche d'Information (SRI)

I.3.1Définition

Un Système de Recherche d'Informations (SRI) est un système informatique qui permet de retourner à partir d'un ensemble de documents, ceux dont le contenu correspond le mieux à un besoin en informations d'un utilisateur, exprimé à l'aide d'une requête [2].

I.3.2 Architecture d'un SRI

Historiquement, la gestion des documents concernait surtout des spécialistes : bibliothécaires, documentalistes ou conservateurs. Ceux-ci doivent d'une part stocker les documents et en assurer la pérennité, et d'autre part en rendre l'accès possible. Avec l'explosion de la quantité d'informations mises à disposition du grand public et des entreprises, notamment au travers d'Internet, l'automatisation du stockage et de la consultation de l'information est devenue un

besoin. Le développement d'outils et de méthodes pour gérer ces quantités d'informations est bien plus qu'un besoin, une nécessité. Ce sont les SRI qui assurent le stockage et permettent la consultation d'informations. 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.

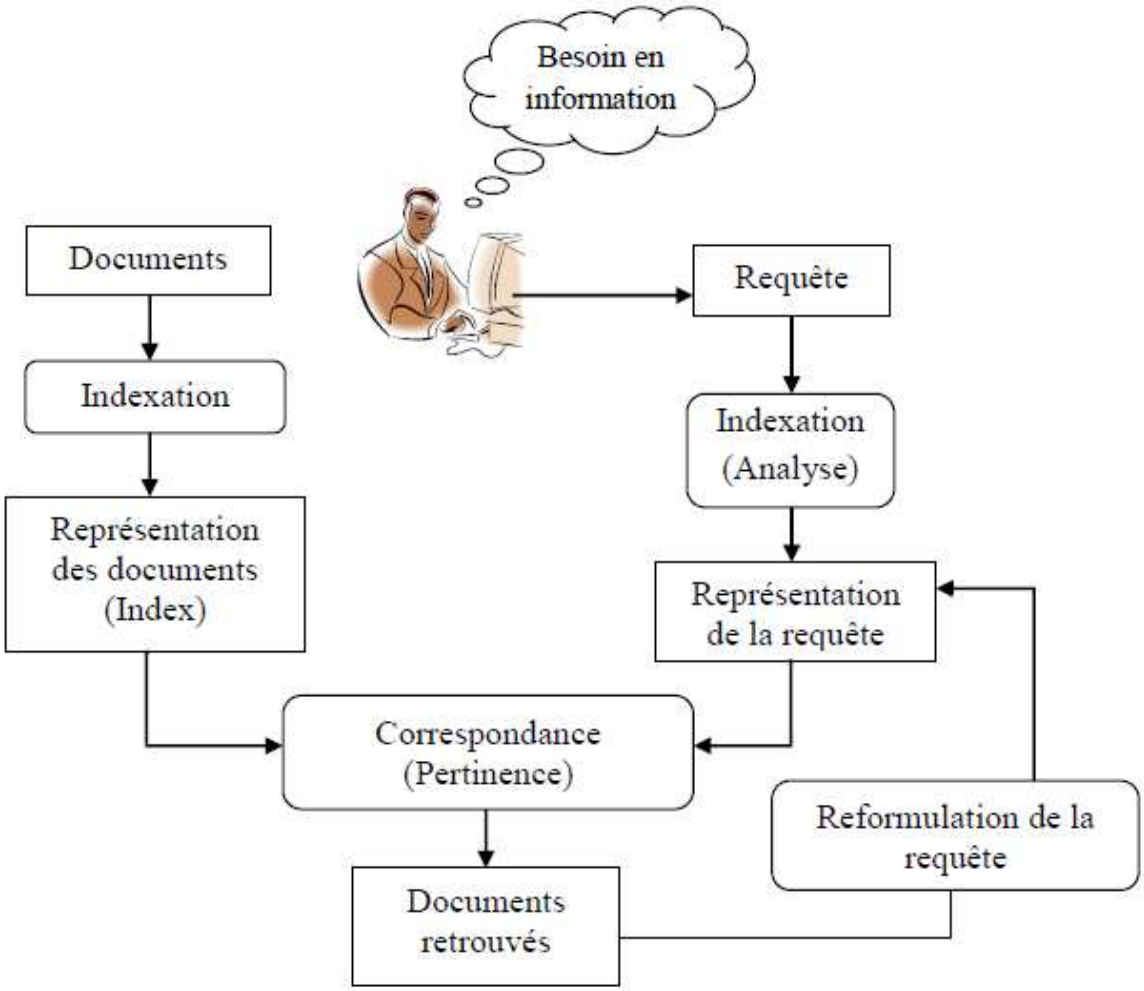

Figure I.1 Architecture générale d'un Système de Recherche d'Information [3].

I.3.3 Concepts de base

I.3.3.1 Besoin en information et Requête

La requête est l'expression du besoin en informations de l'utilisateur. A cet effet, divers types de langages d'interrogation sont proposés dans la littérature. Une requête peut être décrite par: le langage naturel, le langage à base de mot clés ou le langage booléen. Le langage le plus utilisé est le langage naturel. Les requêtes soumises au SRI par les utilisateurs peuvent ne pas refléter leurs besoins en information. Cela est dû, d'une part, au fait que l'utilisateur ignore le fonctionnement interne du SRI, et il n'a qu'une vision restreinte des documents disponibles dans la collection. D'autre part, le SRI n'a souvent aucune connaissance a priori de ses utilisateurs (centres d'intérêts, niveaux, parcours, etc.). Ce biais entre la requête et le besoin en information est une des difficultés majeures de tout système de recherche d'information. Afin de remédier partiellement à ce problème un mécanisme de reformulation de requêtes peut être intégré dans les SRI [3].

I.3.3.2 Document et collection de documents

Le document constitue le potentiel d'informations élémentaire d'une base documentaire. La taille d'un document et son contenu sémantique dépendent en grande partie du domaine d'application considéré. Le document représente l'information élémentaire recherchée par un SRI. Cette information structurée (HTML, XML) ou non structurée (textuelle), peut apparaitre sous plusieurs formes (texte, image, vidéo, son) et dans différents langages (français, anglais, arabe, etc.). L'ensemble des documents sur lequel porte une recherche peut représenter soit tout ou une partie d'un document qui est retournée en réponse à une requête de l'utilisateur [4].

I.3.3.3Pertinence

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].

a) Pertinence Système

Ce type de pertinence est souvent présentée par un score attribué par le SRI a fin d'évaluer l'adéquation du contenu des documents vis-à-vis de celui de la requête. Ce type de pertinence est objectif et déterministe [5].

b) Pertinence utilisateur

La 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.3.4 Les processus d'un SRI

Le SRI s'appui sur trois processus fondamental : l'indexation, la recherche et la reformulation de la requête.

I.3.4.1 Indexation

I.3.4.1.1 Définition

L'indexation est une étape primordiale dans le processus de recherche d'information. Sa qualité dépend en partie de qualité des réponses du système. En effet, les documents et les requêtes dans leur forme texte libre, sont difficiles à exploiter par la machine lors de la recherche. Un traitement préalable permettant leur représentation simplifiée est nécessaire : c'est l'indexation. L'indexation consiste à analyser les documents et les requêtes dans le but d'en définir un ensemble de descripteurs (termes d'index) permettant d'exploiter plus facilement leur contenu lors du processus de recherche. Dans une indexation classique, les termes d'index sont des mots-clés simples ou composés. Ils sont organisés dans une liste de descripteurs, l'index, caractérisant le contenu informationnel d'un document (ou d'une requête). L'ensemble de tous les termes d'index constitue le langage d'indexation [6].

I.3.4.1.2 Classification de l'indexation

L'indexation des documents et requêtes peuvent être réalisée de trois manières distincte : manuelle, semi-automatique et automatique.

Manuelle : Dans ce cas, le document (ou la requête) est analysé par un expert du domaine ou un documentaliste qui se charge d'en représenter le contenu informationnel en utilisant un vocabulaire (ou un langage) contrôlé qui dépend de son savoir propre. Cependant, elle présente les inconvénients suivants :

1) Elle est subjective puisque le choix des termes d'indexation dépend des connaissances des indexeurs dans le domaine (par exemple des termes différents peuvent être affectés à un même document par des indexeurs différents, ou par un même indexeur à des instants différents).

2) Très couteuse à réaliser (en temps et en nombre de personnes impliquées).

3) Difficile à maintenir du fait de l'évolution de la terminologie.

Semi-automatique :la tâche d'indexation est réalisée ici conjointement par un programme informatique et un spécialiste du domaine. Le choix final des descripteurs revient à l'indexeur humain. Dans ce type d'indexation un langage d'indexation contrôlé est généralement utilisé.

Automatique :est un processus d'indexation entièrement automatisé. Il met en œuvre un ensemble de techniques informatisées issues de Traitements Automatiques de la Langue Naturelle (TALN). Ce processus est le plus utilisé en RI, 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.

I.3.4.1.3 Etape d'indexation automatique

Les différentes étapes d'indexation automatique sont discutées ci-dessous :

1) L'analyse lexicale [7] :

L'analyse lexicale est l'étape qui permet de transformer un document textuel en un ensemble de termes (« lexème » est parfois employé). Pendant cette phase, la ponctuation, la casse, et la mise en page sont supprimées.

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 to be).

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. 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.

4) Le choix des descripteurs

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

- **Les mots simples** : les mots simples du texte de document en éliminant les mots vides,
- **Les lemmes** ou les racines des mots extraits.
- **Les 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.

5) La création de l'index

Au terme du processus d'indexation, un ensemble de structure de données sont créés. Ces dernières permettent un accès efficace à la représentation des documents. Le fichier inverse est la structure de données la plus utilisée, 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 qui opère au niveau bit requérant ainsi beaucoup d'opérations pour la compression et la décompression.

D'autres caractéristiques sur un document, permettant de calculer la pertinence a priori d'un document indépendamment de toute requête, peuvent être calculées et stockées a ce stade.

I.3.4.2Correspondance document/requête

Les SRI intègrent un processus de recherche/décision qui permet de sélectionner l'information jugée pertinente pour l'utilisateur. A cet effet, une mesure de similitude (correspondance) entre la requête indexée et les descripteurs des documents de la collection est calculée. Seuls les documents dont la similitude dépasse un seuil prédéfini sont sélectionnés par le SRI. La fonction de correspondance est un élément clé d'un SRI, car la qualité des résultats dépend de l'aptitude du système à calculer une pertinence des documents la plus proche possible du jugement de pertinence de l'utilisateur.

Il existe deux types d'appariement :

- **Appariement exact**

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

- **Appariement approché**

  Le résultat est une liste de documents censés être pertinents pour la requête. Les documents retournés sont triés selon un ordre de mesure. Cet ordre reflète le degré de pertinence document/requête [8].

I.3.4.3 Reformulation de la requête

En RI, l'utilisateur formule son besoin en information par le biais d'une requête dans l'espoir de trouver des réponses pertinentes à ce qu'il recherche. La qualité des réponses dépend d'une

part des termes utilisés par l'utilisateur pour formuler son besoin, et d'autre part des termes utilisés dans l'indexation des documents. Du fait de l'ambiguïté/imprécision de la langue naturelle, l'utilisateur peut formuler sa requête dans un vocabulaire différent de celui utilisé par les auteurs/indexeurs des documents, ce qui a pour conséquence d'influer négativement sur la qualité des résultats de la recherche. Pour résoudre ces problèmes, on introduit le mécanisme de reformulation de requête dans les SRI.

La reformulation de la requête est alors considérée comme un processus ayant pour objectif de générer une nouvelle requête plus ciblée permettant d'obtenir des résultats de recherche plus pertinents que ceux obtenus par la requête initialement formulée par l'utilisateur. La figure I.2, 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.

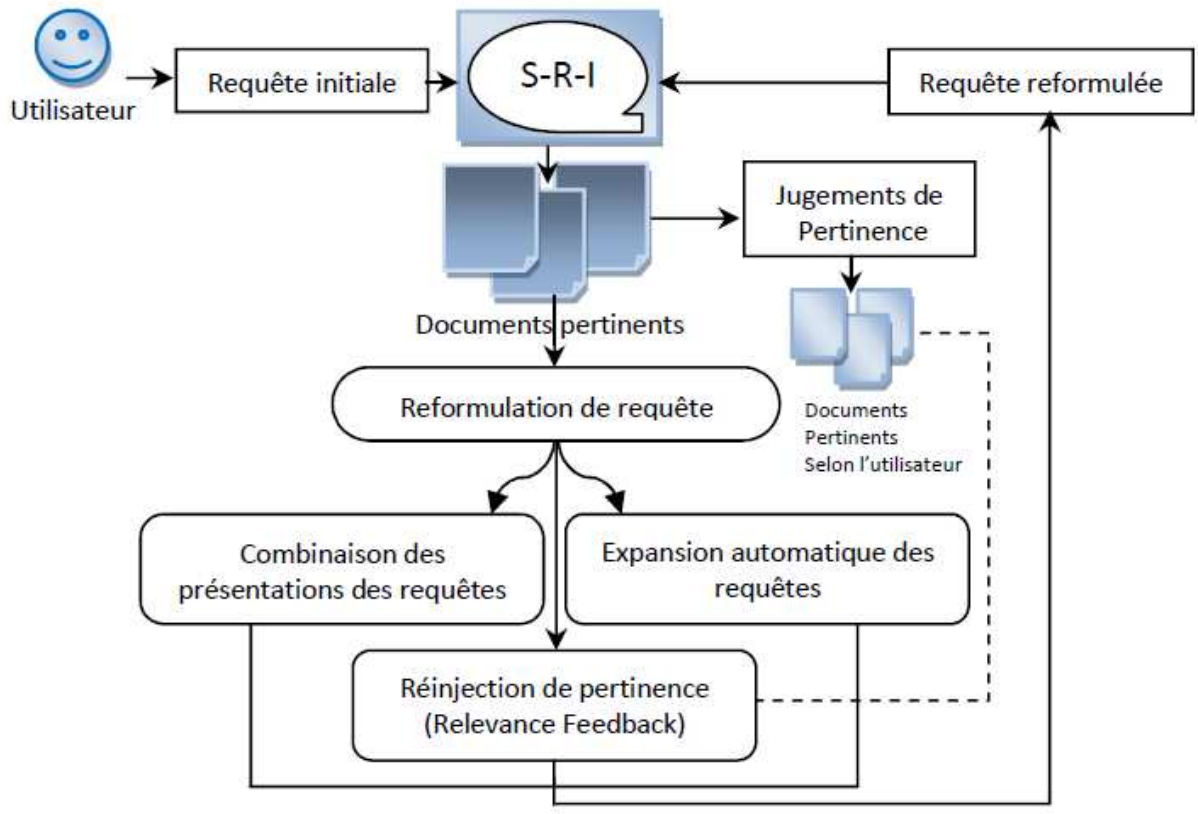

Figure I.2 Techniques d'améliorations des SRI par reformulation de requêtes

I.3.4.3.1 Expansion automatique des requêtes [9]

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. 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.

