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

Mémoire De fin d'études

En vue de l'obtention du diplôme de
Master en Informatique
Option : Conduite de Projets Informatiques (CPI)

Thème :

Implémentation et évaluation d'un modèle de la RI basé sur la position des termes

Proposé et Dirigé par : Mr HAMMACHE .A

Réalisé par : Mlle MENSOUS Sabrina

Mlle DAHMANI Kahina

Promotion 2014/2015

Remerciements

Nous remercions tout d'abord « DIEU » tout puissant de nous avoir donné la santé et le courage d'amener ce projet à terme.

Nous tenons particulièrement à remercier Mr Hammache, notre promoteur, pour la finesse de ses attitudes sur le plan aussi bien humain que scientifique. Il a su trouver le juste équilibre entre la liberté qu'il nous a laissée dans le choix des grandes orientations et dans la détermination des pistes à suivre
De lui, nous avons toujours reçu non seulement les encouragements dont nous avions tant besoin, mais aussi les précieux conseils pratiques que seul un homme, ayant des qualités humaines comme lui, peut amener à prodiguer.
Nous lui en sommes infiniment gré.

Nous adressons nos remerciements aux membres du jury, pour l'honneur qu'ils nous font d'accepter de juger notre travail.

A tous ceux qui ont contribués de prés ou de loin à l'élaboration de ce modeste travail, qu'ils trouvent ici l'expression de nos remerciements les plus sincères.

MERCI

Dédicace

Je dédie ce modeste travail à

A mes chers parents qui sont la source de mon

courage et de ma persévérance, qu'aucune dédicace

ne saurait exprimer l'amour et la reconnaissance que

j'ai toujours pour eux , je vous dédie aujourd'hui ma

réussite. Puisse Dieu, le tout puissant, vous préserver

et vous accorder santé, longue vie et bonheur.

A mes frères Mohammed Said et Amazigh.

A toutes mes soeurs pour tout ce

qu'elles ont fait pour moi.

A ma binôme Nina et toute sa famille.

A toute ma famille,mes cousins et cousines et mes oncles et tantes.

Mes amis(es) et mes camarades de labo 8 pour toute l'ambiance

de travail qu'on a partagé ensemble, Toutes les

Dédicace

Je dédie ce travail

A La lumière de ma vie qui n'a jamais cessé de m'éclaire

et à qui je dois tout : mes très chers parents.

A ma sœur et mon frère

A tous (tes) mes amis (es).

A toutes mes cousines.

A mon binôme kahina et à toute sa famille.

A toutes les personnes qui pensent à moi de prés ou de loin.

Nina

SOMMAIRE

Sommaire

Chapitre II : Les schémas de pondération

Chapitre III : Implémentation et tests du modèle CTR et ses extensions

Table des figures

Chapitre I : Recherche d'information classique

Chapitre III : Expérimentation et Evaluation d'un modèle basé sur la position des termes

Table des figures

Table des figures

Liste des tableaux

Introduction générale

La Recherche d'Information (RI) est définie comme étant l'ensemble des méthodes et techniques pour l'acquisition, l'organisation, le stockage, la recherche et la sélection d'information pertinente pour un utilisateur. Elle est historiquement liée aux sciences de l'information et à la bibliothéconomie. Effectivement, les premiers systèmes ont été construits afin d'aider les bibliothécaires à retrouver des documents contenus dans des bases bibliographiques. Cependant, l'avènement d'Internet et plus particulièrement du Web ainsi que la prolifération de la masse documentaire a conduit à révéler la RI au grand jour.

En effet, face à ces innovations, le développement de moyens performants pour la recherche d'information, est devenu indispensable, afin de permettre à chacun de trouver une information précise répondant à son besoin en information. La RI a alors évolué vers des tâches de plus en plus nombreuses et diversifiées. Les Systèmes de Recherche d'Information (SRI) doivent aujourd'hui savoir traiter des volumes gigantesques de données, s'adapter aux nouveaux modes de communication, gérer la nature multimédia de l'information (l'image, le son, la vidéo, le texte, etc.).

La plus part des SRI existants représentent les documents comme un ensemble de mots clés, ce que l'on appelle communément une représentation en sac de mots. Ces mots clés sont généralement pondérés en utilisant des schémas de pondération tels que TF-IDF et BM25, 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.
Le facteur de position des termes a été récemment introduit, dans les formules de pondération, sous différents points de vues : la structure de document, la proximité des termes de la requête dans un document et le modèle CTR, ce dernier est basé sur l'exploitation de la position chronologique des termes de la requête dans un document. Son intuition est simple : "les termes les plus importants et pertinents du document, sont généralement placés juste au début du document".

Notre travail se situe dans le contexte de la RI et a un double objectifs, le premier consiste à implémenter et évaluer le modèle CTR sous la plate-forme Terrier. Le second consiste à proposer des extensions à ce modèle, ensuite les implémenter et les évaluer.

Afin de bien mener cette étude, nous avons opté d'organiser notre mémoire comme suit :
Chapitre I : ce chapitre est consacré à la recherche d'information classique et les concepts de base des SRI en présentant la description générale et l'architecture des SRI et le processus de RI, et ensuite nous allons décrire en détail les différents modèles de RI existants ainsi que l'étape d'évaluation des SRI.

Chapitre II: dans ce chapitre nous présentons en détail d'une part les facteurs de pondération existants, d'autre part nous présentons les travaux clés de la littérature exploitant ces facteurs dans leurs modèles de pondération.

Chapitre III : dans le troisième et dernier chapitre, nous présentons l'implémentation et l'évaluation du modèle CTR est ses extensions. Au premier lieu nous décrirons les outils et le langage de programmation utilisés. Au second lieu, nous présentons les collections utilisés et les résultats des expérimentations munies.

À la fin de ce document, une conclusion fait le bilan sur l'ensemble de cette étude et indique les perspectives de développement de notre travail.

Chapitre I:

"La Recherche d'Information Classique"

I.1 .Introduction

La numérisation des documents et le développement des technologies internet engendre une augmentation incessante de la masse de documents disponibles, par conséquent, le problème n'est plus la disponibilité de l'information mais la capacité de la sélection de l'information répondant aux besoins précis d'un utilisateur, à partir des représentations qu'il perçoit.

La conception et la mise en œuvre d'outils notamment les systèmes de recherche d'information (SRI), devient une nécessité absolue.

Le but d'un SRI est de retrouver, parmi une collection de documents préalablement stockés, les documents qui répondent au besoin utilisateur exprimé sous forme de requête. Pour cela, un SRI met en œuvre un ensemble de processus de sélection des documents pertinents pour la requête.

Ce chapitre a pour objectif de présenter le domaine de la RI. Principalement, nous présentons les concepts de la RI, l'architecture des systèmes de RI, les modèles de RI. Enfin l'évaluation des SRI.

I.2.Définition et historique de la RI

I.2.1.Définition de la RI

Plusieurs définitions ont été proposées pour la RI **[1]**:

La Recherche d'Information (RI) est le processus qui permet, à partir d'une expression des besoins d'information d'un utilisateur, de retrouver l'ensemble des documents contenant l'information recherchée et ce par la mise en œuvre d'un mécanisme qui permet de mesurer la pertinence entre la requête de l'utilisateur et les documents ou plus exactement entre la représentation de la requête et la représentation des documents.

La recherche d'information (Information Retrieval) est un domaine historiquement lié aux sciences de l'information et à la bibliothéconomie. Elle traite l'information dans la manière de l'organiser et de la façon de la sélectionner**.**
Le « Vocabulaire de la documentation ».- ADBS[1], nous propose les définitions suivantes afin de différencier la recherche de l'information de la recherche d'information:

- Recherche d'information : Ensemble des méthodes, procédures et techniques permettant, en fonction de critères de recherche, de sélectionner l'information dans un ou plusieurs fonds documentaires plus ou moins structurés.
- Recherche de l'information : Ensemble des méthodes, procédures et techniques ayant pour objectif d'extraire d'un document ou d'un ensemble de documents les informations pertinentes.

[1] Créée en 1963, l'ADBS, forte de ses 4 000 adhérents, c'est la première association de professionnels de l'information et de la documentation en Europe.

I.2.2. Evolution de la RI

L'évolution historique de la RI suit les étapes suivantes **[2]**:

- **Dans les années 1940 :**

Le domaine de la RI date des années **1940**, dès la naissance des ordinateurs. Il s'agissait alors d'automatiser des applications de bibliothèques. Les premiers systèmes ont été utilisés par des libraires et permettaient d'effectuer des recherches booléennes.

- **Dans les années 1950 :**

On commençait de petites expérimentations en utilisant des petites collections de documents (références bibliographiques). Le modèle utilisé est le modèle booléen.

- **Dans les années 1960 et 1970 :**

Des expérimentations plus larges ont été menées notamment avec le système **SMART** (*Salton's Magical Automatic Retriever of Text*) développé à la fin des années **1960** et au début des années **1970** par **G. Salton [16].**

- **Dans les années 1980 :**

Ces années ont été influencées par le développement de l'intelligence artificielle (IA). On a alors tenté d'intégrer des techniques de l'IA en RI, avec le développement par exemple, de systèmes experts pour la RI.

- **Dans les années 1990 (surtout à partir de 1995) :**

Sont les années de l'Internet. Cela a eu pour effet de propulser la RI en avant scène de beaucoup d'applications. La problématique de la RI s'est élargie. Par exemple, on traite maintenant plus souvent des documents multimédia qu'avant**.**

I.3 .Les Systèmes de Recherche d'Information

I.3.1.Définition d'un SRI

Un système de recherche d'information (SRI) est un ensemble de logiciels assurant l'ensemble des fonctions nécessaires à la recherche de l'information. Il offre des techniques et des outils permettant de localiser et de visualiser l'information pertinente relativement à un besoin en information, exprimé par un utilisateur sous forme de requête **[3].**

I.3.2.L'architecture d'un SRI

Le rôle d'un système de Recherche d'Information(SRI) est de mettre en œuvre des techniques et des moyens permettant de retourner les documents pertinents d'une collection en réponse à un besoin en information d'un utilisateur, exprimé par un langage de requêtes qui peut être le langage naturel, une liste de mots clés ou un langage booléen **[3].**

Afin d'atteindre cet objectif, un processus d'indexation des documents de la collection est effectué. Il permet de construire une représentation synthétique des documents, appelée index. Lorsque l'utilisateur formule sa requête un processus similaire est effectué sur la requête. Il consiste à analyser la requête et établir une représentation interne. Puis, le système établit une correspondance entre la représentation de la requête et la représentation des documents (index) pour sélectionner et présenter les documents qui répondent le mieux au besoin de l'utilisateur (les documents pertinents).

Le SRI s'appuie sur des modèles de RI pour établir cette correspondance entre les documents et la requête.

L'architectures générale d'un SRI illustrée par la figure I.1 fait ressortir des éléments constitutifs tels que : le document, le besoin en information la requête et la pertinence, ainsi que trois principales fonctionnalités : l'indexation, la recherche et la reformulation de la requête. Dans ce qui suit, nous détaillons ces éléments et processus.

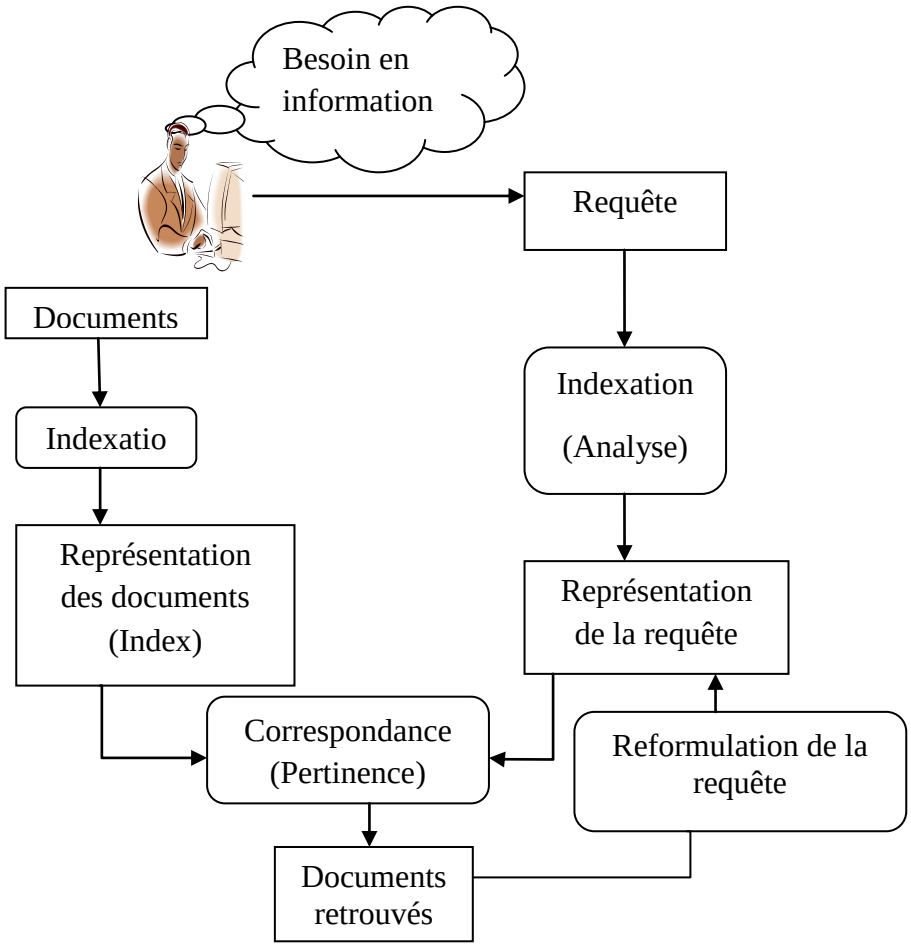

Figure I.1 : Architecture générale d'un Système de Recherche d'Information **[4].**

La Figure I.1 fait ressortir des concepts (éléments) de base et un ensemble de processus que nous décrivons dans ce qui suit.

I.3.3.Concepts de base

I.3.3.1.Besoin en information et requête

La requête est une expression approximative du besoin en information de l'utilisateur. Ce dernier est une expression mentale des informations que l'utilisateur recherche **[5]**.

Les requêtes soumises au SRI par les utilisateurs peuvent ne pas refléter leurs besoins en information. Cela est du, 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 (centre d'intérêt, niveaux, parcours, etc.). Ce biais entre la requête et le besoin en information est une difficulté majeure de tout système de recherche d'information. Afin de remédier partiellement à ce problème un mécanisme de reformulation de requête peut être intégré dans Les SRI.

I.3.3.2.Document et collection de documents

La notion de document est définie de manière à l'affranchir du support. C'est un ensemble constitué d'information portée par un support. L'unité de base constitutive du document est l'information. Ce qui caractérise le document est le fait que l'information y est délimitée et structurée sous forme de mots, de sons, d'images ou de vidéos. La notion est définie de manière à y inclure tous les supports connus ou qui pourront un jour exister **[40].**

La collection de documents constitue l'ensemble des informations exploitables, compréhensibles et accessibles par l'utilisateur. Une collection comporte un ensemble de documents **[40].**

I.3.3.3.Pertinence

La pertinence est une notion fondamentale et cruciale dans le domaine de la RI. Elle est le critère primaire pour l'évaluation des systèmes de recherche d'informations. Les travaux de recherche récents s'accordent sur la difficulté de la définition de la pertinence **[6].**

La pertinence est aussi définit par la correspondance entre un document et une requête, une mesure d'informativité du document à la requête **[23].**

On distingue deux types de pertinences :

a) pertinence Système

La pertinence système est souvent présentée par un score attribué par le SRI afin d'évaluer l'adéquation du contenu des documents vis-à-vis de celui de la requête. Ce type de pertinence est objectif et déterministe **[7].**

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 ou le sujet a évolué **[8].**

I.3.4. Les processus de SRI

L'objectif fondamental d'un processus de RI est de sélectionner les documents ‘‘les plus proches’’ du besoin en information de l'utilisateur décrit par une requête. Ceci induit deux principales phases dans le déroulement du processus : l'indexation et l'appariement requête-document. Ainsi que , un troisième processus est souvent utilisé pour rapprocher la pertinence système de celle de l'utilisateur, ce dernier est nommé, la reformulation de la requête.

I.3.4.1.L'Indexation

I.3.4.1.1.Définition

L'indexation est une étape très importante dans le processus de RI. Elle consiste à déterminer et à extraire les termes représentatifs du contenu d'un document ou d'une requête avec leur poids associés. La qualité de la recherche dépend en grande partie de la qualité de l'indexation.

Ce processus effectue le transfert de l'information contenue dans le texte d'un document vers un autre espace de représentation traitable par un système informatique **[9].**

L'indexation recouvre un ensemble de techniques visant à transformer les documents (ou requêtes) en substituts ou descripteurs capables de représenter leur contenu **[10].** Ces descripteurs forment le langage d'indexation représenté selon une structure souvent basé sur un ensemble de mots clés ou groupes de mots représentant le contenu textuel du document.

I.3.4.1.2.Types d'indexation

L'indexation peut se réaliser de différentes manières :

1) **Selon le langage :** l'indexation peut alors être: libre ou contrôlée **[11].**

   - ***l'indexation libre :*** les termes sont extraits du texte à indexer.

- ***l'indexation contrôlée :*** elle s'appuie sur un ensemble prédéfini de termes par exemple un thésaurus. Elle consiste à sélectionner les termes de ce thésaurus qui indexent ce texte.

2) Selon le moteur : l'indexation peut alors être: manuelle, automatique, ou semi-automatique.

a) L'indexation manuelle

C'est une opération réalisée par un spécialiste ou un documentaliste qui consiste à recenser les sujets ou concepts dont traite un document et à les représenter à l'aide d'un vocabulaire contrôlé prédéfini. Elle est très souvent critiquée pour son coût ; la personne chargée de l'analyse des documents doit posséder les connaissances minimales à la compréhension des centres d'intérêt du document, sous peine d'obtenir une indexation incorrecte, et pour son haut degré de subjectivité ; même si l'indexation s'appuie sur un langage documentaire[2], des descripteurs différents peuvent être proposés pour représenter un même document.

b) L'indexation automatique

Le processus d'indexation est entièrement automatisé. Il présente l'avantage d'une régularité du processus, car l'indexation automatique fournit toujours le même index pour le même document.