I.3.4.3.2 La combinaison des présentations de 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 [10]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 [11]. 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 [12]. 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 utilisateur différents, soit des présentations d'une même requête dans des langages différents.

Tamine et al, proposent dans [13] 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.

I.4.3.3 La réinjection de pertinence

Le processus de réinjection de pertinence, comporte principalement trois étapes :
l'échantillonnage, l'extraction des évidences et la réécriture de la requête.

- L'échantillonnage : cette étape permet de construire un échantillon de documents à partir des éléments jugés par l'utilisateur. Cet échantillon est caractérisé par le nombre d'éléments jugés et le nombre d'éléments jugés pertinents.
- L'extraction des évidences est l'étape la plus importante, elle consiste en général à extraire les termes pertinents qui serviront à l'enrichissement de la requête initiale. Plusieurs approches ont été développées, la plus reconnue est celle de Rocchio [14] adaptée au modèle vectoriel.
- La réécriture de la requête consiste à construire une nouvelle requête en combinant la requête initiale avec les informations extraites dans l'étape précédente.

Le processus général de la réinjection de pertinence peut être renouvelé plusieurs fois pour une même séance de recherche : on parle alors de la réinjection de pertinence à itérations multiples.

D'une manière générale, la phase d'échantillonnage ne présente pas de problématique

spécifique. Le seul point abordé à ce niveau concerne le nombre d'éléments à évaluer pour pouvoir effectivement constituer un échantillon représentatif.

La problématique principale de la réinjection de pertinence réside dans les deux autres phases: l'extraction des termes (ils sont alors pondérés pour sélectionner les éléments les plus pertinents) et la réécriture de la requête avec repondération des termes.

Dans la plupart des approches de la littérature, les deux phases sont effectuées avec des méthodes de pondération des termes similaires. Cependant, certaines méthodes et particulièrement celles basées sur le modèle probabiliste, utilisent des méthodes de pondération différentes.

I.4 Les modèles de recherche d'information

La première fonction d'un système de recherche d'information est de mesurer la pertinence d'un document vis-à-vis d'une requête. Un modèle de RI a pour rôle de fournir une formalisation du processus de recherche d'information. Il doit accomplir deux rôles:

1) Créer une représentation interne pour un document ou une requête basée sur les termes de l'indexation.

2) Définir une méthode de comparaison entre une représentation de document et une représentation de requête afin de déterminer leur degré de similarité (mesure de pertinence).

Les trois principales classes ou modèles de recherche d'information sont :

a)Les modèles basées sur la théorie des ensembles : par exemple le modèle booléen, ce sont les plus simple à avoir été mis en place.

b)Les modèles algébriques :comme le modèle vectoriel basé sur les calcules vectoriels.

c)Les modèles probabilistes : qui sont basés sur la théorie des probabilités.

Nous présentons dans ce qui suit les modèles les plus connus :

I.4.1Le modèle booléen

Ce modèle est basé sur la théorie des ensembles, le document est représenté par un ensemble de termes. La requête est représentée par un ensemble de mots clés reliés par des opérateurs booléens (AND, OR et NOT). L'appariement requête document est strict et se base sur des opérations ensemblistes selon les règles suivantes [16]:

$$\begin{aligned}
&\textbf{RSV}(d, t_i) = 1 \ \textbf{SI} \ t_i \in d, 0 \text{ si non} \\
&\textbf{RSV}(d, t_i \text{ AND } t_j) = 1 \ \textbf{SI} \ (t_i \in d) \wedge (t_j \in d), 0 \text{ si non} \\
&\textbf{RSV}(d, t_i \text{ OR } t_j) = 1 \ \textbf{SI} \ (t_i \in d) \vee (t_j \in d), 0 \text{ si non}
\end{aligned} \qquad \text{(I.1)}$$

RSV(d, NOT t_i) = 1 **SI** $t_i \notin$ d,0 si non

Ce modèle présent les avantages suivants :

1) Simplicité de conception du modèle.

2) Possibilité de structurer une requête avec des opérateurs logiques.

Cependant, le modèle booléen a des inconvénients :

1) Difficultés de formulation des requêtes car elles sont complexes (Ambigüité ET/OU).

2) L'absence d'ordre pendant la sélection des documents (dépend de l'ordre des opérateurs).

3) Pas de pondération des termes de la requête.

4) La sélection des documents est basée sur une décision binaire.

I.4.2 Le modèle vectoriel

Ce modèle repose sur des bases mathématiques des espaces vectoriels, il est proposé par Salton en 1971[17]**.**

Dans ce modèle les documents et les requêtes sont représentés sous forme de vecteur dans un espace vectoriel engendré par les N termes d'indexation. Le mécanisme de recherche consiste à retrouver les vecteurs documents qui se rapprochent le plus du vecteur de la requête, et la pertinence document-requête est donnée par la mesure de similarité entre les vecteurs correspondants et ceci nécessite la spécification d'une fonction de calcul (mesure) de similarité entre les vecteurs.

Soient les *N* termes d'indexation $(t_1, t_2,\ldots, t_N)$, les documents et les requêtes sont donc des vecteurs dans un espace vectoriel de dimension *N* :

$\boldsymbol{Dj=(d_{1j}, d_{2j},\ldots, d_{Nj})}$

$\boldsymbol{Q=(q_{1k}, q_{2k},\ldots, q_{Nk})}$

Où $\boldsymbol{d_{ij}}$ et $\boldsymbol{q_{ik}}$ correspondent respectivement aux poids du terme $\boldsymbol{t_i}$ dans le document $\boldsymbol{D_j}$ et dans la requête $\boldsymbol{Q_n}$**.**

La fonction de similarité qui permet de mesurer la ressemblance des documents et de la requête est réalisée en utilisent l'une des mesures présentes dans le tableau suivant[18]

Les mesures	**Les termes de l'espace vectoriels**	**Les formules**
Produit scalaire	$\|X \cap Y\|$	$\sum_{i=1}^{K} X_i Y_i$
Mesure de Dice	$\frac{2\|X \cap Y\|}{\|X\|+\|Y\|}$	$\frac{2\sum_{i=1}^{K} X_i Y_i}{\sum_{i=1}^{K} X_{i2} + \sum_{i=1}^{K} Y_{i2}}$
Mesure de jaccard	$\frac{\|X \cap Y\|}{\|X\|+\|Y\|-\|X\ Y\|}$	$\frac{\sum_{i=1}^{K} X_i Y_i}{\sum_{i=1}^{K} X_{i2} + \sum_{i=1}^{K} Y_{i2} - \sum_{i=1}^{K} X_i Y_i}$
Mesure de cosinus	$\frac{\|X \cap Y\|}{\|X\| \quad \sqrt{\|Y\|} \quad \sqrt{}}$	$\frac{\sum_{i=1}^{K} X_i Y_i}{\sqrt{\sum_{i=1}^{K} X_{i2} + \sum_{i=1}^{K} Y_{i2}}}$

Tableau I.1 Les mesures de similarité du modèle vectoriel.

X ∩ Y : est l'ensemble des termes apparaissant dans l'intersection de la requête et de document.

I.4.3 Le modèle probabiliste

I.4.3.1 Le modèle de base [3]

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 :

$$RSV(q, d) = \frac{P(Per/q, d_i)}{P(NPer/q, d_i)} \quad \text{(I.2)}$$

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 $P(Per \mid q,d_i)$ et $P(NPer \mid q,d_i)$: la probabilité qu'un document d_i 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 :

$$P(Per/q, d_i) = \frac{P(Per/q).P(d_i/Per, q)}{P(d_i)} \quad \text{(I.3)}$$

$$P(NPer/q, d_i) = \frac{P(NPer/q).P(d_i/NPer, q)}{P(d_i)} \quad \text{(I.4)}$$

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 :

$$RSV(q, d) = \frac{P(d_i/Per, q)}{P(d_i/NPer, q)} \quad \text{(I.5)}$$

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

$$P(d_i/Per, q) = \Pi t_j \in d_i P(t_j/Per, q). \Pi t_j \in d_i \ 1 - P(t_j/\not{P}er, q) \quad (I.6)$$

$$P(d_i/NPer, q) = \Pi t_j \in d_i P(t_j/N Per, q) \ . \ \Pi t_j \in d_i \ 1 - P(t_j/\!/ NPer, q) \quad (I.7)$$

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

En posant $p_i = P(t_j|Per, q)$, $q_i = P(t_j|NPer, q)$ et $p_i = q_i$ pour les termes 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 :

$$RSV(d,q) = \sum_{t \in q} log \frac{p_i(1-q_i)}{q_i(1-p_i)} \quad (I.8)$$

Afin de classer les documents avec cette formule, il faut estimer les valeurs des deux probabilités p_i et q_i. En l'absence de collection (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.4.3.2 Le modèle de langage

Par modèle de langue, on désigne une fonction de probabilité P qui assigne une probabilité **P(s)** à un mot ou à une séquence de mots **s=** $m_1 m_2 .. m_n$ dans une langue.

Cette fonction permet d'estimer la probabilité de générer cette séquence de mots à partir du modèle de la langue **:**

$$P(s) = \prod_{i=1}^{n} P(m_i|m_1 \ldots m_{i-1}) \quad (I.9)$$

Lorsque le nombre de mots dans la séquence est élevé, la probabilité de génération devient très faible. On utilise alors un modèle de langue n-gramme (on ne considère que les l prédécesseurs) : $P(m_i|m_1 \ldots m_{i-1})$ devient $P(m_i|m_{i-1+1} \ldots m_{i-1})$.

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 :

$$P(q|M_d) = \pi_{t \in Q} P(t|M_d) \qquad \text{(I.10)}$$

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

$$P(t|M_d) = \frac{tf\,(t,d)}{|d|} \qquad \text{(I.11)}$$

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

Pour remédier au problème posé par les mots de la requête absents dans le document, qui ont pour effet d'avoir la probabilité $P(t|Md)$ 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 [3].

I.5 Evaluation des SRI

Dès l'apparition des premiers SRI, la pratique d'évaluation desdits systèmes est apparue ; les premières évaluations datent de 1953.

L'évaluation des SRI est abordée selon deux angles différents. L'un est dit "paradigme système", qui vise à évaluer les performances du système essentiellement en termes de qualité des documents retournés par le système, c'est-à-dire leur pertinence vis-à-vis des besoins en information des utilisateurs. L'autre est dit "paradigme usager", qui est centré sur la satisfaction de l'utilisateur, et non sur les performances intrinsèques du système, en modélisant le comportement des utilisateurs en situation de recherche.

Nous présentons ci-dessous seulement l'approche basé "système", la plus utilisée dans le domaine de la RI. Elle se base sur deux éléments essentiels à savoir : des mesures d'évaluation et des collections de test**[28]**.

I.5.1Mesures d'évaluation

Pour évaluer les performances des SRI, un certain nombre de mesures standards sont proposées. Ces mesures permettent d'avoir une base homogène d'évaluation. Dans les sections suivantes, nous focalisons notre attention sur ces trois mesures principales: le rappel, la précision, la MAP (Mean Average Precision).

I.5.1.1Le Rappel et la Précision

Le rappel et la précision sont deux mesures de base pour évaluer les performances des systèmes :

a) Le rappel :

Si on suppose que (P) est le nombre de documents pertinents restitués, et que (N) le nombre total de documents pertinents, le rappel est le rapport P sur N exprimé ainsi :

$$rappel = \frac{\mathrm{P}}{\mathrm{N}}$$

b) La précision :

Si on suppose que (P) est le nombre de documents pertinents restitués par le système, et que (R) le nombre total de documents restitués, la précision est le rapport P sur N exprimé ainsi :

$$précision = \frac{\mathrm{P}}{\mathrm{R}}$$

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

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.

1.5.1.2La moyenne des précisions non interpolée (MAP)

La précision moyenne non interpolée (Average Mean 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 :

$$AP_q = \frac{1}{N}\sum_{i=1}^{N} pr(d_i) \qquad (I.12)$$

Avec

$$pr(d_i) = \begin{cases} \frac{r_{n_i}}{n_i} \; si \; d_{ij} \; est \; retrouvé \\ 0 \; sinon \end{cases}$$

Où n_i dénote le rang du document d_i qui a été retrouvé et qui est pertinent pour la requête, r_{ni} est 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 :

$$MAP = \frac{1}{M}\sum_{j=1}^{M} AP_{qj} \qquad (I.13)$$

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

I.5.2 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 sont utilisées pour 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 [29].

I.5.2.1 La collection TREC

TextREtrieval Conférence (TREC) est un programme conçu comme une série d'ateliers dans le domaine de la Recherche d'information (RI). Ce programme est soutenu conjointement par le National Institute of Standards and Technology (NIST) et par l'ARDA (Advanced Research and Development Activity) centre du Département de la Défense des États-Unis. Il a débuté en 1992 dans le cadre du projet TIPSTER. Son but est d'encourager les travaux dans le domaine de la recherche d'information en fournissant l'infrastructure nécessaire à une évaluation objective à grande échelle des méthodologies de recherche textuelle et accroitre la rapidité du transfert de technologie **[30]**.

Parmi, les tâches proposées dans TREC on peut citer : recherche d'information sur le web, recherche d'information médicale, etc. Chaque collection est composée d'un certain nombre de documents, les documents sont codés à l'aide de SGML dans un format spécifique TREC. La figure suivante présente un exemple d'un document TREC :

```
<DOC>
<DOCNO> WSJ920324-0113 </DOCNO>
<DOCID> 920324-0113. </DOCID>
<HL> Venture of Kimbaco</HL>
<DATE> 03/24/92 </DATE>
<SO> WALL STREET JOURNAL (J), PAGE C9 </SO>
<CO> H.TSI </CO>
<MS> FINANCIAL (FIN) </MS>
<IN> ALL BANKS, BANKING NEWS AND ISSUES (BNK)
SECURITIES (SCR) </IN>
<NS> JOINT VENTURES (JVN) </NS>
<RE> FAR EAST (FE)
HONG KONG (HK)
PACIFIC RIM (PRM)
SOUTH KOREA (SK)
</RE>
<LP>
NEW YORK -- South Korean merchant banking firm Kimbaco said it joined
with Hong Kong brokerage house Peregrine Securities to form a new
investment firm Kimbaco Peregrine Capital Ltd.
The firm will seek out cross-border transactions and direct investment
opportunities in Asia, with special emphasis on U.S.-Korean ventures.
</LP>
<TEXT>
</TEXT>
</Doc>
```

Figure I.3 : Exemple d'un document TREC.