Les techniques basiques de l'indexation automatique se font selon les étapes suivantes :

1. La simplification du contenu documentaire :

Cette étape comprend les traitements suivants :

- La tokenisation (extraction des mots) : La tokenisation consiste à segmenter un texte (document) en plusieurs unités atomiques, appelées jetons ou tokens dans ce cas les tokens seront les mots composant notre texte.
- L'élimination des mots vides : appelés aussi mots outils (« stop words » en anglais) sont des mots qui permettent de structurer une phrase. On citera les articles, les conjonctions de coordination, les verbes auxiliaires, etc. Ces mots ne portent pas de sens. On peut donc choisir de les éliminer. Pour cela, il existe, pour un certain nombre de langues, des dictionnaires de mots vides aussi appelés anti-dictionnaires. Attention certains mots vides ont pour homographes des mots significatifs. Par exemple, le mot « or ».
- La racinisation : (« stemming » en anglais) consiste à rechercher la forme tronquée d'un mot à partir de laquelle peuvent être reconstruites ses différentes variantes morphologiques permettant ainsi de les regrouper en considérant leur racine.

  Les mots : **étud**e, **étud**iant et **étud**ier, sont construits à partir de la racine **étud**. Cette opération peut être réalisée assez simplement. Par exemple, l'algorithme de Porter proposé par Martin Porter en 1980**[41].** Il se présente comme un ensemble de règles dont

[2]Un langage documentaire permet de représenter de manière univoque les notions identifiées dans les documents et dans les requêtes des utilisateurs.

l'application successive à un mot de l'anglais produit la racine de ce mot. Par contre, la racinisation peut entraîner une perte de sens, car la racine extraite peut être commune à des mots se rapportant à des concepts différents : Les mots **port**, **port**es (ouverture) et **port**era ont la même racine **port** mais se rapportent à trois concepts différents.
La racinisation permet d'augmenter le rappel mais peut diminuer la précision (le rappel et la précision seront définis plus loin).

- La lemmatisation : consiste à remplacer un mot par son lemme. Les mots port, portes et portera seront remplacés par leurs lemmes : port, porter ou porte selon le contexte et porter. Le lemme : forme canonique (infinitif pour les verbes, singulier pour les noms, etc.) qui constitue en général les entrées dans un dictionnaire de cette langue.

2. La sélection des meilleurs index :

Elle consiste à éliminer les termes non discriminants et à réduire la taille de l'index pour cela la conjecture de Luhn **[12]** et la loi de Zipf **[24]** sont utilisées.

- **La conjecture de Luhn :** elle offre une approche basée sur le critère de fréquence soulignant que :
  - ✓ Un terme d'indexation qui apparait trop fréquemment dans un texte est un mot vide qui ne joue aucun rôle sémantique dans ce texte, son rôle est seulement syntaxique, donc, il ne doit pas être utilisé dans le langage d'indexation. C'est une autre manière de supprimer les mots vides.
  - ✓ Les mots de fréquence quasi nulle (très rares) sont considérés comme peu pertinents.
  - ✓ Un terme d'indexation de fréquence intermédiaire est considéré comme significatif, il représente le contenu sémantique de l'unité documentaire et appartient au langage d'indexation.

  Pour cela deux seuils de fréquence (seuil max et seuil min), sont fixés et seuls les termes entre ces deux seuils sont alors pertinents pour représenter les documents.
- **Loi de Zipf :** c'est une loi empirique énoncée en 1949 par G.K Zipf **[24]**. Selon Zipf, les mots dans les documents ne s'organisent pas de manière aléatoire mais suivant une loi inversement proportionnelle à leur rang. Le rang d'un mot est sa position dans la liste décroissante des fréquences des mots du corpus. Ainsi, la fréquence du second mot le plus fréquent dans le corpus est la moitié de celle du premier, la fréquence du troisième mot le plus fréquent, son tiers, etc. Formellement, cette loi s'exprime par la probabilité d'apparition du nième mot le plus fréquent dans une collection de n'importe quelle langue est approximativement inversement proportionnelle à n (rang), soit :

$$P(n) = \frac{CN}{n} \qquad \text{(I.1)}$$

On en déduit que: fréquence * rang = Constante

3. La construction de l'index :

L'indexation des documents génère des structures d'index sophistiquées notamment les fichiers directs et fichiers inverses.

- Le fichier direct : l'indexation est orientée document en lui attribuant un ensemble de descripteurs.
- Le fichier inverse : l'indexation est orientée requête, pour chaque terme on lui associé les documents qui l'indexe.

Exemple: Soit quatre documents : d1, d2, d3, d4 tel que :

d1 = {bd, xml}
d2 = {bd, xml}
d3 = {bd, réseau}
d4 = {xml}

Le fichier direct obtenu est :

d1	Bd	Xml
d1	Bd	Xml
d3	Bd	Réseau
d4	Xml	

Le fichier inverse obtenu est :

XML	d1	d2	d4
Bd	d1	d2	d3
réseau	d3		

Figure I.2 : Exemple de fichier direct et inverse.

La requête est aussi indexée, retournant comme résultat un ensemble de termes (mots clés) exprimant le besoin en information de l'utilisateur.

c) L'indexation semi-automatique

Appelée aussi indexation supervisée, ce mode d'indexation jumelle entre l'indexation manuelle et l'indexation automatique toute fois le choix final des index reste tout de même aux indexeurs humains.

Bien que le vocabulaire contrôlé soit souvent utilisés pour l'indexation manuelle et semi-automatique, alors que le vocabulaire libre quand à lui est utilisé pour l'indexation automatique, plusieurs travaux de recherche ont été abordés dans la littérature pour une indexation contrôlée automatique en utilisant des ressources terminologiques.

I.3.4.2 .L'appariement requête- document

Le processus d'appariement document-requête permet de mesurer la pertinence d'un document vis-à-vis d'une requête. Ce processus (Matching) effectue une comparaison entre les représentants des documents et des requêtes construits lors de la phase d'indexation (entre la requête indexée et les descripteurs des documents).

La comparaison revient à calculer un score représentant la pertinence du document vis-à-vis de la requête. Cette valeur est calculée à partir d'une fonction ou d'une probabilité de similarité notée RSV (Q, D) (Retrieval Status Value), Où Q est une requête et D un document. Le processus d'appariement est lié au processus d'indexation et de pondération des termes.

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 [**13].**

Il existe deux types d'appariement **[14] :**

A) L'appariement exact:

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

B) L'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.

I.3.4.3. La reformulation de la requête

L'utilisateur exprime ses besoins sous forme d'une requête et la soumet ensuite aux SRI qui lui retournent des documents sensés être pertinents à sa requête. Cependant, Il est rare que l'utilisateur soit satisfait de la première réponse retournée par le SRI. Pour cela les chercheurs en RI se sont orientés vers l'intégration d'une étape supplémentaire dans le processus de recherche : la reformulation de la requête.

La reformulation de la requête est donc un processus ayant pour objectif de générer une nouvelle requête plus ciblée que celle initialement formulée par l'utilisateur. Ce processus consiste généralement à modifier la requête initiale de l'utilisateur via la réinjection de la pertinence (relevance feedback), pseudo-réinjection de la pertinence (pseudo-relevance feedback ou blind feedback).

A. La reformulation de requête par réinjection de pertinence :

On lui distingue deux façons :

- **Réinjection positive**: elle consiste à ajouter à la requête utilisateur des termes auxquels il n'aurait pas pensé et qui apparaissent dans les documents retournés pertinents.

- **Réinjection négative:** elle consiste à enlever de la requête utilisateur des termes qui apparaissent dans des documents retournés non pertinents.

Cette reformulation peut être automatisée, une fois que les documents pertinents ont été marqués par l'utilisateur.

B. La reformulation de requête par pseudo-réinjection de la pertinence

La reformulation par pseudo-réinjection de la pertinence (Blind Feedback ou encore Pseudo Relevance Feedback, notée PRF) utilise des techniques de réinjection automatique à l'aveugle pour construire la nouvelle requête. Ainsi, l'idée de base de la méthode PRF est basée sur l'hypothèse que les n premiers documents sont pertinents (documents pseudo pertinents) ainsi ils sont utilisés pour améliorer les performances de la RI. Le processus de la PRF consiste généralement à ajuster ou modifier le poids des termes (repondération) et ajouter les termes les plus pertinents qui sont extraits à partir des premiers documents retournés (considérés comme le contexte local de la requête).

I.4.Les modèles de la RI

Dit aussi modèles d'appariement document-requête, le rôle d'un modèle de recherche d'information est d'offrir un formalisme de représentation des documents et des requêtes, et déterminer les fonctions de calcul de similarité entre un document et une requête ainsi représentés, autrement dit, le modèle accomplit le rôle de fournir un cadre théorique pour la modélisation de la mesure de pertinence **[15].**

Depuis la naissance des premiers systèmes de RI dans les années 60, plusieurs modèles de recherche d'information ont été proposés dans la littérature. Sommairement, ils peuvent être classés en trois catégories principales : modèles booléens, modèles vectoriels et modèles probabilistes.

I.4.1 .Les modèles booléens

Les modèles booléens sont les premiers modèles utilisés en RI **[16].** Ils sont inspirés de la logique booléenne et de la théorie des ensembles. Ils englobent le modèle booléen classique, le modèle booléen étendu **[15]** et le modèle booléen flou **[17].**

I.4.1.1 Le modèle booléen classique

Il est reconnu pour faire une recherche très restrictive et obtenir, pour un utilisateur expérimenté une information exacte et spécifique. Il considère que les termes de l'index sont présents ou absents d'un document, en conséquence, les poids des termes dans l'index sont binaires c.à.d. $w_{ij}= \{0,1\}$.

Dans ce modèle, un document d est représenté comme une conjonction logique des termes non pondérés. Et une requête Q est composée de termes liés par trois connecteurs logiques ET($\wedge$), OU($\vee$) et NON ($\neg$).

Exemple : Q= (t1 ∧ t2) ∧ (t3 ∧ ¬t4), d=t1∧ t2∧ t3 ∧ … ∧ tn

Où d : le document.
Q : la requête
ti : terme d'indexation

La correspondance RSV (dj,Qk), entre une requête Qk et un document dj retourne un résultat binaire déterminée comme suit :

RSV (d_j,q_i)=1 si qi ∈ dj ; 0 sinon,
RSV (d_j,q_i ∧ d_k)=1 si RSV(d_j,q_k)=1 et RSV (d_i,q_k)=1 ; 0 sinon,
RSV (d_j,q_i ∨ d_k)=1 si RSV(d_j,q_k)=1 ou RSV (d_i,q_k)=1 ; 0 sinon,
RSV (d_j, ¬ q_i)=1 si RSV (d_j,q_i)=0 ; 0 sinon,

Sachant que q_i est un terme de la requête Q_k ,
La réponse à une requête Q_k est l'ensemble des documents qui sont similaires à cette requête :
Rep(QK) = {dj ∈ D | RSV (Dj, QK) = 1} Où D est l'ensemble des documents constituant le corpus.

Ce modèle se caractérise par sa simplicité et son caractère intuitif. Cependant, il possède les inconvénients suivants :

- ✓ Le système détermine un ensemble de documents non ordonnés comme réponse à une requête. Les usagers doivent encore fouiller dans cet ensemble pour trouver des documents qui les intéressent.
- ✓ Tous les termes dans un document ou dans une requête étant pondérés de la même façon (0 ou 1), ils ont la même importance dans le document.
- ✓ La difficulté au niveau de l'expression de la requête surtout en cas de requête complexe.

Par conséquent, de nos jours le modèle booléen classique n'est utilisé que dans très peu de systèmes. C'est plutôt des extensions de ce modèle qu'on utilise pour corriger ses lacunes ; le modèle booléen étendu qui intègre la pondération des termes de l'index permettant ainsi d'ordonner les documents retournés par le système et le modèle booléen flou quant à lui permet de modéliser les notions 'imprécision', 'incertitude' de l'information. De ce fait, un terme peut indexer totalement, partiellement ou pas du tout un document. De même un document peut être totalement, partiellement ou pas du tout similaire à une requête.

I .4.2 . Les modèles vectoriels

Appelés aussi modèles algébriques, ils reposent sur les bases mathématiques des espaces vectoriels. Leur forme d'implémentation la plus connue est le système SMART**[15]** qui est un des premiers travaux expérimentaux en RI. Trois variations principales y sont distinguées : le

modèle vectoriel standard, le modèle vectoriel généralisé, le modèle LSI (Latent Semantic Indexing) et le modèle connexionniste.

I.4.2.1 Le modèle vectoriel standard

Ce modèle est basé sur une intuition géométrique, les requêtes et les documents sont représentés dans l'espace vectoriel engendré par les termes d'indexation **[18].**L'espace est de dimension N (N étant le nombre de termes d'indexation de la collection de documents). Ainsi, chaque document et requête sont respectivement perçus comme un vecteur document et un vecteur requête :

$D_j = (w_{1j}, w_{2j}, \ldots, w_{nj})$; w_{ij}=poids du terme t_i dans le document D_j
$Q = (w_{1q}, w_{2q}, \ldots, w_{nq})$; w_{iq}=poids du terme t_i dans la requête Q.

Les termes de poids nul représentent les termes qui n'apparaissent pas dans le document alors que les poids positifs représentent les termes assignés.

Le modèle vectoriel permet de mesurer la similarité entre le document D et la requête Q en fonction du degré de corrélation vectorielle entre eux. Ce mécanisme assure un appariement partiel ou optimal (partial match, best match) entre les documents et la requête.

Les principales mesures de similarité utilisées dans ce modèle sont :

Produit scalaire : $$RSV(D_j, Q) = \sum_{i=1}^{n} w_{ij} * w_{iq} \quad (I.2)$$

La formule de Dice : $$RSV(D_j, Q) = \frac{2*\sum_{i=1}^{n} w_{ij}*w_{iq}}{\sum_{i=1}^{n} w_{ij}+\sum_{i=1}^{n} w_{iq}} \quad (I.3)$$

La formule de Jaccard : $$RSV(D_j, Q) = \frac{\sum_{i=1}^{n} w_{ij}*w_{iq}}{\sum_{i=1}^{n} w_{ij}+\sum_{i=1}^{n} w_{iq}-\sum_{i=1}^{n} w_{ij}*w_{iq}} \quad (I.4)$$

La formule de cosinus : $$RSV(D_j, Q) = \frac{\sum_{i=1}^{n} w_{ij}*w_{iq}}{\sqrt{\sum_{i=1}^{n} w_{ij}^2}*\sqrt{\sum_{i=1}^{n} w_{iq}^2}} \quad (I.5)$$

Le mécanisme de recherche consiste à retrouver les vecteurs documents qui s'approchent le plus du vecteur requête.
La figure suivante montre un exemple de représentation de trois documents et d'une requête.

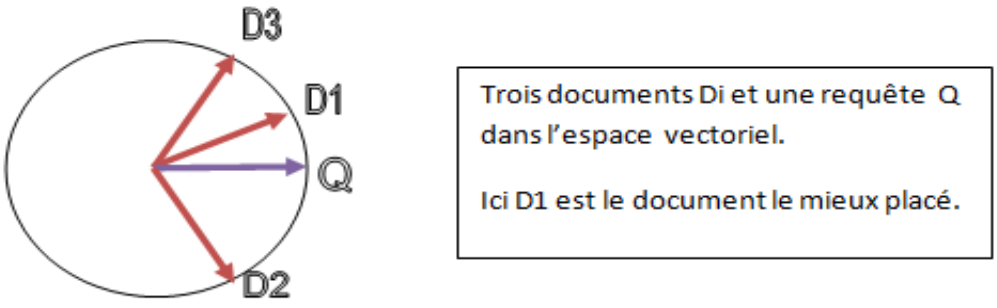

Figure I.3: Modélisation des vecteurs documents et requête

L'un des avantages du modèle vectoriel est le fait de trier les résultats d'une recherche, en plaçant en tête les documents jugés les plus similaires à la requête. Il permet aussi à l'utilisateur d'exprimer son besoin en information en langage naturel ou sous forme d'une liste de mots-clés contrairement au modèle booléen où les termes de la requête doivent être reliés par des connecteurs logiques, et donc il considère que les termes de l'index sont tous indépendants ce qui se présente comme étant l'inconvénient majeur du modèle vectoriel standard mais reste tout de même le modèle le plus utilisé en RI.

I.4.3. Les modèles probabilistes

Ces modèles reposent sur la théorie de la probabilité ainsi que des mesures statistiques. L'intérêt sur les modèles probabilistes a pris son envole à partir des années 1990, où les approches probabilistes se sont montrées performantes dans TREC qui est un programme d'évaluation des SRI (par exemple, le système OKAPI **[19]**). Ils englobent le modèle probabiliste général, le modèle de réseau différentiel (Document Network), et le modèle de langue.

I.4.3.1 Le modèle probabiliste général

L'idée de base du modèle probabiliste 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 de l'utilisateur. Pour ce faire, deux probabilités conditionnelles sont utilisées dans le processus de décision :

- P(R/D) : probabilité du document D sachant qu'il est pertinent pour la requête.
- P(NR/D) : probabilité du document D sachant qu'il n'est pas pertinent pour la requête.