Chaque collection TREC a généralement un nombre de requêtes correspondantes entre (50 et 100). Chaque requête TREC est structurée comme suit: un identifiant de requête unique TREC, un titre, une description 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 suivante :

```
<top>
<num> Number: 562
<title> world population growth
<desc> Description:
What is the outlook for world population growth?
<narr> Narrative:
Relevant documents include projections of and
discussion of world population growth. Growth of
individual nations' populations is relevant, but
data on states within the U.S. is not relevant.
</top>
```

FigureI.4 : Exemple d'une requête TREC.

I.6 Conclusion

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

II.1Introduction

Le calcul de la pertinence d'un document vis-à-vis d'une requête utilisateur dépend essentiellement de la pondération des termes. La pondération des termes est l'élément principal dans tout modèle ou processus de recherche d'information [31].

En effet, le processus de pondération doit fournir une représentation compacte et instructive du contenu des documents. Il doit fournir un indicateur d'importance permettant de discriminer les termes les uns vis-à-vis des autres.

Cet indicateur d'importance (poids de termes) est souvent mesuré en utilisant trois statistiques: la fréquence de terme, la fréquence en document de terme et la longueur de document. Ces trois facteurs, on les nomme facteurs classiques.

Ce chapitre a pour but de présenter les différents facteurs pris en compte dans la pondération : facteurs classiques et les facteurs de : la structure de document, la position des termes de la requête dans les documents et la proximité des termes de la requêtes dans les documents.

II.2 Les facteurs de pondération classiques

De manière générale, la majorité des formules de pondération des termes est construite par combinaison de deux facteurs. Un facteur de pondération locale quantifiant la représentativité locale d'un terme dans le document, et un second facteur de pondération globale mesurant la représentativité globale du terme vis-à-vis de la collection des documents.

II.2.1Pondération locale

La pondération locale permet de mesurer l'importance d'un 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 *t*erm *f*requency), elle est exprimée selon l'une des formules suivantes:

–**La fonction brut de tf_{ij}** (term frequency) : correspond au nombre d'occurrences du terme t_i dans le document D_j.

–**La fonction binaire** : elle vaut 1 si la fréquence d'occurrence du terme dans le document est supérieure ou égale à 1, et 0 sinon.

–**La fonction logarithmique** : combine tf_{ij} avec un logarithme, donnée par:

$$\alpha + log\ (tf_{ij}) \qquad \text{(II.1)}$$

Où :

α est une constante. Cette fonction permet d'atténuer l'effet des larges différences entre les fréquences d'occurrence des termes dans le document.

–**Fonction normalisée** : permet de réduire les différences entre les valeurs associées aux termes du document, et elle est donnée comme suit :

$$0{,}5 + 0{,}5 * \frac{tf_{ij}}{max_{ti \in Dj} tf_{ij}} \quad \text{(II.2)}$$

Où :

$max_{ti \in Dj} tf_{ij}$ est la plus grande valeur de tf_{ij} des termes du document D_j.

II.2.2 Pondération globale

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. Un facteur de pondération globale est alors introduit. Ce facteur nommé *idf* (*i*nverted *d*ocument *f*requency), dépend d'une manière inverse de la fréquence en documents du terme et exprimé avec l'une des formules suivantes :

$$idf(t_i) = log\ (\frac{N}{n_i}) \quad \text{(II.3)}$$

$$idf(t_i) = \log\left(\frac{N-n_i}{N}\right) \quad \text{(II.4)}$$

$$idf(t_i) = \log\left(1 + \frac{N}{n_i}\right) \quad \text{(II.5)}$$

Où :

$\boldsymbol{n_i}$ est le nombre de documents où le terme $\mathbf{t_i}$ apparaît dans une collection de documents de taille ***N*.**

II.2.3 La langueur du document

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, elle ne donne pas de bons résultats sur des collections où la taille des documents est hétérogène, car 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, la similarité de ces documents vis-à-vis de la requête augmente.
Pour remédier à ce problème, des travaux ont proposé d'intégrer la taille du document dans les formules de pondération, comme facteur de normalisation [32]**.** Nous présentons dans la section suivante quelques schémas de pondération intégrant la langueur de document.

II.3 Les schémas de pondération classiques

Dans cette section nous décrivons brièvement les approches de pondération qui sont référencées comme les approches de base en RI à savoir, les modèles BM25 et TF-IDF.

II.3.1 Le Schéma de pondération "BM25"

Le schéma de pondération BM25 (BM pour "Best Match") a été développé par Robertson et al. **[33]**, est un schéma de pondération basé sur le modèle probabiliste. Ils utilisent la distribution de probabilité de 2-Poisson. De plus, ils supposent que la longueur d'un document influe directement sur la proportion de termes qu'il emploie. Il s'agit d'une hypothèse similaire à la mesure cosinus du modèle vectoriel (hypothèse de document monothématique). Le score d'un document *d* par rapport à une requête *q* dans la fonction de pondération BM25 est calculée comme suit :

$$RSV_{BM25}(q,d) = \sum_{t \in q \cap d} \left(\frac{tf}{tf + k_1 . n_b} . log \left(\frac{N - df_t + 0{,}5}{df_t + 0{,}5} \right) . qtf \right) \qquad \text{(II.6)}$$

Où :

tf: fréquence d'apparition du terme *t*, dans le document d.

N : nombre total de documents dans la collection,

df_t: nombre de documents contenant le terme *t*,

qtf: fréquence d'apparition du terme *t* dans la requête,

k_1: paramètre d'influence de la fréquence des termes.

n_b: facteur de normalisation, calculé comme suit :

Où : $$n_b = (1 - b) + b \quad \frac{dl}{dl_{avg}}$$

dl: nombre de termes dans le document (longueur de document),

dl_{avg}: Taille moyenne de documents de la collection.

b = 0.75 ; l'augmentation de la valeur de b augmente la pénalisation des documents longs, avec une valeur de 1 étant la limite supérieure.

La complexité apparente de la formule (par rapport à celle du modèle vectoriel) et sa grande efficacité (ce modèle est l'un des plus performants actuellement) montre bien les avantages de la modélisation probabiliste **[34]**.

II.3.2 Le schéma de pondération TF-IDF

Du fait de la double pondération (locale et globale), les fonctions de pondérations sont souvent référencées sous le nom de *Tf-Idf*. Cette technique de pondération a été utilisée au début par le modèle vectoriel classique. Une formule *Tf-Idf* combine donc les deux critères vus précédemment :

– L'importance du terme pour un document (*Tf*),

– Le pouvoir de discrimination de ce terme (*Idf*).

Ainsi, une valeur de *Tf-Idf* élevée pour un terme signifie que ce terme est important dans le document et en plus, il apparaît dans peu de documents de la collection. Avec une telle combinaison, nous présentons ci-dessous quelques formules de *Tf-Idf* proposées [34] :

$$RSV_{tf\text{-}idf} = f(t_i, d_i) * log(N/n_i) \quad \text{(II.7)}$$

$$RSV_{tf\text{-}idf} = (1+log(f(t_i, d_i))) * log(N/n_i) \quad \text{(II.8)}$$

$$RSV_{tf\text{-}idf} = (0.5 + 0.5 * f(t_i, d_i) / Maxtf) * log(N - n_i / n_i) \quad \text{(II.9)}$$

La formule TF-IDF qui intègre la langueur de document est présentée ci-dessous [63] :

$$TF - IDF = tf * idf \quad \text{(II.10)}$$

Où :

$$tf = \frac{K_1 * tf}{(tf + k_1 * \left(1 - b + b * \frac{dl}{dl_{avg}}\right))}$$

$$idf = Idf . log(N / dft + 1)$$

Où: $f(t_i, d_i)$ est la fréquence du terme t_i dans le document d_i .

En plus des trois facteurs de pondération (*tf, idf, taille document*), étudiés précédemment, un autre facteur de pondération, la position du terme dans le document, énoncé par Luhn**[61]** est récemment utilisé sous différents points de vues : la structure de document, la position

chronologique du terme dans le document, et la proximité des termes de la requête dans le document. Nous présentons, ci-dessous ces trois facteurs.

II.4 Les facteurs basés sur la position du terme

II.4.1 Le facteur de la structure de document

À la naissance de la recherche d'information les processus des RI consiste à retrouver a partir des collections de document, les documents qui correspondent le mieux au besoin utilisateur.

Les SRI classique utilisent des collections de documents non structurés, c'est-à-dire uniquement le contenu textuel est utilisé. Les SRI classiques sont souvent développés et utilisés dans des environnements bien contrôlés tel que les bibliothèques, où les collections de documents sont généralement de petites tailles et les utilisateurs ont des besoins en informations bien spécifiques est avec l'apparition du Web (HTML, XML), l'ensemble des documents disponibles est devenu très important. Ces nouveaux types de document ont d'autres caractéristiques tel que la structure interne, et la structure externe (liens entre documents), Il est reconnu que la pertinence de la recherche peut être améliorée en tenant compte de la structure d'un document, Plusieurs moteurs de recherche utilisent les balises de HTML pour améliorer la fonction d'appariement des documents.

Cutler *et al* **[35]** ont mené une étude sur l'apport de la structure des pages HTML pour améliorer la pertinence de la RI. La méthode d'indexation proposée consiste à associer à chaque terme un vecteur de fréquence noté *tfv*, qui contient la fréquence d'apparition du terme dans une des classes de balises. Six classes de balises ont été utilisées : Anchor, <H1>- <H2>, <H3>-<H6>, STRONG, TITLE et le texte plein (toutes les autres balises).

Lors de la recherche un autre vecteur à six éléments noté *CIV* (Class Importance Vector) est utilisé, chaque élément de ce vecteur représente un facteur d'importance associé pc (II.11)
classe de balises. L'importance (poids : w) d'un terme dans un document est alors calculée comme suit :

$$w = (tf * CIV) * idf$$

Ainsi, la traditionnelle formule de pondération *tf* est étendue comme suit : *tf * CIV*, qui tient compte de la fréquence du terme dans une classe et de l'importance accordée à cette classe. Les expérimentations menées ont montré que, l'usage de la structure des pages HTML améliore sensiblement la pertinence de la RI [62].

La RI est adaptée à un autre format de données qui le XML. le langage XML est maintenant largement utilisé, en particulier pour les référentiels de données scientifiques, les bibliothèques numériques et sur le Web. De nombreux systèmes sophistiqués ont été proposées **[37, 38, 39, 40, 41] [42, 43, 44, 45, 46]**.

Les documents XML contiennent souvent des sous-champs (éléments), par exemple les collections INEX de IEEE contiennent des domaines tels que le titre, abs, bdy, bm, st etc. Les chercheur ont jugé utile d'exploiter la structure interne de document pour améliorer la performance au niveaux de l'élément **[47]**, la formule ci-dessous permet la recherche au niveau des éléments:

$$\mathbf{w}f_j(e,\overline{d},C) = \frac{(k'_1+1)tf'_{e,j}}{k'_1((1-b)+b\frac{el'}{avel'})+tf_{e,J}} \log\frac{N-df_j+0.5}{df_j+0.5} \qquad \text{(II.12)}$$

Où :

$tf'_{e,j}$ désigne la fréquence de terme pondérée de j-ième terme t dans e.

el' est la longueur de l'élément pondéré.

$avel'$ est la longueur moyenne de l'élément dans la collection.

k'_1 paramètre d'influence de la fréquence des termes

II.4.2 Le facteur de la position de terme de requête

A.D. Troy et al **[50]** ont introduit un modèle nommé le CTR, (position chronologique d'un terme dans un document). Il est définie comme le rang du terme dans la séquence de mots du document. Intuitivement, cette technique utilise la position chronologique d'un terme (CTR) qui capture les positions des termes tels qu'ils apparaissent, dans l'ordre, dans un document. L'intuition de ce modèle vient de fait que dans les écrits journalistiques ou contributions scientifiques (articles scientifiques) les auteurs placent les termes importants et pertinents tout au début de leur article puis progressivement, ils incluent les termes les moins importants.

Le auteurs de ce modèle ont évalué l'apport de diverses combinaisons de CTR avec le modèle Okapi BM25 afin d'identifier la formule la plus efficace.

Le facteur de chronologique (R) est soit additionné à la fréquence ou multiplié par la fréquence.

Les formules utilisées et les combinaisons de ce facteur sont données dans le tableau suivant :

.

Base Formulae		
	FORMULA	DESCRIPTION
A	$\sum_{t \in d \cap q} ln \frac{N - df + 0.5}{df + 0.5} \cdot \frac{tf}{0.5 + 1.5 \cdot \frac{dl}{avdl} + tf} \cdot \mathcal{R}$	Multiplicative
B	$\sum_{t \in d \cap q} ln \frac{N - df + 0.5}{df + 0.5} \cdot \left(\frac{tf}{0.5 + 1.5 \cdot \frac{dl}{avdl} + tf} + \mathcal{R} \right)$	Additive
$\mathcal{R}$		
	FORMULA	DESCRIPTION
a	$C \cdot (1 - \frac{tr - 1}{dl})$	Rank Percentage
b	$1 + (C \cdot (1 - \frac{tr - 1}{dl}))$	Rank Percentage Multiplicative Boost
c	$(1 - C) + (C \cdot (1 - \frac{tr - 1}{dl}))$	Rank Percentage Multiplicative Scale
d	$C \cdot \frac{1}{tr}$	Inverse Rank
e	$C \cdot \frac{1}{log(tr)}$	Inverse Log Rank
f	$C \cdot (1 - \frac{tr - 1}{maxdl})$	Rank Maximum Percentage
g	$C \cdot (1 - \frac{log(tr + 9)}{log(dl + 10)})$	Rank Log Percentage
h	$C \cdot (1 - \frac{log((tr - 1/D) + 10)}{log((dl/D) + 10)})$	Scaled Rank Log Percentage
i	$C \cdot (1 - \frac{log((tr - 1/D) + 10)}{log((maxdl/D) + 10)})$	Scaled Rank Log Maximum Percentage
j	$C - (C \cdot D \cdot \frac{log((tr - 1/30) + 10)}{log((dl/30) + 10)})$	Range Limited Scaled Rank Log Percentage

Tableau II.1: Les formules utilise dans le facteur chronologique.

D'après les résultats des expérimentations des auteurs de ce modèle **[50]**, la formule la plus efficace est la formule j.