À noter que les variables « document pertinent », « document non pertinent » sont indépendantes.
Ainsi, le score d'appariement entre le document D et la requête Q est donné par :

$$\text{RSV (D, Q)}= \frac{\text{P(R/D)}}{\text{P(NR/D)}} \tag{I.6}$$

En appliquant le théorème de Bayes et après simplification, on a :

$$\text{RSV (D,Q)}= \frac{\text{P(R/D)}}{\text{P(NR/D)}} \approx \frac{\text{P(D/R)}}{\text{P(D/NR)}} \tag{I.7}$$

Où P(D/R) (respectivement P(D/NR)) est la probabilité que le document appartienne à l'ensemble R des documents pertinents (respectivement à l'ensemble NR des documents non pertinents). Le document D répond à la requête Q si RSV (D, Q)>1
Il existe plusieurs méthodes pour l'estimation de ces différentes probabilités. La plus connue est celle du modèle BIR (Binary Independance Retrieval), où, on considère que la variable document D($t_1 = x_1$, $t_2 = x_2$, .. , $t_n = x_n$) tel que :

$$x_i = \begin{cases} 1 \text{ si ti indexe le document D} \\ 0 \text{ sinon} \end{cases} \quad \text{(I.8)}$$

La distribution des termes suit une loi de Bernoulli ; P (D/R) peut s'écrire alors :

$$P(D/R) = \sum_i p(t_i = x_i/R) = \sum_i p(t_i = 1/R)^{x_i} * p(t_i = 0/R)^{1-x_i} \quad \text{(I.9)}$$

En suivant la même procédure, on développe P(D/NR).
En posant :

- P_i = P (x_i=1/pert) entraine 1-p_i = P (x_i=0/pert)
- q_i = P (x_i=1/nonpert) entraine 1-q_i=P(x_i=0/nonpert)

RSV (D, Q) peut s'écrire, après transformation, comme suit :

$$RSV(D, Q) = \frac{p_i^{x_i} * (1-p_i)^{1-x_i}}{q_i^{x_i} (1-q_i)^{1-x_i}} \quad \text{(I.10)}$$

Et en passant par la fonction logarithmique, on obtient :

$$RSV(D, Q) = \sum_{i;x_i=1} \log \frac{p_i(1-q_i)}{q_i(1-p_i)} \quad \text{(I.11)}$$

L'obstacle majeur avec ce modèle est de trouver des méthodes pour estimer les probabilités utilisées pour évaluer la pertinence qui soient théoriquement fondées et efficaces.

I.4.3.2.Le modèle de langue

Les modèles probabilistes de langue sont exploités avec beaucoup de succès dans divers domaine : la reconnaissance de la parole **[43]**, la traduction automatique **[44],** la recherche d'information **[45],** etc.
L'utilisation des modèles de langue en RI remonte à 1998**[45].** 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 $P(Q|\, M_d)$ **[45]**. Le modèle de langue utilisé est souvent le modèle uni-gramme, la probabilité $P(Q|\, M_d)$ est alors exprimée ainsi :

$$P(Q|\, M_d) = \prod_{t \in Q} p(t|M_d) \quad \text{(I.12)}$$

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

$$P(t|\, M_d) = \frac{tf(t,d)}{|d|} \quad \text{(I.13)}$$

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

I.5. Evaluation des SRI

Dès l'apparition des premiers SRI, la pratique d'évaluation des systèmes est apparue ;les premières évaluations datent de 1953**[20].**

L'évaluation des SRI est abordée selon deux angles différents. L'un est dit « paradigme système », qui vise à évaluer les préférences 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 préférences intrinsèques du système ,en modélisant le comportement des utilisateurs en situation de recherche.

Nous présentons ci-dessous seulement l'approche basée « 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.

I.5.1.Les mesures d'évaluation

Le principale objectif d'un système de recherche d'information est de restituer à l'utilisateur tous les documents pertinents et de rejeter tous les documents non pertinents .Cet objectif est évalué à l'aide de différents mesures d'évaluation [**21].**
On présente ci-dessous les plus utilisées.

I.5.1.1.La précision

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

$$\text{Précision} = \frac{SP}{R} \qquad \text{(I.14)}$$

I.5.1.2.Le rappel

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

$$\text{Rappel} = \frac{SP}{P} \qquad \text{(I.15)}$$

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.

Les mesures de précision-rappel ne sont pas statiques non plus (c'est-à-dire qu'un système n'a pas qu'une mesure de précision et de rappel). Le comportement d'un système peut varier en faveur de précision ou en faveur de rappel (en détriment de l'autre métrique). Ainsi, pour un SRI, on a une courbe de précision-rappel qui a en général la forme suivante:

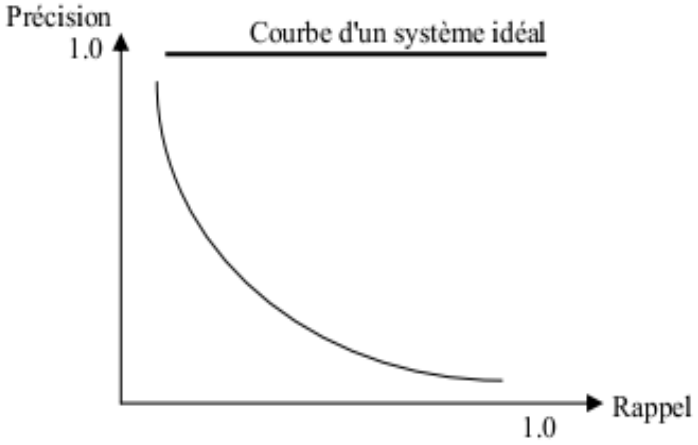

Figure I.4 : Courbe d'évaluation de la pertinence d'un SRI **[26].**

I.5.1.3.La précision moyenne non interpolée (MAP)

La précision moyenne non interpolée (Average Mean Précision) est calculée en deux étapes. D'abord on calcule la précision pour une requête donnée (APq), ainsi pour chaque document pertinent retrouvé on calcule sa précision ($pr(d_i)$) qui est égale au membre 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 :

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

Avec

$$Pr\,(d_i)=\begin{cases}\frac{r_{ni}}{ni} & si\ d_{ij}\ est\ retrouvé \\ 0 & sinon\end{cases} \qquad (I.16)$$

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

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

$$\text{MAP} = \frac{1}{M}\sum_{j=1}^{M} AP_{q_j} \quad \text{(I.17)}$$

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

I.5.2.Les collections de tests

Pour évaluer le degré de pertinence des réponses d'un système de recherche d'information à une requête, il est nécessaire de connaître l'ensemble des documents pertinents pour une requête donnée. C'est à cette fin que des collections de tests ont été élaborées. Une collection de tests comprend : des corpus de documents, une liste de requêtes prédéfinies et des jugements de pertinence.

Corpus de documents : c'est un ensemble de documents à indexer, il s'agit de l'ensemble des informations accessibles et exploitables sur lesquelles le système sera évalué. Un groupe d'experts dans un domaine participent généralement à la constitution d'un corpus documentaire homogène et cohérent couvrant le domaine. Dans le cas général et pour un souci d'optimalité, la base documentaire constitue des représentations simplifiées mais suffisantes pour ces documents. Ces représentations sont étudiées de telle sorte que la gestion (ajout suppression d'un document) ou l'interrogation (recherche) de la base se font dans les meilleures conditions de coût.

Requête : c'est une représentation d'un besoin d'information. Il est souhaitable que le protocole d'évaluation prend en compte une requête directement dans la forme où elle a été soumise au système, et non un besoin ou une question exprimée sous forme libre et détaillée (éventuellement beaucoup plus riche d'informations). Ceci restreint d'emblée les objectifs même de l'évaluation, qui ne permettra pas d'effectuer des mesures de performance relatives (entre SRI), si ces SRI n'offrent pas le même langage de requêtes et n'exploitent pas les mêmes enrichissements associés aux documents **[22].**

Jugements de pertinence: ils sont manuellement établis et constituent la liste des documents pertinents pour chaque requête. Ils peuvent être portés par l'utilisateur lui-même, ou par un observateur expert "extérieur " qui se concentrera plus facilement sur la " pertinence ". Ils peuvent prendre des formes booléennes variées (le document est pertinent ou il ne l'est pas) ou plus nuancées, généralement par l'usage de pondérations. L'utilisation de poids ou de pondérations implique la définition d'une échelle de mesure **[22].**

I.5.2.1.La collection TREC

La compagne TREC constitue à ce jour la compagne de référence dans le cadre de l'évaluation des systèmes de recherche d'information et cela depuis son lancement en 1992.

L'objectif de cette campagne est de proposer une plate-forme qui réunit des collections de test, des taches spécifiques et des protocoles d'évaluation pour chaque tache afin de mesurer les différentes stratégies de recherche.

L'initiative la plus importante actuellement pour la construction de collections de test est sans conteste TREC (Text Retrieval Conference) **[25].** TREC est plus qu'une collection de test, c'est un programme d'évaluation des SRI, initié en 1992 par le NIST (National Institute of Standards and Technology) aux Etats-Unis.

TREC fournit une plate-forme comportant des collections de test, des tâches spécifiques et des protocoles d'évaluation pour chaque tâche, pour l'évaluation et la comparaison d'expérimentations sur des collections volumineuses de textes. Il faut noter que les collections TREC représentent aujourd'hui un référentiel incontournable en RI.

Son objectif est de proposer un standard pour comparer les différents modèles de RI, indépendamment de la méthode de l'indexation ou bien du modèle qu'ils implémentent. Afin de mesurer l'efficacité des SRI de manière standard, il suffit de fournir une collection de test, des requêtes et les jugements à ces requêtes.