II.4.3Les facteurs de proximité

La plupart des SRI existants représentent les documents comme un ensemble de mots clés, ce que l'on appelle communément une représentation par sac de mots. Ces mots clés sont généralement pondérés en utilisant des schémas de pondération tels que *tf * idf, BM25* ou le modèle de langue uni-gramme. Tous ces modèles supposent que les mots clés sont indépendants. Cette hypothèse d'indépendance entre termes facilite grandement les calculs. De ce fait, l'ordre des termes dans une phrase est donc ignoré. Ceci peut de toute évidence conduire à l'ambigüité entre termes, qui pourraient engendrer des résultats contenant beaucoup de documents non pertinents. Il est par conséquent nécessaire de développer des modèles de recherche d'information allant au de là de la représentation avec une liste de mots clés. Parmi les pistes les plus investies on trouve deux modèle: le modèle MRF **[51]** et le modèle de langue de position **[52].**

II.4.3.1 Le modèle MRF

Metzler et Croft **[51]** ont élaboré un cadre formel pour la modélisation des dépendances entre termes en utilisant les champs de Markov, nommé (MRF). Une structure de graphe non orienté est utilisée pour modéliser les distributions jointes. Dans ce cadre, ils ont proposé de modéliser deux types de dépendance: dépendance séquentielle (SD : Sequential Dependency), capturant les relations entre les paires de termes adjacents de la requête, et la dépendance complète (FD : Full Dependency), capturant les relations entre toutes les paires de termes de la requête. Ces deux modèles de dépendance ont été interpolés linéairement avec un modèle uni-gramme, selon la formule suivante:

$$\begin{aligned} \text{Score}_{MRF}(q, d) = \lambda_u & \sum_{t_i \in q} log(P(q_i | d)) \\ + \lambda_s & \sum_{t_i \in q} \sum_{\substack{t_i \in q \\ j=i+1}} log(P(<q_i, q_j>_w | d)) \\ + \lambda_f & \sum_{t_i \in q} \sum_{\substack{t_i \in q \\ j \neq i}} log(P(<q_i, q_j>_w | d)) \end{aligned} \quad \text{(II.13)}$$

Où les paramètres λ_u, λ_s et λ_f permettent de contrôler respectivement, le poids du modèle uni-gramme, du modèle de dépendance séquentielle et du modèle de dépendance complète et le paramètre w définit la longueur de la fenêtre du texte permettant de compter les occurrences de la paire $<q_i, q_j>$ dans le document *d*.

Notre travail consiste à étendre ce modèle (MRF). Nous présentons ce modèle, en détail, ainsi que l'extension proposée dans le chapitre suivant.

II.4.3.2 Le modèle de longue de position

Lv and Zhai **[52]** ont proposé un modèle de langue nommé « Positional Language Model : PLM ». Dans ce dernier, un modèle de langue pour chaque position du document est créé, il est exprimé ainsi :

$$P(t \mid d, i) = \frac{c'(t; i)}{\sum_{t' \in V} c'(t'; i)} \qquad \text{(II.14)}$$

Où V est le vocabulaire et $c'(t; i)$ est la fréquence virtuelle du terme *t* à la position *i* obtenue par la propagation de l'ensemble des occurrences du terme *t* dans toutes les positions du document. cette propagation est réalisé en utilisant des fonction de densité (Gaussian kernel, Triangle kernel, Circle kernel, Hamming kernel).

Afin de calculer le score d'un document vis-à-vis d'une requête, ils utilisent les scores obtenus sur les différentes positions du document. Différentes stratégies de combinaison de scores ont été utilisées : la stratégie de la meilleure position, la stratégie multi-position et la stratégie Multi-σ.

Le modèle ainsi défini a été expérimenté sur plusieurs collections de test TREC. Les auteurs ont rapporté que ce modèle améliore les résultats du modèle de Tao et Zhai **[53]**.

Les approches utilisant la notion de proximité pour intégrer les relations entre termes ont montré que cette source d'information est utile pour la RI. Cependant, le principal problème de ces approches est l'absence d'une mesure bien reconnue pour le calcul de la proximité entre termes **[54]**.

II.5 Conclusion

Dans ce chapitre nous avons décrit les facteurs classiques de pondération. Nous avons, particulièrement, introduit le facteur local (TF), le facteur global et la longueur de document. Nous avons aussi passé en revue les schémas de pondération classiques, à savoir BM25, et TF-IDF. Nous avons étudié d'autres facteurs de pondération, telles que le facteur de la structure de document, le facteur de la position des termes de la requête dans le document et le facteur de proximité, notamment deux modèles sont présentés : le modèle de langue de position et le modèle MRF.

Ce derniers modèle va être décrit en détail ainsi qu'une extension de ce modèle dans le chapitre suivant.

III.1 Introduction

Après avoir présenté dans le chapitre précédent les facteurs de pondération. où nous avons noté l'utilisation d'un nouveau facteur qui est la position des termes dans le document, selon différents points de vues. Notamment, la proximité des termes de la requête dans le document. L'objectif de notre étude et d'étendre un des modèles utilisant ce point de vue; ce modèle est le modèle MRF.

Ainsi, dans ce chapitre notre objectif est de présenter en détail ce modèle et une extension basée sur le facteur de couverture spatiale. Puis nous présentons les outils de développement utilisés, et enfin quelques résultats obtenus avec le modèle MRF de base et le modèle MRF étendu. Les expérimentations sont effectuées sur la collection de test TREC AP88.

III.2 Présentation du modèle MRF

Nous présentons dans cette section le modèle MRF, précisément nous décrivons l'intuition et la formalisation de ce modèle.

III.2.1 Intuition du modèle MRF

La plupart des SRI existants représentent les documents comme un ensemble de mots clés, ce que l'on appelle communément une représentation par sac de mots. Ces mots clés sont généralement pondérés en utilisant des schémas de pondération tels que *tf*idf*, *BM25* ou le modèle de langue uni-gramme qui prennent en compte les statistiques suivantes : la fréquence du terme dans le document (*tf*), sa fréquence dans la collection (*idf*) et la taille du document. Tous ces modèles supposent que les mots clés sont indépendants. Cette hypothèse d'indépendance entre termes facilite grandement les calculs. De ce fait, l'ordre des termes dans une phrase est donc ignoré. Ceci peut de toute évidence conduire à l'ambigüité entre termes, qui pourraient engendrer des résultats contenant beaucoup de documents non pertinents. A titre d'exemple, prenons la requête «recherche d'information », avec la représentation en sac de mots, un document comportant dans une partie le mot « recherche » et dans une autre partie le rôle de « l'information dans le développement d'une entreprise », serait présenté à l'utilisateur comme pertinent. Or, on voit bien que ce document n'est pas pertinent car il ne traite pas de la « recherche d'information », dans ce cas-là les termes de la requête sont indépendant, plusieurs travaux ont été réalisés pour outrepasser cette hypothèse d'indépendance entre termes. Parmi les travaux les plus aboutis on trouve le modèle MRF **[51]**.

III.2.2Formalisation du modèle MRF

L'approche de champ de Markov a pour objectif la prise en compte de la dépendance entre termes. Elle est également appelée modèle graphique non orienté. Cette approche est couramment utilisée dans le domaine de l'apprentissage automatique. Cette approche est utilisée en RI pour modéliser les dépendances entre termes, notamment: la dépendance séquentielle et dépendance complète, comme le montre les graphes de la figure III.1.

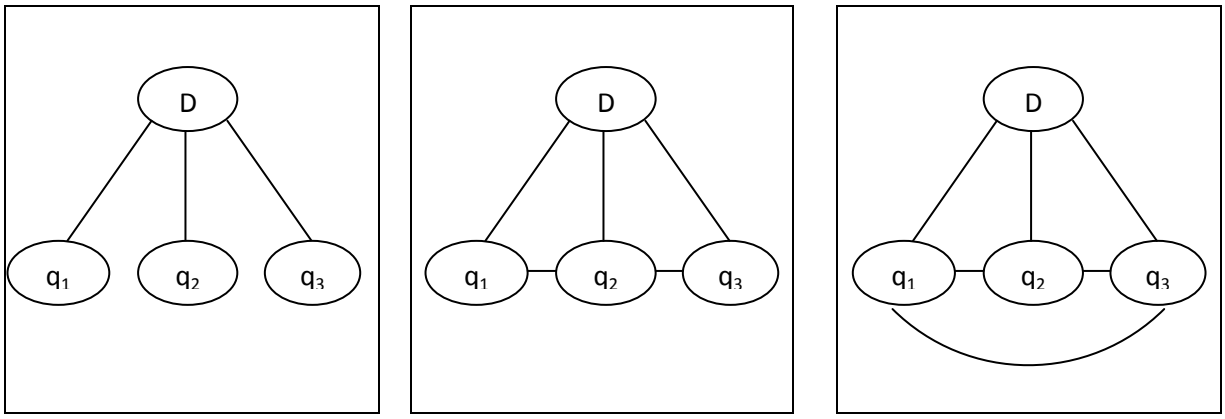

Figure III.1: Trois types de dépendances entre termes.

Ces graphes comportent des nœuds, le nœud D représente le document et les nœuds q_1, q_2 et q_3 représentent les termes de la requête. Le graphe de gauche de la figure III.1 illustre La représentation en sac de mot (indépendance entre termes), le graphe au centre présente la dépendance séquentielle, et le graphe de droite présente dépendance complète.

Les trois représentation sont interpolées linéairement selon la formule suivante:

$$Score_{MRF}(q,d) =$$

Où les paramètres λ_u, λ_s et λ_f permettent de contrôler respectivement, le poids du modèle uni-gramme, du modèle de dépendance séquentielle et du modèle de dépendance complète.

il est a noter que : $\lambda_T + \lambda_O + \lambda_U = 1$. Le paramètre w définit la longueur de la fenêtre du texte permettant de compter les occurrences de la paire $<q_i, q_j>$ dans le document *d*.

III.3 Présentation de l'extension de MRF

Comme nous l'avons cité plus haut le modèle MRF a trois composantes : le modèle uni-gramme, la composante séquentielle et la composante de dépendance entière.

Nous avons constaté que la première composante prend en compte uniquement la fréquence d'apparition du terme de la requête dans le document. Cependant, la fréquence à elle seule ne reflète pas l'importance du terme dans le document.

Nous proposons un nouveau facteur, nommé "couverture spatiale". Par ce dernier nous entendons qu'un terme qui apparait dans beaucoup de parties d'un document est meilleur qu'un autre qui apparait dans peu de parties, même si ce dernier est plus fréquent que le premier. Pour bien expliquer cette notion de couverture, prenant l'exemple suivant :

soit deux documents d1 et d2, contenant les termes dans l'ordre indiqué.

d1 = t1 t2 t3 t4 | t4 t5 t1 t7 | t9 t8 t10 t1 | t4 t1 t2 t5

P1 P2 P3 P4

d2 = t7 t1 t2 t1 t1 t8 t1 t9 | t10 t4 t3 t7 | t8 t11 t13 t4

P1 P2 P3 P4

Où tf (t1, d1) = 4 ; tf (t1, d2) = 4 ; | partie | = 4.

Nous pouvons remarquer que le terme t1 offre une meilleure couverture spatiale au document " d1 ", car il couvre toutes les parties (4 parties) de ce document que pour le document " d2 " pour lequel il couvre que deux parties sur quatre, et même si le terme t1 à la même fréquence dans les deux documents il doit avoir une plus grande importance dans le document "d1" que dans le document "d2".

III.3.1 Présentation du facteur de couverture spatiale

Pour formaliser notre idée de " couverture spatiale " nous proposons un ensemble de formules qui calculent ce nouveau facteur. Ces formules sont présentées dans le tableau suivant :

	Formule de calcul de couverture spatiale	Description
1	$\alpha \times f$	$f = nb_{occw}/nb_w;\ nb_w = \frac{\lvert D\rvert}{\lvert w\rvert}$; nb_{occw} c'est le nombre de fenêtres (parties) ou le terme est apparait et $\lvert w\rvert$ c'est la taille de la fenêtre.
2	$\alpha \times f$	$f = \frac{nb_{occw}}{nb_w};\ nb_w = \frac{\boldsymbol{AVLD(C)}}{\lvert w\rvert}$; $AVLD(C)$ *est la taille moyenne de la fênetre*
3	$\beta + (\alpha + (1 - \alpha) \times f)$	
4	$log(\beta + (\alpha + (1 - \alpha) \times f))$	

Tableau III.1 Formules de calcul de la couverture spatiale.

III.4 Les outils utilisés

Dans ce qui suit, nous allons présenter notre environnement technique et spécifier les différents outils utilisés, nous commençons par la plateforme Terrier sous laquelle nous avons implémenté notre approche, puis JAVA que nous avons utilisé comme langage de programmation et Netbeans qui est l'outil sur lequel nous avons programmé.

III.4.1 Présentation de la plate forme 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 éprouvée. 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 plateforme 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.

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, etc.) ainsi que les scores des documents. Le résultat renvoyé à l'utilisateur, est la liste des documents jugés pertinents et leurs scores respectifs **[58]**.

La figure III.3 illustre l'architecteur de Terrier.

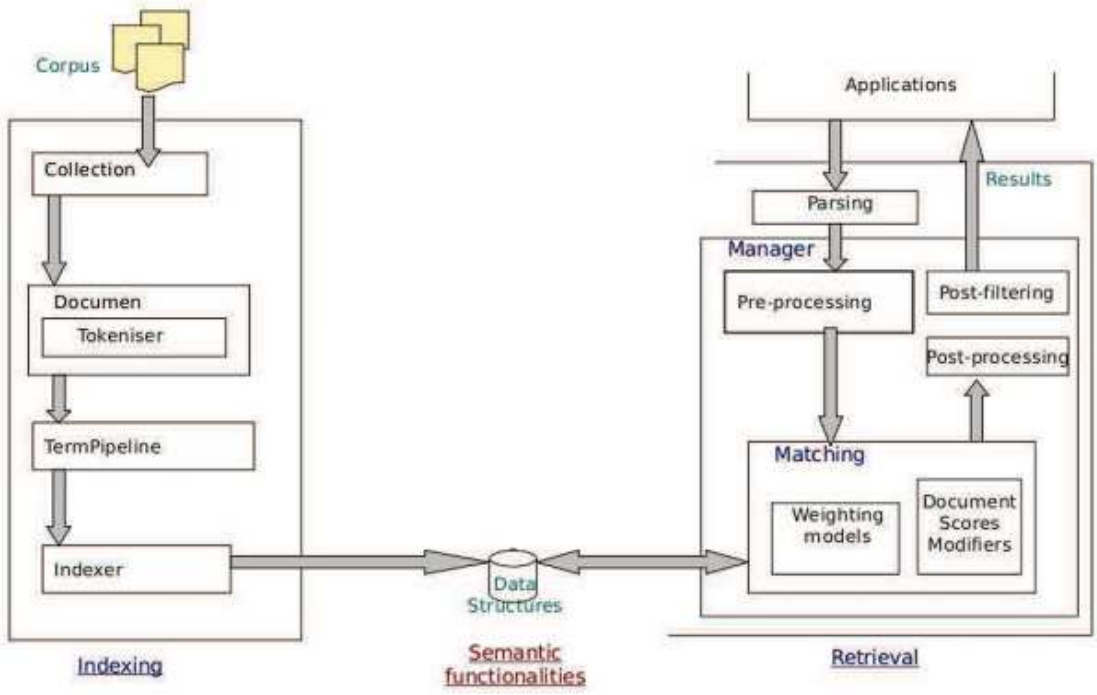

Figure III.2 Architecteur de Terrier

III.4.2 Le langage Java

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

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'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications.