I.6. Conclusion

Dans ce chapitre, nous avons présenté les concepts de base de la recherche d'information. Ainsi, le processus de recherche et ses éléments de base tels que : le SRI, la collection et les requêtes qui expriment le besoin en information de l'utilisateur. L'architecture des systèmes de RI, les modèles de RI. Enfin l'évaluation des SRI.

Dans le chapitre suivant nous traitons de l'un des éléments clé de la RI à savoir la pondération des termes.

Chapitre II:

"Les facteurs de pondération"

II.1.Introduction

Tel qu'on l'a introduit dans le chapitre précédent, la pondération constitue la clé de voûte de la majorité des modèles et approches de RI, proposés depuis 1960.

Le but d'une fonction de pondération est d'affecter un poids à un terme dans un document pour traduire son importance dans ce dernier. Comme, l'efficacité d'un SRI est mesurée principalement par le rappel et la précision définis dans le chapitre précédent, la pondération a un grand impact sur ces deux mesures.

L'objectif de la fonction de pondération, est toujours de trouver les termes représentants au mieux le contenu du document, pour cela, plusieurs formules sont proposées dans la littérature et se basent sur des outils : statistiques **[27] [28]**, sémantiques **[30]** ou probabilistes **[33].** Les méthodes statistiques furent exploitées dès le début de développement de la RI, la loi de Zipf **[24]** et la conjecture de Luhn **[12]** furent les bases des différentes méthodes statistiques développées a cet effet.

Dans les années 1970, la fonction de pondération a été étendue à TF-IDF (Term Frequency - Inverted Document Frequency) afin de tenir compte de la spécificité du terme pour un document, et d'autres formules de pondération ont été développées dans les années 1990 telles que celle basée sue la distribution de Poisson et le modèle de langage.

Ce chapitre a pour objectif de présenter les principaux schémas de pondération. Principalement, nous présentons les facteurs utilisés dans la pondération. Les facteurs classiques, le facteur de proximité entre terme, le facteur de la structure de document et le facteur de position du terme dans un document.

II.2.Les facteurs de la pondération

Une bonne formule de pondération est celle qui assure à la fois un rappel et une précision élevés. Trois principales considérations apparaissent importantes afin d'atteindre cet objectif:

1) Le TF (Terme Frequency) : un terme très fréquent dans un document peut être important et de poids élevé, d'où la suggestion d'utiliser un facteur pour mesurer sa fréquence dans un document.

2) Le DF (Document Frequency) : Certains termes sont très fréquents dans la majorité des documents de la collection, ce qui peut affecter la performance de la recherche, d'où la nécessité d'une mesure qui va prendre en compte ce point dans toute la collection « idf » (inversed document frequency).Ainsi un terme avec une fréquence en document basse est un bon terme.

3) Un même terme dans des documents de tailles différentes peut avoir des poids différents, pouvant engendrer la considération de pertinence élevée dans les documents longs même s'ils ne le sont pas, alors il faut prendre compte de la taille document « dl » (document length).

Ces trois facteurs, nous les notons facteurs classiques. A coté de ces derniers, d'autres facteurs ont été explorés à savoir: la structure de document, la proximité des termes de la requête dans le document et la position du terme dans un document.

II.2.1.Les facteurs classiques de la pondération

Pour une bonne indexation et une bonne pondération, les aspects d'exhaustivité (description documentaire la plus complète) et de spécificité (meilleur discrimination entre les documents) sont pris en compte et sont équilibrés. L'équilibre entre l'exhaustivité et la spécificité d'un document se trouve dans le schéma de pondération « TF-IDF ». Cette approche est basée sur la combinaison de deux facteurs : la pondération locale (TF: Term Frequency) qui est la représentativité locale d'un terme au sein du document lui-même et la pondération globale (idf : Inverted Document Frequency) pour la représentativité globale au sein de la collection de documents.

II.2.1.1. La pondération locale TF

Elle permet de mesurer la représentativité et le poids local d'un terme dans un document. Plusieurs variantes de la fonction de pondération sont proposées:

- **La fonction binaire :** elle vaut 1 si le terme appartient au document en question et 0 sinon.
- **La fonction brute** : est une fonction proportionnelle à la fréquence d'occurrence d'un terme dans un document, elle retourne simplement le nombre d'occurrences du terme t_i dans le document D_j, notée tf_{ij} .
- **La fonction logarithmique :** elle combine la fonction brute avec un logarithme dans le but de diminuer l'effet de larges différences entre les fréquences des termes d'un document pour les rapprocher les uns des autres, elle est exprimée comme suit :

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

  Où

  α est un paramètre qui prend une valeur dans l'intervalle [0…1].
  tf_{ij} est la fréquence du terme t_i dans le document D_j.

  Cette fonction est proposée en 1988 par Salton & Buckley **[29]** et montre qu'un document à grande fréquence du terme t_i n'est pas plus pertinent qu'un autre qui contient plusieurs termes de la requête un petit nombre de fois.
- **La fonction normalisée :** elle permet de réduire les différences entre les poids accordés aux termes d'un document, en accordant une valeur de poids minimale (par exemple 0.5 dans la seconde formule) pour tous les termes du document et un poids ne dépassant pas une valeur maximale pour les termes présents plusieurs fois, elle est exprimée de deux manières suivantes :

$$w_{ij} = \frac{tf}{\max tf} \quad \text{Ou} \quad w_{ij} = 0.5 + 0.5\frac{tf}{\max tf} \qquad \text{(II.2)}$$

Où : w_{ij} : le poids du terme t_i dans le document D_j.

tf : la fréquence du terme t_i dans le document D_j.
max tf : la fréquence maximale des termes du document D_j.

II.2.1.2.La pondération globale DF

Contrairement à la pondération locale, la pondération globale mesure le poids du terme dans la collection toute entière, ainsi cette méthode améliore les performances dans le cadre de la recherche d'informations **[31].** Ainsi, un terme très fréquent dans la collection n'aura pas le même impact que le moins fréquent, un poids plus important est affecté aux termes à moindre fréquence dans la collection car ils ont un pouvoir plus discriminatoire entre les documents que ceux ayant des fréquences élevées. Dans cette méthode, le facteur « IDF» est calculé de deux manières suivantes:

$$\text{IDF} = \log \frac{N}{n_i} \tag{II.3}$$

$$\text{IDF} = \frac{(N - n_i)}{n_i} \tag{II.4}$$

Où : n_i : est le nombre de documents contenant le terme t_i.
N : est le nombre de documents dans la collection.

II.2.2.3 La longueur du document

Certains documents sont plus longs que d'autres, ce qui induit:

- **La verbosité** : les documents les plus longs traitent plus longuement du même thème ce qui implique que les termes sont plus fréquents.
- **Les variations thématiques** : les documents les plus longs traitent plus de thèmes, ce qui implique l'utilisation d'une terminologie plus variée.

Les documents les plus longs contiennent plusieurs termes, avec des valeurs «TF» plus élevées et des termes plus distincts. Ce facteur augmente le score de ces documents vis à vis des requêtes.

Ainsi, les documents les plus longs peuvent être globalement regroupées en deux catégories: la première catégorie contienne des documents détaillés qui répètent essentiellement le même contenu, la longueur du document ne modifie pas les poids relatifs des termes différents. La deuxième catégorie possède des documents couvrant de multiples sujets différents, dans lesquels les termes de recherche correspondent probablement à de petits segments des documents, les poids relatifs des termes sont tout à fait différent d'un court document unique qui correspond aux termes de la requête. La compensation de ce phénomène est une forme de la longueur du document qui est indépendante des fréquences du terme et documents **[32].**

Si plusieurs documents ont la même pertinence, on préfère alors le plus court.

L'introduction de ce facteur dans les schémas de pondération est donnée dans les modèles BM25 et TF-IDF présentés dans les sections suivantes.

II.2.2 Exemples de schémas de pondération classique

II.2.2.1.La formule Okapi BM25

II.2.2.1.1.Définition

Okapi BM25 (« Best Matching » ou meilleur arrangement) est une fonction de classement des documents en fonction de leurs pertinences, elle est développée en 1976 par S.E. Robertson et al **[19] [27].** Le nom de la fonction est BM25, cependant elle est généralement appelée « Okapi BM25 » référençant le Système de Recherche d'Information OKAPI, étant le premier à l'avoir appliqué. La BM25 se présente sous de multiples variantes et instances avec un ensemble de paramètres partagés.

II.2.2.1.2. La formalisation du modèle BM25

Le modèle BM25 est un modèle basé sur la représentation en sac de mots, c'est-à-dire il effectue la correspondance entre un document et les termes de la requête sans prendre en compte les liens existants entre termes. Ce n'est pas un modèle unique, mais en fait c'est une famille de modèles de pondération, différents en termes de paramètres. Une des instanciations la plus connue de ce modèle est la suivante :
Étant donné une requête Q, contenant des mots-clés q_1…..q_n, le score d'un document D est donné comme suit :

$$\text{Score}_{\text{BM25}}(D, Q)=\sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i,D).(K_1+1)}{f(q_i,D)+K_1.(1-b+b.\frac{|D|}{avgdl})} \quad \text{(II.5)}$$

Où $f(q_i, D)$ est la fréquence du terme q_i dans le document D. $|D|$ est la longueur du document D, et $avgdl$ est la longueur moyenne des documents de la collection. K_1 et b sont des paramètres, généralement choisis, en l'absence d'une optimisation avancée, prennent les valeurs suivantes : K_1[1.2,2.0] et b=0.75. $IDF(q_i)$ est la fréquence inverse en document du terme q_i , il est généralement calculé comme suit:

$$\text{IDF}(q_i) = \log\frac{N-n(q_i)+0.5}{n(q_i)+0.5} \quad \text{(II.6)}$$

Où

$n(q_i)$ est le nombre de documents qui contient le termeq_i.
N est le nombre de documents dans la collection.

II.2.2.2.Le modèle TF-IDF

II.2.2.2.1.Définition du modèle TF-IDF

Le modèle TF-IDF (*Term Frequency-Inverse Document Frequency*) est une méthode de pondération souvent utilisée en recherche d'information. La mesure TF-IDF combine les deux critères suivants: l'importance du terme dans un document (donné par TF), et le pouvoir de discrimination de ce terme (donné par IDF). Ainsi, un terme qui a une valeur de TF-IDF élevée doit être à la fois important dans ce document, et aussi il doit apparaître dans peu d'autres documents **[34].**

II.2.2.2.2.Formalisation

Parmi les formules résultantes de cette combinaison, nous citerons les suivantes :

$$w_{ij} = Tf * \log\left(\frac{N}{n_i}\right) \quad \text{(II.7)}$$

$$w_{ij} = (1 + \log(Tf_{ij})) * \log\left(\frac{N}{n_i}\right) \quad \text{(II.8)}$$

$$w_{ij} = (0.5 + 0.5 * \frac{Tf_{ij}}{maxTf}) * \log\left(\frac{N-n_i}{n_i}\right) \quad \text{(II.9)}$$

Où w_{ij}: est le poids du terme t_i dans le document D_j.
tf_{ij} : est la fréquence du terme t_i dans le document D_j.
n_i: est le nombre de documents contenant le terme t_i.
N: est le nombre de documents dans la collection.
$max\ tf$: est la fréquence maximale des termes du document D_j.

La mesure TF-IDF est une bonne approximation de l'importance d'un terme dans un document, particulièrement dans des corpus de documents de tailles homogènes. Cette mesure a eu en revanche un succès très limité dans les corpus de tailles très variables du fait qu'elle ne tient pas en compte d'un aspect très important du document : sa longueur.

Le problème posé est que les termes appartenant aux documents longs apparaissent très fréquemment et l'emportent en poids sur les termes appartenant à des documents moins longs; par conséquent, les documents longs auront alors plus de chance d'être sélectionnés. Pour y remédier à ce problème on a introduit la normalisation par la taille de document **[35] [36]**.

Parmi les formules qui ont introduit la taille de document on trouve la formule TF-IDF de Sparck Jones **[34]** qui est la base de toutes les autres formules **[39] [36]**. La formule TF-IDF de Spark Jones **[34]** est donnée comme suit :

$$w = \frac{tf*(K_1+1)*\log\left(\frac{N}{n}\right)}{K_1*\left((1-b)+b*\frac{dl}{avdl}\right)+tf} \quad \text{(II.10)}$$

Où : b : constante

dl: taille d'un document

$avdl$: taille moyenne des documents

$h_1 eth_2$: paramètres dépendants de la nature des requêtes et documents.

Les paramètres *K1* et *b* présentent respectivement l'influence du poids par rapport à la fréquence et l'effet de longueur du document.

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 dans**[28]** est récemment utilisé sous différents points de vues : la structure de document, la position chronologique des termes dans le document et la proximité des termes de la requête dans un document. Nous présentons, ci-dessous ces trois facteurs.

II.2.3 Le facteur de proximité

L'intuition considérée dans les proximités des termes de la requête dans un document est la suivante :

<< Un bon document est celui dans lequel les termes de la requête apparaissent proche les uns des autres >>.

Par exemple, considérant la requête **<< moteur** de **recherche>>** et les deux documents suivants qui contiennent les termes de la requête :

d_1 : << Un **moteur de recherche** est une application web permettant de retrouver des ressources …>>

d_2 :<< Le développement d'un **moteur** utilisant uniquement l'énergie solaire nécessite de la **recherche** …>>.

Intuitivement, le document d_1 est plus approprié à la requête car les termes de la requête y apparaissent d'une manière contiguë dans ce document. Par contre, le document d_2 contient les termes de la requête éloignés les uns des autres, et leur combinaison ne fait pas référence au sens du terme recherché << moteur de recherche >>.

Il ya plusieurs travaux qui ont intégré cette notion de proximité principalement nous présentons deux travaux :

II.2.3 .1.Le modèle MRF [38]

Metzler et Croft **[38]** ont élaboré un cadre formel pour intégrer la proximité (dépendance) 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 séquentiel (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 :

$$Score_{MRF}(q,d) = \lambda_u \sum_{t_i \in q} \log\big(P(q_i|d)\big) + \lambda_s \sum_{t_i \in q} \sum_{t_i \in q\ etj=i+1} \log\left(P(< q_i, q_j > w|d)\right) +$$

$$\lambda_f \sum_{t_i \in q} \sum_{t_i \in q\ etj \neq i} \log\left(P\big(< q_i, q_j > w\big|d\big)\right) \qquad \text{(II.11)}$$

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 l'occurrence de la paire <q_i,q_j> dans le document d.

II.2.3 .2.Modèle de langue de position

Lv et Zahi **[39]** ont proposé un modèle de langue nommé <<positional Language Model :PLM>>
Dans ce dernier, un modèle de langue pour chaque position du document est crée, il est exprimé ainsi :

$$P(t\,|d,i) = \frac{c'(t,i)}{\sum_{t' \in V} c'(t',i)} \qquad \text{(II.12)}$$

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 réalisée en utilisant une fonction de densité (Gaussian Kernel, Trangle 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-gamma.

Ces approches utilisant la notion de proximité pour intégrer les relations entre termes ont montré que cette source d'information est utile pour la RI.

II.2.4 Le facteur de la structure de document

Etant un format d'échange de données important et une norme de stockage d'information, XML est maintenant largement utilisé, c'est pour cela que les praticiens ont jugé utile d'exploiter la structure interne du document pour améliorer la performance de la recherche d'information **[37].**

Parmi les études effectuée dans le domaine on trouve celle de Rebertson et al **[37]** qui ont proposé une combinaison linéaire des fréquences de terme basée sur le modèle BM25, ce nouveau modèle proposé est nommé BM25E.

II.2.4 .1.Le modèle BM25E

En partant de l'hypothèse qu'un élément peut être traité comme un document. Le modèle BM25 a été adapté pour pondérer les termes dans les éléments, sauf qu'il peut hériter des informations d'autres éléments du document. La fonction proposée nommé BM25E est une fonction qui a été dérivée de la fonction BM25 **[19] [27].**

Ainsi, chaque élément pouvant être un nœud feuille qui est constitué de l'information textuelle, ou un nœud interne donc représente un sous-arbre xml ou encore un nœud unique racine qui représente tout le document, donc l'élément hérite des informations d'autres éléments spécialement lorsqu'il s'agit d'un nœud feuille ou interne . Les détails de ce modèle sont les suivants :

Supposons que nous avons :

tf : la fréquence de terme dans l'élément *e*.
el : la taille de l'élément.
$tf_{e,j}$: la fréquence du nième terme dans l'élément *e*.
el_e : la taille de l'élément.
df_j :le nombre d'élément qui contient le terme j.
avel : la taille moyenne de l'élément de la collection.
La formule proposée et qui utilise les paramètres cité ci-dessus, est définie comme suit :

$$w_j(e,d,C) = \frac{(k_1+1)tf_{e,j}}{k_1((1-b)+b\frac{el_e}{avel})+tf_{e,j}} log\frac{N-df_j+0.5}{df_j+0.5} \quad \text{(II.11)}$$

Où $w_j(e,d,C)$ est le poids du terme t_j dans l'élément *e* appartenant au document d.

II.2.5 Le facteur de position

En 1950 Luhn **[28]** a proposé que « la fréquence du terme dans un document donne une bonne indication sur l'importance du terme et que la position relative du terme dans une phrase détermine l'importance de la phrase dans le document. »

Luhn a donc combiné ses deux propositions pour mesurer l'importance du terme dans la phrase et pour pouvoir ensuite mesurer l'importance de la phrase dans le document.

Suite à la proposition de Luhn, de nouveaux travaux sont apparus exploitant la position du terme dans le document. Parmi ces travaux on trouve celui de D.Troy et al **[49]**; où ils déclarent (proposition) qu'un bon document est celui où les termes de la requête apparaissent dans les premières position de ce document. Le modèle correspondant à cette proposition est nommé, le modèle CTR, pour Chronological Term Rank. Ce modèle est décrit en détail dans le chapitre suivant.