Pour cela, divers plateformes et frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées en Java. Le langage Java reprend en grande partie la syntaxe du langage C++, très utilisée par les informaticiens. Néanmoins, Java a 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. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.).

Java permet de développer des applications client-serveur. Côté client, les applets sont à l'origine de la notoriété du langage. C'est surtout côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des applets.

Java a donné naissance à un système d'exploitation (JavaOS), à des environnements de développement (eclipse/JDK), des machines virtuelles (MSJVM **(en)**, JRE) applicatives multi plate-forme (JVM), une déclinaison pour les périphériques mobiles/embarqués (J2ME), une bibliothèque de conception d'interface graphique (AWT/Swing), des applications lourdes (Jude, Oracle SQL Worksheet, etc.), des technologies web (servlets, applets) et une déclinaison pour l'entreprise (J2EE). La portabilité du bytecode Java est assurée par la machine virtuelle Java, et éventuellement par des bibliothèques standard incluses dans un JRE. Cette machine virtuelle peut interpréter le bytecode ou le compiler à la volée en langage machine. La portabilité est dépendante de la qualité de portage des JVM sur chaque système d'exploitation **[59]**.

III.4.3 L'environnement de développement (NetBeans)

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, XML, Groovy, PHP et HTML de façon native ainsi que bien d'autres (comme Python ou Ruby) par l'ajout de greffons. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web). Conçu en Java, NetBeans est disponible Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java. NetBeans constitue par ailleurs une plate forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate forme.

NetBeans est aussi une plate-forme générique pour le développement d'applications pour stations de travail (bibliothèque Swing (Java)).Elle fournit des ressources pour développer les éléments structurants de ces applications: gestion des menus, des fenêtres, configuration, gestion des fichiers, gestion des mises à jour **[60]**.

La figure ci-dessous présente un aperçu de l'interface de l'IDE NetBeans 8.0.2.

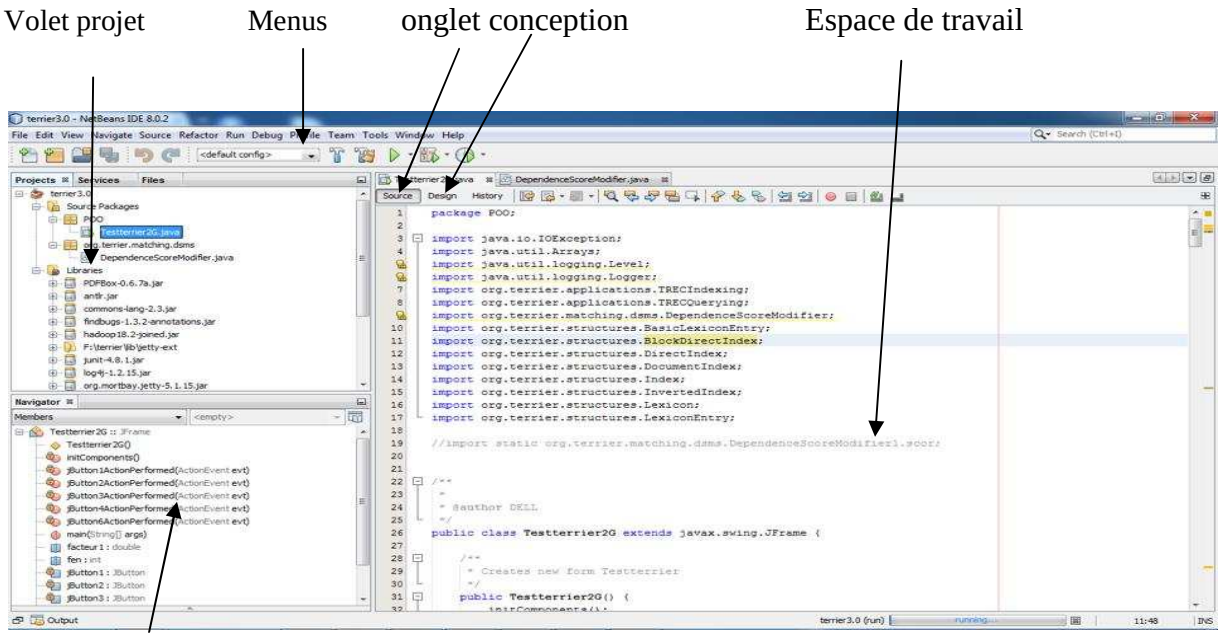

Figure III.3 L'environnement de développement NetBeans.

III.5 Interface de l'application

L'application développée à l'interface suivante, elle contient quatre boutons, chacun effectue une tâche déterminée, nous présentons le fonctionnement de chaque bouton dans ce qui suit :

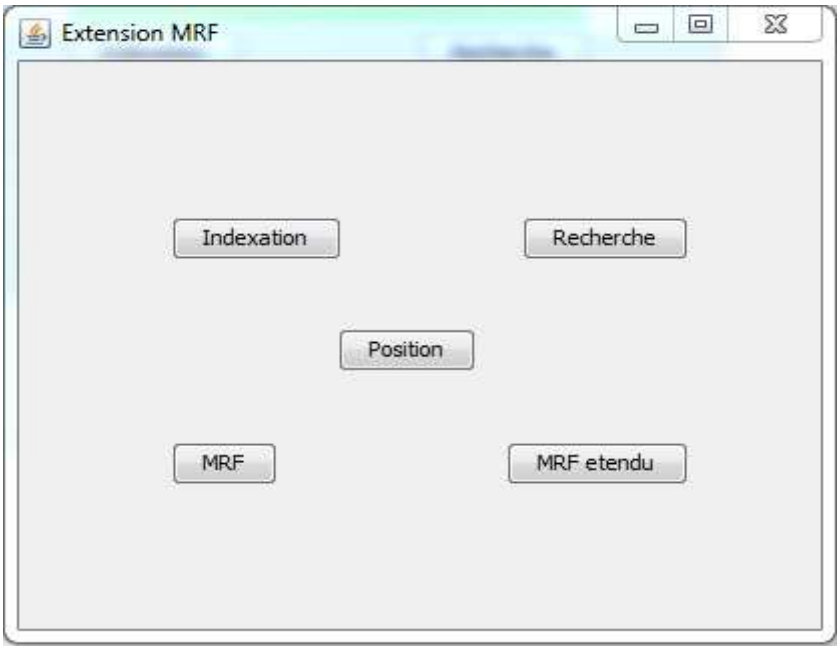

Figure III.4 Interface de l'application

a. Bouton Indexation (Création de l'index) :

Après la configuration des propriétés comme le chemin de l'index et du fichier "collection.spec" qui contient les adresses des documents de la collection. Nous créerons l'index en cliquant sur ce bouton.

b. Bouton Recherche (La recherche):

Après la spécification des différentes propriétés comme le modèle de pondération, dans notre cas nous avons utilisé le modèle (BM25), nous effectuons la recherche en utilisant :

- La class TRECQuerying : qui initialise l'index inversé, le lexique et les structures d'index du document.
- La méthode processQueries():qui appartient à la class TRECQuerying, elle exécute la mise en correspondance en utilisant le modèle de pondération spécifié.

c. Bouton MRF

Il fait appelle à la classe ***DependenceScoreModifier*** qui modifie le score des 1000 premiers document retournés par la première recherche (BM25).

d. Bouton MFR étendu

Nous illustrons le traitement effectué lorsqu'on clique sur ce bouton par un algorithme.

Entrées : docids [ ] : vecteur qui contient les identifiant des documents retournés par une première recherche.

terms [ ] : vecteur qui contient les termes de la requêtes.

score [ ] : vecteur qui contient les scores des documents retournés.

Sortie : score [ ] : vecteur qui contient les nouveaux scores des documents.

Début

pour (d=0 jusqu'au |docids| -1)

On prend le $d^{ème}$ document et en récupère la matrice ***termes[][]*** de document.

où on aura tout les identifiant des termes, leurs fréquences et leurs positions.

pour (i=0 jusqu'à | termes[] | - 1)

On prend le $i^{ème}$ terme

pour (j=0 jusqu'à la taille des termes de document)

On prend le $j^{ème}$ terme et en le compare avec $i^{ème}$ terme de la requête

Si (les deux termes sont égaux)

On récupère sa fréquence ainsi que ses positions (dans tf et position[]).

On initialise (F) la taille de la fenêtre qui prends les valeurs 10, 25, 50.

On calcul nb_fen de document qui égale à(taille de Doc/F)

On initialise le nombre de fenêtres occupées par le terme à 1(nbfenocc=1).

Si (on a plusieurs positions de terme dans le Doc (position[] >1))

pour (k=0 jusqu'à taille des positions de terme dans le document)

Si ((position[k+1] - position[k]) > taille de fenêtre)

nbfenocc ++;

Fin pour

// On calcule la couverture spatiale du terme.

facteur = nb fenêtres occupées par le terme / nb de fenêtres total

score [i] = score [i] +formule (facteur) // recalculer le score.

// formule est l'une des formule citées dans **Tableau III.1.**

Fin Si

Fin Si

Fin pour

Fin pour

Fin

III.6 Résultats et expérimentations

Dans cette section nous décrivons d'abord la collection de test utilisée ensuite nous présentons les résultats obtenus.

III.6.1 Collection de test utilisée

Nous avons mené nos expérimentations en utilisant la collection de test TREC AP88 (Associated Press 1988) décrite dans le tableau III.2.

Nous avons utilisé 100 requêtes (51-151) de la collection TREC, où le champ titre est seulement pris en compte.

AP88		
Nombre de documents dans la collection	**Nombre de termes dans la collection**	**Taille moyenne d'un documents**
79919	**144186**	**235.085**

Tableau III.2 Description de la collection de test utilisée

III.7 Evaluations et résultats

Une comparaison entre les résultats obtenus avec la méthode MRF et ceux de notre approche (MRF étendu) est nécessaire pour pouvoir évaluer notre approche. Nous utilisons la mesure MAP (Moyenne average Précision présentée dans le premier chapitre) pour l'évaluation de ces résultats.

Nous commençons par présenter les résultats obtenus par la recherche simple avec le modèle de pondération BM25.

III.7.1 Résultat obtenu avec la recherche simple

Le résultat de l'évaluation de la recherche simple basée sur le modèle de pondération BM25 avec Le MAP est illustré dans le tableau suivant :

Recherche simple	
Modèle de recherche	MAP (Moyenne average Précision)
BM25	0.1296

Tableau III.3 Résultat obtenu avec la recherche simple (BM25).

III.7.2 Résultats obtenus avec le modèle MRF

Nous avons utilisé la dépendance séquentielle du modèle MRF. Pour evaluer ce modèle nous avons procédé comme suit : nous avons varié la taille de la fenêtre, **ngram.length**, qui prend les valeurs 2, 5, 10, 15 et 20 et pour chacune des valeurs, nous avons varié les deux paramètres (poids) **proximity.w_o**, **proximity.w_u** qui prends leurs valeurs entre 0.1 et 1.0. avec un pas de 0.1.

Les tableaux suivants présentes les résultats obtenus:

proximity.w_o	proximity.w_u	proximity.ngram.length= 2	proximity.ngram.length=5
0.1	0.1	0.1301	0.1315
0.1	0.2	0.1312	0.1338
0.1	0.3	0.1318	0.1345
0.1	0.4	0.1324	0.1358
0.1	0.5	0.1327	0.1366
0.1	0.6	0.1333	0.1379
0.1	0.7	0.1341	0.1394
0.1	0.8	0.1345	0.1400
0.1	0.9	0.1350	0.1412
0.1	1	0.1351	0.1429
0.2	0.1	0.1312	0.1338
0.2	0.2	0.1324	0.1358
0.2	0.3	0.1333	0.1379

0.2	0.4	0.1345	0.1400
0.2	0.5	0.1351	0.1429
0.2	0.6	0.1359	0.1446
0.2	0.7	0.1369	0.1464
0.2	0.8	0.1376	0.1480
0.2	0.9	0.1382	0.1486
0.2	1	0.1388	0.1499
0.3	0.1	0.1318	0.1345
0.3	0.2	0.1333	0.1379
0.3	0.3	0.1350	0.1412
0.3	0.4	0.1359	0.1446
0.3	0.5	0.1369	0.1470
0.3	0.6	0.1382	0.1486
0.3	0.7	0.1394	0.1503
0.3	0.8	0.1408	0.1520
0.3	0.9	0.1415	0.1530
0.3	1	0.1421	0.1539
0.4	0.1	0.1324	0.1358
0.4	0.2	0.1345	0.1400
0.4	0.3	0.1359	0.1446
0.4	0.4	0.1376	0.1480
0.4	0.5	0.1388	0.1499
0.4	0.6	0.1408	0.1520
0.4	0.7	0.1414	0.1535
0.4	0.8	0.1425	0.1547
0.4	0.9	0.1429	0.1553
0.4	1	0.1436	0.1560
0.5	0.1	0.1327	0.1366
0.5	0.2	0.1351	0.1429
0.5	0.3	0.1369	0.1470
0.5	0.4	0.1388	0.1499
0.5	0.5	0.1411	0.1521
0.5	0.6	0.1421	0.1539
0.5	0.7	0.1428	0.1552
0.5	0.8	0.1436	0.1560
0.5	0.9	0.1457	0.1562
0.5	1	0.1462	0.1574
0.6	0.1	0.1333	0.1379
0.6	0.2	0.1359	0.1446
0.6	0.3	0.1382	0.1486
0.6	0.4	0.1408	0.1520
0.6	0.5	0.1421	0.1539
0.6	0.6	0.1429	0.1553
0.6	0.7	0.1452	0.1564
0.6	0.8	0.1460	0.1573
0.6	0.9	0.1464	0.1573
0.6	1	0.1468	0.1572
0.7	0.1	0.1341	0.1394
0.7	0.2	0.1369	0.1464
0.7	0.3	0.1394	0.1503
0.7	0.4	0.1414	0.1535
0.7	0.5	0.1428	0.1552
0.7	0.6	0.1452	0.1564
0.7	0.7	0.1461	0.1573
0.7	0.8	0.1466	0.1572

0.7	0.9	0.1468	0.1572
0.7	1	0.1463	0.1574
0.8	0.1	0.1345	0.1400
0.8	0.2	0.1376	0.1480
0.8	0.3	0.1408	0.1520
0.8	0.4	0.1425	0.1547
0.8	0.5	0.1436	0.1560
0.8	0.6	0.1460	0.1573
0.8	0.7	0.1466	0.1572
0.8	0.8	0.1467	0.1573
0.8	0.9	0.1460	0.1574
0.8	1	0.1457	0.1574
0.9	0.1	0.1350	0.1412
0.9	0.2	0.1382	0.1486
0.9	0.3	0.1415	0.1530
0.9	0.4	0.1429	0.1553
0.9	0.5	0.1457	0.1562
0.9	0.6	0.1464	0.1573
0.9	0.7	0.1468	0.1572
0.9	0.8	0.1460	0.1574
0.9	0.9	0.1458	0.1573
0.9	1	0.1447	0.1571
1	0.1	0.1351	0.1429
1	0.2	0.1388	0.1499
1	0.3	0.1421	0.1539
1	0.4	0.1436	0.1560
1	0.5	0.1462	0.1574
1	0.6	0.1468	0.1572
1	0.7	0.1463	0.1574
1	0.8	0.1457	0.1574
1	0.9	0.1447	0.1571
1	1	0.1434	0.1570

Tableau III.4 Résultats obtenus avec le modèle MRF (pour **ngram.length = 2 et 5**).

proximity.w_o	proximity.w_u	proximity.ngram.length= 10	proximity.ngram.length=15
0.1	0.1	0.1329	0.1333
0.1	0.2	0.1342	0.1349
0.1	0.3	0.1354	0.1361
0.1	0.4	0.1370	0.1377
0.1	0.5	0.1388	0.1395
0.1	0.6	0.1401	0.1409
0.1	0.7	0.1415	0.1422
0.1	0.8	0.1424	0.1439
0.1	0.9	0.1441	0.1451
0.1	1	0.1458	0.1465
0.2	0.1	0.1342	0.1349
0.2	0.2	0.1370	0.1377
0.2	0.3	0.1401	0.1409
0.2	0.4	0.1424	0.1439
0.2	0.5	0.1458	0.1465
0.2	0.6	0.1476	0.1482
0.2	0.7	0.1492	0.1500
0.2	0.8	0.1503	0.1506

0.2	0.9	0.1511	0.1518
0.2	1	0.1522	0.1526
0.3	0.1	0.1354	0.1361
0.3	0.2	0.1401	0.1409
0.3	0.3	0.1441	0.1451
0.3	0.4	0.1476	0.1482
0.3	0.5	0.1495	0.1502
0.3	0.6	0.1511	0.1518
0.3	0.7	0.1526	0.1537
0.3	0.8	0.1535	0.1552
0.3	0.9	0.1542	0.1558
0.3	1	0.1548	0.1561
0.4	0.1	0.1370	0.1377
0.4	0.2	0.1424	0.1439
0.4	0.3	0.1476	0.1482
0.4	0.4	0.1503	0.1506
0.4	0.5	0.1522	0.1526
0.4	0.6	0.1535	0.1552
0.4	0.7	0.1543	0.1558
0.4	0.8	0.1551	0.1562
0.4	0.9	0.1558	0.1573
0.4	1	0.1567	0.1578
0.5	0.1	0.1388	0.1395
0.5	0.2	0.1458	0.1465
0.5	0.3	0.1495	0.1502
0.5	0.4	0.1522	0.1526
0.5	0.5	0.1538	0.1554
0.5	0.6	0.1548	0.1561
0.5	0.7	0.1555	0.1572
0.5	0.8	0.1567	0.1578
0.5	0.9	0.1566	0.1580
0.5	1	0.1572	0.1582
0.6	0.1	0.1401	0.1409
0.6	0.2	0.1476	0.1482
0.6	0.3	0.1511	0.1518
0.6	0.4	0.1535	0.1552
0.6	0.5	0.1548	0.1561
0.6	0.6	0.1558	0.1573
0.6	0.7	0.1567	0.1578
0.6	0.8	0.1569	0.1584
0.6	0.9	0.1573	0.1582
0.6	1	0.1573	0.1579
0.7	0.1	0.1415	0.1422
0.7	0.2	0.1492	0.1500
0.7	0.3	0.1526	0.1537
0.7	0.4	0.1543	0.1558
0.7	0.5	0.1555	0.1572
0.7	0.6	0.1567	0.1578
0.7	0.7	0.1569	0.1584
0.7	0.8	0.1574	0.1580

0.7	0.9	0.1572	0.1578
0.7	1	0.1573	0.1581
0.8	0.1	0.1424	0.1439
0.8	0.2	0.1503	0.1506
0.8	0.3	0.1535	0.1552
0.8	0.4	0.1551	0.1562
0.8	0.5	0.1567	0.1578
0.8	0.6	0.1569	0.1584
0.8	0.7	0.1574	0.1580
0.8	0.8	0.1573	0.1578
0.8	0.9	0.1574	0.1581
0.8	1	0.1582	0.1578
0.9	0.1	0.1441	0.1451
0.9	0.2	0.1511	0.1518
0.9	0.3	0.1542	0.1558
0.9	0.4	0.1558	0.1573
0.9	0.5	0.1566	0.1580
0.9	0.6	0.1573	0.1582
0.9	0.7	0.1572	0.1578
0.9	0.8	0.1574	0.1581
0.9	0.9	0.1582	0.1578
0.9	1	0.1576	0.1574
1	0.1	0.1458	0.1465
1	0.2	0.1522	0.1526
1	0.3	0.1548	0.1561
1	0.4	0.1567	0.1578
1	0.5	0.1572	0.1582
1	0.6	0.1573	0.1579
1	0.7	0.1573	0.1581
1	0.8	0.1582	0.1578
1	0.9	0.1576	0.1574
1	1	0.1568	0.1567

Tableau III.5 Résultat obtenu avec le modèle MRF (pour **ngram.length = 10 et 15**).

proximity.ngram.length= 20					
proximity.w_o	proximity.w_u	resultat	proximity.w_o	proximity.w_u	resultat
0.1	0.1	0.1339	0.6	0.1	0.1417
0.1	0.2	0.1354	0.6	0.2	0.1498
0.1	0.3	0.1366	0.6	0.3	0.1532
0.1	0.4	0.1383	0.6	0.4	0.1553
0.1	0.5	0.1402	0.6	0.5	0.1570
0.1	0.6	0.1417	0.6	0.6	0.1582
0.1	0.7	0.1437	0.6	0.7	0.1588
0.1	0.8	0.1452	0.6	0.8	0.1588
0.1	0.9	0.1463	0.6	0.9	0.1587
0.1	1	0.1478	0.6	1	0.1582
0.2	0.1	0.1354	0.7	0.1	0.1437

0.2	0.2	0.1383	0.7	0.2	0.1510
0.2	0.3	0.1417	0.7	0.3	0.1544
0.2	0.4	0.1452	0.7	0.4	0.1566
0.2	0.5	0.1478	0.7	0.5	0.1581
0.2	0.6	0.1498	0.7	0.6	0.1588
0.2	0.7	0.1510	0.7	0.7	0.1587
0.2	0.8	0.1524	0.7	0.8	0.1585
0.2	0.9	0.1532	0.7	0.9	0.1583
0.2	1	0.1544	0.7	1	0.1584
0.3	0.1	0.1366	0.8	0.1	0.1452
0.3	0.2	0.1417	0.8	0.2	0.1524
0.3	0.3	0.1463	0.8	0.3	0.1553
0.3	0.4	0.1498	0.8	0.4	0.1575
0.3	0.5	0.1517	0.8	0.5	0.1585
0.3	0.6	0.1532	0.8	0.6	0.1588
0.3	0.7	0.1544	0.8	0.7	0.1585
0.3	0.8	0.1553	0.8	0.8	0.1583
0.3	0.9	0.1562	0.8	0.9	0.1584
0.3	1	0.1570	0.8	1	0.1579
0.4	0.1	0.1383	0.9	0.1	0.1463
0.4	0.2	0.1452	0.9	0.2	0.1532
0.4	0.3	0.1498	0.9	0.3	0.1562
0.4	0.4	0.1524	0.9	0.4	0.1582
0.4	0.5	0.1544	0.9	0.5	0.1588
0.4	0.6	0.1553	0.9	0.6	0.1587
0.4	0.7	0.1566	0.9	0.7	0.1583
0.4	0.8	0.1575	0.9	0.8	0.1584
0.4	0.9	0.1582	0.9	0.9	0.1579
0.4	1	0.1585	0.9	1	0.1579
0.5	0.1	0.1402	1	0.1	0.1478
0.5	0.2	0.1478	1	0.2	0.1544
0.5	0.3	0.1517	1	0.3	0.1570
0.5	0.4	0.1544	1	0.4	0.1585
0.5	0.5	0.1557	1	0.5	0.1588
0.5	0.6	0.1570	1	0.6	0.1582
0.5	0.7	0.1581	1	0.7	0.1584
0.5	0.8	0.1585	1	0.8	0.1579
0.5	0.9	0.1588	1	0.9	0.1579
0.5	1	0.1588	1	1	0.1576

Tableau III.6 Résultat obtenu avec le modèle MRF (pour **ngram.length = 20**).

On remarque que le meilleur résultat obtenu dans le tableau est **" 0.1588 "** et les paramètres qui nous retourne ce résultat sont : (0.6 , 0.7 , 20) , (0.6 , 0.8 , 20) , (0.7 , 0.6 , 20) , (0.8 , 0.6 , 20) , (0.9 , 0.5 , 20) , (1 , 0.5 , 20) , (0.5 , 0.9 , 20) , (0.5 , 1 , 20) respectivement (proximity.w_o, proximity.w_u, ngram.length).

III.7.3 Résultats obtenus avec le modèle MRF Etendu (notre approche)

Dans cette étape nous avons évalué notre approche (MRF étendu) en utilisant les quatre formules citées dans le Tableau III.1. Les résultats sont présentés ci-dessous :

III.7.3.1 Résultat obtenu avec la formule (1)

La formule (1) à deux paramètres a faire varier. La taille de la fenêtre qui prend les valeurs 10, 25 et 50 pour chaque valeur nous avons varié le paramètre α, avec des valeurs allant de 0.2 à 0.8 avec un pas de 0.2.

Le tableau suivant résume les résultats obtenus :

NB fenêtre total = taille de document / taille de fenêtre		
Taille de fenêtre	α	Résultat
10	0.2	0.1568
	0.4	0.1567
	0.6	0.1567
	0.8	0.1565
25	0.2	0.1565
	0.4	0.1562
	0.6	0.1557
	0.8	0.1551
50	0.2	0.1563
	0.4	0.1557
	0.6	0.1549
	0.8	0.1539

Tableau III.7 Résultat obtenu avec le modèle MRF Etendu(pour **formule (1)**).

De ce tableau, nous avons eu le meilleur résultat **" 0.1568 "**avec les valeurs des paramètres (10 , 0.2) respectivement (taille de la fenêtre , α)

III.7.3.2 Résultats obtenus avec la formule (2)

La formule (2) à les mêmes paramètres que la formule (1).

Le tableau suivant résume les résultats obtenus :

NB fenêtre total = taille moyenne de document / taille fenêtre		
Taille de fenêtre	α	Résultat
10	0.2	0.1568
	0.4	0.1568
	0.6	0.1567
	0.8	0.1567
25	0.2	0.1567
	0.4	0.1565
	0.6	0.1563
	0.8	0.1560

	0.2	0.1564
	0.4	0.1563
	0.6	0.1560
50	0.8	0.1554

Tableau III.8 Résultat obtenu avec le modèle MRF Etendu(pour **formule (2)**).

De ce tableau, nous avons eu le meilleur résultat " **0.1568** " avec les valeurs des paramètres (10 , 0.2) , (10 , 0.4) respectivement (taille de la fenêtre , α)

III.7.3.3 Résultats obtenus avec la formule (3)

La formule (3) à trois paramètres a faire varier. La taille de la fenêtre qui prend les valeurs 10, 25 et 50 pour chaque valeur nous avons fait une combinaison entre les deux autres paramètres qui sont α et β, chacun de ces paramètres prend les valeurs 0.2, 0.5 et 0.8.

Le tableau suivant résume les résultats obtenus :

Taille fenêtre	A	β	Résultat
10	0.2	0.2	0.1546
	0.2	0.5	0.1528
	0.2	0.8	0.1504
	0.5	0.2	0.1531
	0.5	0.5	0.1509
	0.5	0.8	0.1466
	0.8	0.2	0.1515
	0.8	0.5	0.1471
	0.8	0.8	0.1414
25	0.2	0.2	0.1532
	0.2	0.5	0.1509
	0.2	0.8	0.1487
	0.5	0.2	0.1524
	0.5	0.5	0.1500
	0.5	0.8	0.1451
	0.8	0.2	0.1509
	0.8	0.5	0.1471
	0.8	0.8	0.1407
50	0.2	0.2	0.1519
	0.2	0.5	0.1497
	0.2	0.8	0.1458
	0.5	0.2	0.1511
	0.5	0.5	0.1488
	0.5	0.8	0.1429
	0.8	0.2	0.1506
	0.8	0.5	0.1462
	0.8	0.8	0.1400

Tableau III.9 Résultat obtenu avec le modèle MRF Etendu(pour **formule (3)**).

De ce tableau, nous avons eu le meilleur résultat " **0.1546** " avec les valeurs des paramètres (10 , 0.2 , 0.2) respectivement (taille de la fenêtre , α , β).

III.7.3.4 Résultats obtenus avec la formule (4)

La formule (4) à les mêmes paramètres que la formule (3).

Le tableau suivant résume les résultats obtenus :

Taille fenêtre	α	β	Résultat
10	0.2	0.2	0.1565
	0.2	0.5	0.1578
	0.2	0.8	0.1566
	0.5	0.2	0.1578
	0.5	0.5	0.1566
	0.5	0.8	0.1558
	0.8	0.2	0.1566
	0.8	0.5	0.1558
	0.8	0.8	0.1550
25	0.2	0.2	0.1500
	0.2	0.5	0.1523
	0.2	0.8	0.1527
	0.5	0.2	0.1523
	0.5	0.5	0.1558
	0.5	0.8	0.1549
	0.8	0.2	0.1558
	0.8	0.5	0.1550
	0.8	0.8	0.1543
50	0.2	0.2	0.1164
	0.2	0.5	0.1165
	0.2	0.8	0.1164
	0.5	0.2	0.1168
	0.5	0.5	0.1165
	0.5	0.8	0.1159
	0.8	0.2	0.1165
	0.8	0.5	0.1160
	0.8	0.8	0.1156

Tableau III.10 Résultat obtenu avec le modèle MRF Etendu(pour **formule (4)**).

De ce tableau, nous avons eu le meilleur résultat " **0.1578** "avec les valeurs des paramètres (10 , 0.2 , 0.5) , (10 , 0.5 , 0.2) respectivement (taille de la fenêtre , α , β).

III.7.4 Comparaison entre le modèle MRF et le MRF Etendu :

Nous présentons dans la figure suivante les meilleures précisions obtenues avec les quatre formules et le modèle MRF de base tel que :

MRF_S : Le MRF de base.

MRF_A : Le modèle MRF étendu avec la formule 1.

MRF_B : Le modèle MRF étendu avec la formule 2.

MRF_C : Le modèle MRF étendu avec la formule 3.

MRF_D : Le modèle MRF étendu avec la formule 4.

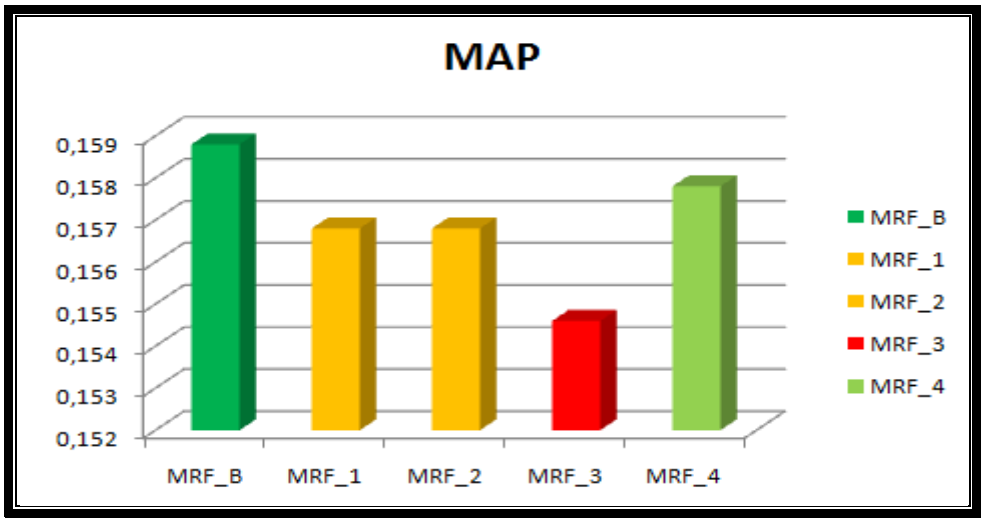

Figure III.5 Comparaison entre les meilleurs précisions obtenus de différents modèles.

Cette figure nous montre que notre approche (MRF Etendu) n'a pas donné d'amélioration comparativement avec le modèle MRF de base. Le tableau suivant nous illustre les taux d'amélionration de notre approche par rapport aux modèle MRF de base et le modèle de ponderation BM25 :

Formules	Modèles			Taux d'améliorations	
	BM25	MRF de base	MRF etendu	BM25	MRF de base
Formule 1	0.1296	0.1588	0.1568	+ 20.99	- 1.26
Formule 2	0.1296	0.1588	0.1568	+ 20.99	- 1.26
Formule 3	0.1296	0.1588	0.1546	+ 19.29	- 2.64
Formule 4	0.1296	0.1588	0.1578	+ 21.76	- 0.63

Tableau III.11 Taux d'améliorations obtenus avec notre approche.

De ce tableau on remarque que notre approche a porté une amélioration par rapport au modèle BM25 de l'ordre de 20.99%, 20.99%, 19.99%, 21.76% respectivement avec les formules (1),

(2), (3), (4). Et par rapport au modèle MRF de base il n'y a pas d'amélioration avec les quatre formules.

Dans la section suivante nous illustrons l'évaluation requête par requête.

III.7.5 Evaluation requête par requête :

Afin d'analyser en détail les résultats présentés précédemment, nous avons effectué une comparaison requête par requête entre notre modèle et le modèle MRF, le tableau suivant illustre les résultats obtenus.

requêtes	MAP MRF	MAP MRF étendu				Taux d'amélioration			
	Base	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)
51	0,6017	0,6337	0,6337	0,6278	0,6342	3,2	3,2	2,61	3,25
52	0,7140	0,6855	0,6853	0,6857	0,6839	-2,85	-2,87	-2,83	-3,01
53	0,2071	0,2060	0,2060	0,2055	0,2063	-0,11	-0,11	-0,16	-0,08
54	0,4432	0,4221	0,4221	0,4118	0,4272	-2,11	-2,11	-3,14	-1,6
55	0,4134	0,4045	0,4046	0,4022	0,4083	-0,89	-0,88	-1,12	-0,51
56	0,8110	0,7996	0,7995	0,7996	0,7995	-1,14	-1,15	-1,14	-1,15
57	0,1859	0,1859	0,1859	0,1857	0,1860	0	0	-0,02	0,01
58	0,0260	0,0248	0,0248	0,0237	0,0250	-0,12	-0,12	-0,23	-0,1
59	0,0100	0,0104	0,0104	0,0106	0,0107	0,04	0,04	0,06	0,07
60	0,0104	0,0075	0,0075	0,0068	0,0079	-0,29	-0,29	-0,36	-0,25
61	0,4798	0,4267	0,4267	0,4196	0,4286	-5,31	-5,31	-6,02	-5,12
62	0,0112	0,0101	0,0100	0,0100	0,0105	-0,11	-0,12	-0,12	-0,07
63	0,2672	0,2681	0,2681	0,2680	0,2681	0,09	0,09	0,08	0,09
64	0,0214	0,0209	0,0209	0,0181	0,0198	-0,05	-0,05	-0,33	-0,16
65	0,0000	0,0000	0,0000	0,0000	0,0000	0	0	0	0
66	0,0000	0,0000	0,0000	0,0000	0,0000	0	0	0	0
67	0,0010	0,0010	0,0010	0,0011	0,0010	0	0	0,01	0
68	0,0339	0,0418	0,0419	0,0412	0,0415	0,79	0,8	0,73	0,76
69	0,4836	0,5548	0,5535	0,5560	0,5555	7,12	6,99	7,24	7,19
70	0,2622	0,2592	0,2592	0,2589	0,2593	-0,3	-0,3	-0,33	-0,29
71	0,0735	0,0658	0,0658	0,0640	0,0643	-0,77	-0,77	-0,95	-0,92
72	0,0014	0,0017	0,0016	0,0018	0,0015	0,03	0,02	0,04	0,01
73	0,0002	0,0003	0,0003	0,0002	0,0003	0,01	0,01	0	0,01
74	0,0004	0,0004	0,0004	0,0005	0,0004	0	0	0,01	0
75	0,0139	0,0139	0,0139	0,0139	0,0140	0	0	0	0,01
76	0,1183	0,1054	0,1060	0,1056	0,1181	-1,29	-1,23	-1,27	-0,02
77	0,0455	0,0457	0,0457	0,0490	0,0445	0,02	0,02	0,35	-0,1
78	0,1672	0,1670	0,1670	0,1641	0,1683	-0,02	-0,02	-0,31	0,11
79	0,0000	0,0000	0,0000	0,0000	0,0000	0	0	0	0
80	0,0047	0,0037	0,0037	0,0049	0,0030	-0,1	-0,1	0,02	-0,17

81	0,1077	0,0957	0,0966	0,0971	0,0970	-1,2	-1,11	-1,06	-1,07
82	0,3979	0,3773	0,3773	0,3762	0,3780	-2,06	-2,06	-2,17	-1,99
83	0,0177	0,0195	0,0193	0,0203	0,0200	0,18	0,16	0,26	0,23
84	0,0746	0,0837	0,0837	0,0753	0,0838	0,91	0,91	0,07	0,92
85	0,0915	0,0808	0,0809	0,0822	0,0805	-1,07	-1,06	-0,93	-1,1
86	0,1453	0,1383	0,1383	0,1341	0,1636	-0,7	-0,7	-1,12	1,83
87	0,0091	0,0080	0,0080	0,0056	0,0081	-0,11	-0,11	-0,35	-0,1
88	0,1096	0,0994	0,0995	0,0996	0,0970	-1,02	-1,01	-1	-1,26
89	0,0043	0,0054	0,0054	0,0046	0,0055	0,11	0,11	0,03	0,12
90	0,2920	0,3610	0,3613	0,3552	0,3932	6,9	6,93	6,32	10,12
91	0,0023	0,0012	0,0012	0,0012	0,0010	-0,11	-0,11	-0,11	-0,13
92	0,0021	0,0013	0,0013	0,0016	0,0010	-0,08	-0,08	-0,05	-0,11
93	0,3403	0,4006	0,4046	0,3954	0,3942	6,03	6,43	5,51	5,39
94	0,0100	0,0100	0,0100	0,0101	0,0094	0	0	0,01	-0,06
95	0,0030	0,0028	0,0028	0,0033	0,0028	-0,02	-0,02	0,03	-0,02
96	0,0092	0,0114	0,0114	0,0106	0,0116	0,22	0,22	0,14	0,24
97	0,1023	0,1153	0,1153	0,1150	0,1158	1,3	1,3	1,27	1,35
98	0,3909	0,3981	0,3981	0,3701	0,4060	0,72	0,72	-2,08	1,51
99	0,2531	0,2569	0,2570	0,2561	0,2570	0,38	0,39	0,3	0,39
100	0,0651	0,0683	0,0683	0,0725	0,0606	0,32	0,32	0,74	-0,45
101	0,1989	0,1666	0,1666	0,1664	0,1667	-3,23	-3,23	-3,25	-3,22
102	0,0281	0,0196	0,0194	0,0194	0,0197	-0,85	-0,87	-0,87	-0,84
103	0,1983	0,1973	0,1973	0,2066	0,1931	-0,1	-0,1	0,83	-0,52
104	0,3354	0,3563	0,3563	0,3563	0,3511	2,09	2,09	2,09	1,57
105	0,0013	0,0011	0,0011	0,0013	0,0011	-0,02	-0,02	0	-0,02
106	0,1967	0,1930	0,1916	0,1758	0,1977	-0,37	-0,51	-2,09	0,1
107	0,5601	0,4442	0,4411	0,4621	0,4390	-11,59	-11,9	-9,8	-12,11
108	0,0439	0,0209	0,0209	0,0207	0,0216	-2,3	-2,3	-2,32	-2,23
109	0,0000	0,0000	0,0000	0,0000	0,0000	0	0	0	0
110	0,3409	0,3264	0,3261	0,3209	0,3270	-1,45	-1,48	-2	-1,39
111	0,6448	0,6531	0,6531	0,6515	0,6508	0,83	0,83	0,67	0,6
112	0,1906	0,1906	0,1906	0,1826	0,1949	0	0	-0,8	0,43
113	0,0305	0,0346	0,0346	0,0322	0,0370	0,41	0,41	0,17	0,65
114	0,0961	0,1005	0,1006	0,1008	0,1005	0,44	0,45	0,47	0,44
115	0,1767	0,2002	0,2016	0,1964	0,1984	2,35	2,49	1,97	2,17
116	0,1111	0,1429	0,1429	0,1429	0,1429	3,18	3,18	3,18	3,18
117	0,0321	0,0404	0,0406	0,0398	0,0436	0,83	0,85	0,77	1,15
118	0,0360	0,0314	0,0314	0,0337	0,0291	-0,46	-0,46	-0,23	-0,69
119	0,0421	0,0384	0,0385	0,0398	0,0365	-0,37	-0,36	-0,23	-0,56
120	0,0082	0,0097	0,0091	0,0089	0,0096	0,15	0,09	0,07	0,14
121	0,0064	0,0030	0,0030	0,0029	0,0031	-0,34	-0,34	-0,35	-0,33
122	0,3330	0,2421	0,2416	0,2410	0,2520	-9,09	-9,14	-9,2	-8,1
123	0,0122	0,0116	0,0117	0,0111	0,0110	-0,06	-0,05	-0,11	-0,12

124	0,1562	0,1542	0,1540	0,1463	0,1567	-0,2	-0,22	-0,99	0,05
125	0,0921	0,1043	0,1042	0,0913	0,0998	1,22	1,21	-0,08	0,77
126	0,1172	0,1245	0,1244	0,1246	0,1245	0,73	0,72	0,74	0,73
127	0,1526	0,1163	0,1164	0,1166	0,1280	-3,63	-3,62	-3,6	-2,46
128	0,2458	0,2685	0,2720	0,2730	0,2803	2,27	2,62	2,72	3,45
129	0,0387	0,0233	0,0235	0,0233	0,0254	-1,54	-1,52	-1,54	-1,33
130	0,1838	0,1636	0,1622	0,1629	0,1615	-2,02	-2,16	-2,09	-2,23
131	0,0298	0,0334	0,0334	0,0328	0,0339	0,36	0,36	0,3	0,41
132	0,4284	0,4114	0,4114	0,4085	0,4128	-1,7	-1,7	-1,99	-1,56
133	0,6917	0,6917	0,6917	0,6917	0,6917	0	0	0	0
134	1,0000	1,0000	1,0000	1,0000	1,0000	0	0	0	0
135	0,1009	0,1354	0,1294	0,1254	0,1373	3,45	2,85	2,45	3,64
136	0,0000	0,0000	0,0000	0,0000	0,0000	0	0	0	0
137	0,0082	0,0083	0,0083	0,0085	0,0080	0,01	0,01	0,03	-0,02
138	0,0462	0,0459	0,0460	0,0488	0,0430	-0,03	-0,02	0,26	-0,32
139	0,0588	0,0407	0,0407	0,0417	0,0396	-1,81	-1,81	-1,71	-1,92
140	0,0030	0,0031	0,0031	0,0034	0,0021	0,01	0,01	0,04	-0,09
141	0,0584	0,0463	0,0464	0,0456	0,0479	-1,21	-1,2	-1,28	-1,05
142	0,0173	0,0182	0,0182	0,0177	0,0185	0,09	0,09	0,04	0,12
143	0,0152	0,0130	0,0127	0,0132	0,0118	-0,22	-0,25	-0,2	-0,34
144	0,0423	0,0462	0,0456	0,0428	0,0493	0,39	0,33	0,05	0,7
145	0,1800	0,2007	0,2034	0,2020	0,2014	2,07	2,34	2,2	2,14
146	0,1641	0,1578	0,1575	0,1535	0,1607	-0,63	-0,66	-1,06	-0,34
147	0,0775	0,0809	0,0813	0,0745	0,0835	0,34	0,38	-0,3	0,6
148	0,0544	0,0611	0,0610	0,0612	0,0638	0,67	0,66	0,68	0,94
149	0,1251	0,0841	0,0841	0,0825	0,0846	-4,1	-4,1	-4,26	-4,05
150	0,0123	0,0092	0,0092	0,0099	0,0083	-0,31	-0,31	-0,24	-0,4

Tableau III.12 Comparaison requête par requête de modèle MRF et le modèle MRF étendu avec les quatre formules.

Les résultats obtenus dans ce tableau nous donnent les résultats suivants :

- Le modèle MRF étendu avec la formule 1 :
  - 39 requêtes améliorées.
  - 48 requêtes dégradés.
  - 13 requêtes nulles.
- Le modèle MRF étendu avec la formule 2 :
  - 39 requêtes améliorées.
  - 48 requêtes dégradés.
  - 13 requêtes nulles.

- Le modèle MRF étendu avec la formule 3 :
  - 41 requêtes améliorées.
  - 49 requêtes dégradés.
  - 10 requêtes nulles.
- Le modèle MRF étendu avec la formule 4 :
  - 42 requêtes améliorées.
  - 49 requêtes dégradés.
  - 9 requêtes nulles.

III.7.6 Analyse des résultats en se basant sur le type de requête

Nous avons aussi effectué des tests requête par requête mais basés sur le type requête. Nous avons classé nos requêtes en trois types :

1. Requête claires si MAP > = 0.2.
2. Requête moyennes si 0.2 > MAP > = 0.05.
3. Requête ambigües si MAP < 0.05.

Le tableau suivant illustre les résultats obtenus, tout en spécifiant le taux de requêtes améliorées pour chaque type de requête.

Requêtes	MRF Base	Formule 1		Formule 1		Formule 1		Formule 1	
		améliorer	taux améliorer	améliorer	taux améliorer	Améliorer	taux améliorer	améliorer	taux améliorer
134	CLAIRES		0.42		0.42		0.37		0.42
56									
52									
133									
111		+		+		+		+	
51		+		+		+		+	
107									
69		+		+		+		+	
61									
54									
132									
55									
82									
98		+		+				+	
110									
93		+		+		+		+	
104		+		+		+		+	
122									
90		+		+		+		+	

63		+		+		+		+	
70									
99		+		+		+		+	
128		+		+		+		+	
53									
101									
103						+			
106								+	
112								+	
57								+	
130									
145		+		+		+		+	
115		+		+		+		+	
78								+	
146									
124								+	
127									
86								+	
149									
76									
126	MOYENNES	+	0.39	+	0.39	+	0.35	+	0.55
116		+		+		+		+	
88									
81									
97		+		+		+		+	
135		+		+		+		+	
114		+		+		+		+	
125		+		+				+	
85									
147		+		+				+	
84		+		+		+		+	
71									
100		+		+		+			
139									
141									
148		+		+		+		+	
138						+			
77		+		+		+			
108									
144		+		+		+		+	
119	AMBIGUES		0.36		0.36		0.47		0.31
129									
118									
68		+		+		+		+	
117		+		+		+		+	

113		+		+		+		+		
131		+		+		+		+		
102										
58										
64										
83		+		+		+		+		
142		+		+		+		+		
143										
75								+		
150										
123										
62										
60										
59		+		+		+		+		
94						+				
96		+		+		+		+		
87										
120		+		+		+		+		
137		+		+		+				
121										
80						+				
89		+		+		+		+		
95						+				
140		+		+		+				
91										
92										
72		+		+		+		+		
105										
67						+				
74						+				
73		+		+				+		
65										
66										
79										
109										
136										

Tableau III.13 Taux d'amélioration obtenus entre le modèle MRF et notre MRF étendu en basent sur le type de requête.

De ce tableau nous avons eu les résultats suivants :

- Le modèle MRF étendu avec la formule 1 :

- 10 requêtes améliorées sur 24 requêtes claires par rapport au modèle MRF de base avec un taux d'amélioration de 42%.
- 12 requêtes améliorées sur 31 requêtes moyennes par rapport au modèle MRF de base avec un taux d'amélioration de 39%.
- 16 requêtes améliorées sur 45 requêtes ambigües par rapport au modèle MRF de base avec un taux d'amélioration de 36%.

- Le modèle MRF étendu avec la formule 2 :
  - 10 requêtes améliorées sur 24 requêtes claires par rapport au modèle MRF de base avec un taux d'amélioration de 42%.
  - 12 requêtes améliorées sur 31 requêtes moyennes par rapport au modèle MRF de base avec un taux d'amélioration de 39%.
  - 16 requêtes améliorées sur 45 requêtes ambigües par rapport au modèle MRF de base avec un taux d'amélioration de 36%.
- Le modèle MRF étendu avec la formule 3 :
  - 9 requêtes améliorées sur 24 requêtes claires par rapport au modèle MRF de base avec un taux d'amélioration de 37%.
  - 11 requêtes améliorées sur 31 requêtes moyennes par rapport au modèle MRF de base avec un taux d'amélioration de 35%.
  - 21 requêtes améliorées sur 45 requêtes ambigües par rapport au modèle MRF de base avec un taux d'amélioration de 47%.
- Le modèle MRF étendu avec la formule 4 :
  - 10 requêtes améliorées sur 24 requêtes claires par rapport au modèle MRF de base avec un taux d'amélioration de 42%.
  - 17 requêtes améliorées sur 31 requêtes moyennes par rapport au modèle MRF de base avec un taux d'amélioration de 55%.
  - 14 requêtes améliorées sur 45 requêtes ambigües par rapport au modèle MRF de base avec un taux d'amélioration de 31%.

III.8 Conclusion

Dans ce chapitre nous avons présenté notre approche qui consiste a étendre le modèle MRF avec le facteur de couverture spatiale, que nous avons intégré via l'utilisation de quatre formules. Puis nous avons testé cette approche. Cependant, notre approche n'apporte pas d'amélioration en utilisant la MAP sur l'ensemble des requêtes. Par contre, nous avons constaté que certaines requêtes ont été améliorées par notre approche.

Conclusion générale

Notre travail se situe dans le contexte de la recherche d'information. La problématique posée dans le cadre de ce mémoire étant l'extension du modèle MRF.

Durant notre étude et pour atteindre cet objectif, nous avons utilisé une base documentaire qui la collection de test TREC AP88 (Associated Press 1988).

Ce travail nous a permis d'acquérir des connaissances dans un domaine en pleine expansion qui est la recherche d'information.

Sur le plan théorique nous avons pu apprendre à développer, à analyser et à formuler nos rapports dans un cadre pédagogique.

Sur le plan conceptuel, nous avons acquis des connaissances sur le fonctionnement des SRI. Enfin, sur le plan pratique, nous nous sommes familiarisés avec la programmation sous l'environnement de développement NetBeans et nous avons amplifié nos connaissances sur le langage Java qui est un langage en pleine évolution.

Les résultats obtenus avec notre approche n'ont pas donné d'amélioration sur la collection utilisée, nous prévoyons de tester cette approche sur d'autres collection plus volumineuses. De plus, nous prévoyons d'explorer d'autre formalisations pour le facteur de couverture spatiale ainsi que l'exploration d'autre techniques d'intégration de ce facteur.

Références bibliographiques

[1]M^{r}. Abdelkrim Bouramoul thèse de doctorat thème Recherche d'information : contextuelle et sémantique sur le web .

[2]Zemirli W.Nesrine Vers le développement d'un système de recherche d'information personnalisé intégrant le profil utilisateur

[3]M^{r} Hammache arezkithèse de doctorat thème Recherche d'Information : un modèle delangue combinant mots simples et motscomposés

[4] Mémoire de Magister : Contribution à la définition d'une approche d'indexation sémantique de documents textuels

[5]M^{r} Hammache arezki thèse de doctorat thème Recherche d'Information : un modèle delangue combinant mots simples et motscomposés

[6]M^{elle} Wassila AZZOUG Mémoire de Magister Option : Spécification de Logiciels et Traitement de l'Information(École Doctorale)

[7] thèse de doctorat université de Toulouse thème :Un modèle de recherche d'information basse sur les graphes et les similarités structurelles pour l'amélioration du processus de recherche d'information

[8] ZemirliW.Nesrine Vers le développement d'un système de recherche d'information personnalisé intégrant le profil utilisateur

[9] M^{r}. Abdelkrim Bouramoulthèse de doctorat thème Recherche d'information : contextuelle et sémantique sur le web .

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

[11] 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.

[12] 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.

[13] 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.

[14] J.J. Rocchio. "Relevance feedback in information retrieval". In The SMART retrieval

system-experiments in automatic document processing, pages 313,323. Prentice Hall Inc, 1971.

[15] M. Lesk. Grab, inverted indexes with low storage overhead. Computing Systems,1(3):207–220, 1988.

[16]*Jean*-charles, lamirel .Indexation et classification et RI.

[17] Josiane Mothe, thèse sur la recherche et l'exploration d'informations Découverte de connaissances pour l'accès à l'information.

[18] mémoire samir.

[19] M. Boughanem, W. Kraaij, and J-Y Nie. Modèles de langue pour la recherche d'information. Les systèmes de recherche d'informations, pages 163–182,2004.

[20] J. M. Ponte and W. B. Croft. A language modeling approach toinformationretrieval.Proc. SIGIR, pages 275–281, 1998.

[21] Mohand Boughanem: les Systèmes de Recherche d'Information : d'un modèleclassique à un modèle connexionniste. Thèse de Doctorat de l'Université Paul Sabatier, 1992.

[22] J. Mothe, Modèle Connexionniste Pour la Recherche d'Informations. Expansion Dirigéede Requêtes et Apprentissage. Thèse de Doctorat en Informatique de l'Université Paul Sabatier deToulouse (Sciences). Octobre 1994.

[23] M. Maron, and J. Kuhns, On relevance, probabilistic indexing andinformation retrieval. Journal of the Association for Computing Machinery 7(1960), pages216–244.

[24] Robertson, S. E. (1977). The probability ranking principle in IR.Journal of Documentation, 33 (4), 294-304.

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

[26] ROBERTSON S,WALKER S, JONES S,GATFORD M. H.-B., « Okapi at 3 », Proceedings of the 3rd Text Retrieval Conference (-3), p. 109-126, 1994.

[27] Modèles de langue pour la recherche d'information de Mohand Boughanem, Institut de Recherche en Informatique de Toulouse, 118 route de Narbonne 31000 Toulouse Cedex 9, France.

[28]http://fr.wikipedia.org/wiki/Text_REtrieval_Conference.

[29] M[r]Hammachearezkithèse de doctorat thème Recherche d'Information : un modèle delangue combinant mots simples et motscomposés

[30] M[r]Hammachearezkithèse de doctorat thème Recherche d'Information : un modèle delangue combinant mots simples et motscomposés

[31]M[elle]Soheila KARBASI thèse doctorat thème : Pondération des termes en Recherche d'Information l'Université Paul Sabatier - Toulouse III

[32]M[r]Hammachearezkithèse de doctorat thème Recherche d'Information : un modèle delangue combinant mots simples et motscomposés

[33]S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau. Okapi at trec-3. In *The Third Text REtrieval Conference(TREC-3) NIST*. D. K. Harman, 1995.

[34]M[elle]Soheila KARBASI thèse doctorat thème : Pondération des termes en Recherche d'Information l'Université Paul Sabatier - Toulouse III

[35] Cutler, M. Shih, Y. Meng, W. Using the Structure of HTML Documents to Improve Retrieval. *Proceedings of the USENIX Symposium on Internet Technologies and Systems Monterey*, pp. 22-22 1997.

[36] Université M'hamed BOUGARA de BOUMERDES Reformulation de requêtes dans les systèmes de recherche d'information dans des documents XML Par : MATAOUI M'hamed

[37] A. Deutsch, M. Fernandez and D. Suciu. Storing semistructured data with STORED. In Proc. SIGMOD, 1999.

[38] J. Harding, Q. Li, B. Moon. XISS/R: XML Indexing and Storage System Using RDBMS. In Proceedings of the 29th VLDB Conference, 2003

[39] Software AG. Tamino XML database. http://www.softwareag.com/tamino/.

[40] XYZFind. XML Database. http://www.xyzfind.com.

[41] HYREX. http://ls6-www.cs.uni-dortmund.de/ir/projects/hyrex/.

[42] N. Fuhr and K. Großjohann. XIRQL: A Query Language for Information Retrieval in XML Documents. In Research and Development in Information Retrieval, 2001.

[43] J. E. Wolff, H. Florke, and A. B. Cremers. Searching and Browsing Collections of Structural Information. In Proc. IEEE Forum on Research and Technology Advances in Digital Libraries, 2000.

[44] T. Schlieder and H. Meuss. Querying and Ranking XML Documents. Special Topic Issue Journal American Society for Informations Systems on XML and Information Retrieval, 2002.

[45] T. Schlieder. Similarity Search in XML Data using Cost-Based Query Transformations. In Proc. 4th Intern. Workshop on the Web and Databases, 2001.

[46] A. Theobald and G. Weikum. The Index-Based XXL Search Engine for Querying XML Data with Relevance Ranking. In Proc. 8th Internation Conf. on Extending Database Technology, 2002.

[47] S. Robertson, H. Zaragoza, M. Taylor. Simple BM25 Extension to Multiple Weighted Fields. CIKM'04, 2004.

[48] Field-Weighted XML Retrieval Based on BM25 Center for Studies of Information Resources School of Information Management Wuhan University, China **Stephen Robertson Andrew Macfarlane**

[49] J. R. Dominick. The Dynamics of Mass Communication. McGraw-Hill Inc., 1990

[50] Enhancing Relevance Scoring with Chronological Term Rank Guo-Qiang Zhang et Adam D. Troy

[51]Metzler, D., Croft, W.B. A Markov random field model for term dependencies, in: R.A. Baeza-Yates, N. Ziviani, G. Marchionini, A. Moffat, J. Tait (Eds.). Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 472–479, 2005.

[52] Lv, Y., Zhai. C. Positional language models for information retrieval. *Proceedings of the 32th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 299-306, 2009.

[53] Tao, T., Zhai, C. An exploration of proximity measures in information retrieval. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pages 295–302, 2007.

[54] Zhao, J., Yun, Y. A proximity language model for information retrieval". *Proceedings of the 32th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 291–298, 2009.

[55] Daniel VIEILLARD, "*Les Technologies associées à XML"*, 2000. *http://daniel.veillard.com/Talks/200011XML/Overview.html*

[56] Stéphane Allorge, *" XML"*, département ASI, 2000.

[57] Bernd Amann, Cours *"XML et les bases de données : Introduction à la gestion de contenus Web et XML"*, Module Données et services sur le Web 2003/2004.

[58] Ines Bannour et Haïfa Zargayouna Une plate-forme open-source de recherche d'information sémantique.

[59] https://fr.wikipedia.org/wiki/Java_(langage)

[60]https://fr.wikipedia.org/wiki/NetBeans#Applications_sur_serveurs_.28applications_Web_et_JAVA_EE.29

[61] H. P. Luhn. The automatic creation of literature abstracts. IBM Journal of Research and Development

[62] Cutler, M. Shih, Y. Meng, W. Using the Structure of HTML Documents to Improve Retrieval. *Proceedings of the USENIX Symposium on Internet Technologies and Systems Monterey*, pp. 22-22 1997.

[63] Robertson S. & Sparck-Jones K., "Simple proven approaches to text retrieval", Tech rep tr356, Computer Laboratory University of Cambridge, 1997.