II.3.conclusion

Nous avons abordé dans ce chapitre un des concepts clé de la RI, à savoir la pondération des termes. Plus exactement, nous avons étudié les différents facteurs utilisés lors de celle-ci. Ces facteurs sont la fréquence locale (TF), la fréquence globale(DF), la longueur de document (DL), la structure de document, la proximité des termes de la requête et enfin la position des termes. Nous avons aussi présenté les travaux clés utilisant ces facteurs.

Dans le chapitre suivant nous nous intéressons plus particulièrement à l'extension et à l'expérimentation du modèle CTR intégré aux modèles TF-IDF et BM25.

III.1.Introduction

Nous avons abordé dans le chapitre précédent les différents facteurs de pondération, dans ce chapitre nous allons présenter en premier temps le modèle CTR en détail, où nous spécifions l'intuition sur laquelle il se base et sa formalisation mathématique.
Dans un second temps, nous présentons l'extension proposée pour ce modèle et sa formalisation.

L'environnement de la mise en œuvre des deux modèles ainsi définis est ensuite présenté, à savoir les outils et les langages utilisés.
Enfin, nous présentons les résultats expérimentaux obtenus sur deux collections de test TREC AP88 et WT10g.

III.2.Présentation du modèle à implémenter

Dans cette section nous décrivons en premier le modèle CTR : son intuition, sa formalisation ainsi que son intégration au sien du modèle BM25, ensuite, nous présentons l'extension du modèle CTR.

III.2.1.Le modèle CTR

Le modèle Chronological Term Rank (CTR) comme nous l'avons introduit dans le second chapitre est un modèle basé sur l'exploitation de la position des termes de la requête dans un document. Ainsi, en plus des facteurs de pondération classiques (TF, IDF) il intègre ce facteur de position pour classer les documents **[35].**

III.2.1.1 Intuition du modèle CTR

L'intuition de ce modèle se base sur le fait que les auteurs des documents utilisent les termes pertinents ou porteurs de sens juste au début du document pour exprimer leur idée, ainsi un terme qui apparait juste au début d'un document est un bon terme. Cette pratique est notamment observée dans le domaine scientifique (articles scientifiques) et le domaine journalistique.

III.2.1.2 Formalisation

Le modèle CTR est défini comme suit: Soit un document $D = (t_1 \dots \dots \dots t_n)$ où t_i sont les termes du document ordonné selon leur apparition dans le document original.
tr_t: = i ou *tr* est le rang du terme t qui indique la première apparition de ce terme dans le document.

Exemple :
Soit le document D suivant :

Position	1	2	**3**	4	5	6	**7**	8	**9**	10
Terme	t_3	t_4	$\boldsymbol{t_1}$	t_2	t_3	t_4	$\boldsymbol{t_1}$	t_5	$\boldsymbol{t_1}$	t_3

Pour ce document tr_{t_1}=3 pour le terme t_1 , car la première apparition du terme t_1 dans le document D est à la position 3; les autres apparition de ce terme sont : 7 et 9.

III.2.1.3 Considérations fonctionnelles

Pour l'intégration de CTR, le modèle BM25 est utilisé comme modèle de base dans la fonction d'appariement, cette dernière est formulée comme suit :

$$Score(q,d)_{BM25} = \sum_{t \in d \cap q} ln \frac{N - df + 0.5}{df + 0.5} . \frac{tf}{0.5 + 1.5 . \frac{dl}{avdl} + tf} \qquad \text{(III.1)}$$

En plus du tf et du df comme défini ci-dessus,
t : est un terme de la requête
N : est le nombre de documents dans la collection
dl : est la longueur du document
$avdl$: est la longueur moyenne des documents de la collection.

L'intégration du modèle CTR au sein du modèle BM25 et fait soit de manière multiplicative ou Additive.

- L'additive est réalisée par l'ajout du facteur CTR à la fréquence.
- Le Multiplicatif est réalisé par la multiplication du facteur CTR avec la fréquence.

Le modèle CTR noté (R) se calcule de différentes manières comme présentées dans le tableau ci dessous :

Formule de base		
	Formules d'intégration du CTR au BM25	***Description***
A	$\sum_{t\in d\cap q} ln\frac{N-df+0.5}{df+0.5}.\frac{tf}{0.5+1.5.\frac{dl}{avdl}+tf}.R$	Multiplication
B	$\sum_{t\in d\cap q} ln\frac{N-df+0.5}{df+0.5}.(\frac{tf}{0.5+1.5.\frac{dl}{avdl}+tf}+R\ )$	Addition
	R	
	Formules de calcul du CTR	
a	$H.(1-\frac{tr-1}{dl})$	
b	$1+(H.\left(1-\frac{tr-1}{dl}\right))$	
c	$(1-H)+(H.\left(1-\frac{tr-1}{dl}\right))$	
d	$H.\frac{1}{tr}$	
e	$H.\frac{1}{\log(tr)}$	
f	$H.(1-\frac{1-tr}{maxdl})$	
g	$H.(1-\frac{\log(tr+9)}{\log(dl+10)})$	
h	$H.(1-\frac{\log(\left(\frac{tr-1}{D}\right)+10)}{\log\left(\left(\frac{dl}{D}\right)+10\right)})$	
i	$H.(1-\frac{\log(\left(\frac{tr-1}{D}\right)+10)}{\log\left(\left(\frac{maxdl}{D}\right)+10\right)})$	
j	$H-(H.D.\frac{\log(\left(\frac{tr-1}{30}\right)+10)}{\log\left(\left(\frac{dl}{30}\right)+10\right)})$	

Tableau III.1. L'intégration du modèle CTR au sein de BM25 sous ses différentes formules.

Ce modèle a été expérimenté sur plusieurs collection TREC, entre autres la collection WSJ90-92. Les résultats obtenus ont été encourageants, il a donné des améliorations par rapport au modèle BM25.

Le meilleur résultat est donné par la formule « j » présentée dans le tableau, utilisant l'addition (A).

III.2.2 Les extensions du modèles CTR

Nous proposons d'étendre le modèle CTR présenté précédemment selon deux directions :

1. La première est basée sur l'intuition suivante : « les auteurs de documents utilisent les termes pertinents ou importants au milieu du document, car c'est à ce niveau qu'ils développent plus leurs idées», ainsi un terme qui apparait au milieu du document doit voir son importance augmenter.

Pour la formalisation de cette extension $(tr_1)_t$ = j ou tr_1 est le rang du terme t qui indique la position moyenne de ce terme dans le document.

Exemple : Soit le document D présenté précédemment :

Position	1	2	**3**	4	5	6	**7**	8	**9**	10
Terme	t_3	t_4	$\boldsymbol{t_1}$	t_2	t_3	t_4	$\boldsymbol{t_1}$	t_5	$\boldsymbol{t_1}$	t_3

Ainsi, pour ce document tr_1 = 7 pour le terme t_1 , car ce terme apparait dans les positions suivantes : 3,7 et 9, donc on prend la position moyenne.

2. La deuxième extension est basée sur l'intuition suivante : « généralement les auteurs de document concluent leur écrits par des termes porteur de sens et importants », donc les termes qui apparaissent à la fin du document doivent voir leur importance augmenter.

Pour la formalisation de cette extension $(tr_2)_t$ = k ou tr_2 est le rang du terme t qui indique la dernière position de ce terme dans le document.

Exemple : Soit le document D présenté précédemment :

Position	1	2	**3**	4	5	6	**7**	8	**9**	10
Terme	t_3	t_4	$\boldsymbol{t_1}$	t_2	t_3	t_4	$\boldsymbol{t_1}$	t_5	$\boldsymbol{t_1}$	t_3

Ainsi, pour ce document tr_2 = 9 pour le terme t_1 . Car ce terme apparait dans les positions suivantes : 3,7 et 9, donc on prend la dernière position.

III.3.Environnement de travail

Dans ce qui suit, nous allons présenter notre environnement technique et spécifier les différents outils utilisés, Nous commençons par la plate-forme Terrier sous laquelle nous avons implémenté le modèle CTR et ses extensions, puis JAVA que nous avons utilisé comme langage de programmation et NetBeans qui est l'outil sur le quel nous avons programmé.

III.3.1.présentation de la plate forme terrier

Terrier, TERabyte RetrlEveR est un système de recherche d'information robuste et efficace qui offre une plate-forme idéale pour manipuler de grandes quantités de données « jusqu'à 25 millions de documents » .Il offre en plus un cadre pour l'évaluation des résultats de recherche pour différentes applications. Développé par le département d'informatique de l'université Glasgow de Scotland. Il est utilisé avec succès pour la recherche Ad-hoc, la recherche web et la recherche inter langages dans des environnements centralisée et distribués **[47].**
Terrier est un logiciel open source entièrement écrit en java. Selon des études comparatives sur les SRI en open sources effectuées par Ricardo Baeza yates [**46]**, Terrier fait parti des trois meilleurs systèmes dans un environnement java, les deux autres sont MG4 et Lucene.

Comme tous les systèmes de RI, Terrier possède les principaux processus suivants :
L'indexation, la recherche et l'évaluation, comme l'indique la figure suivante.

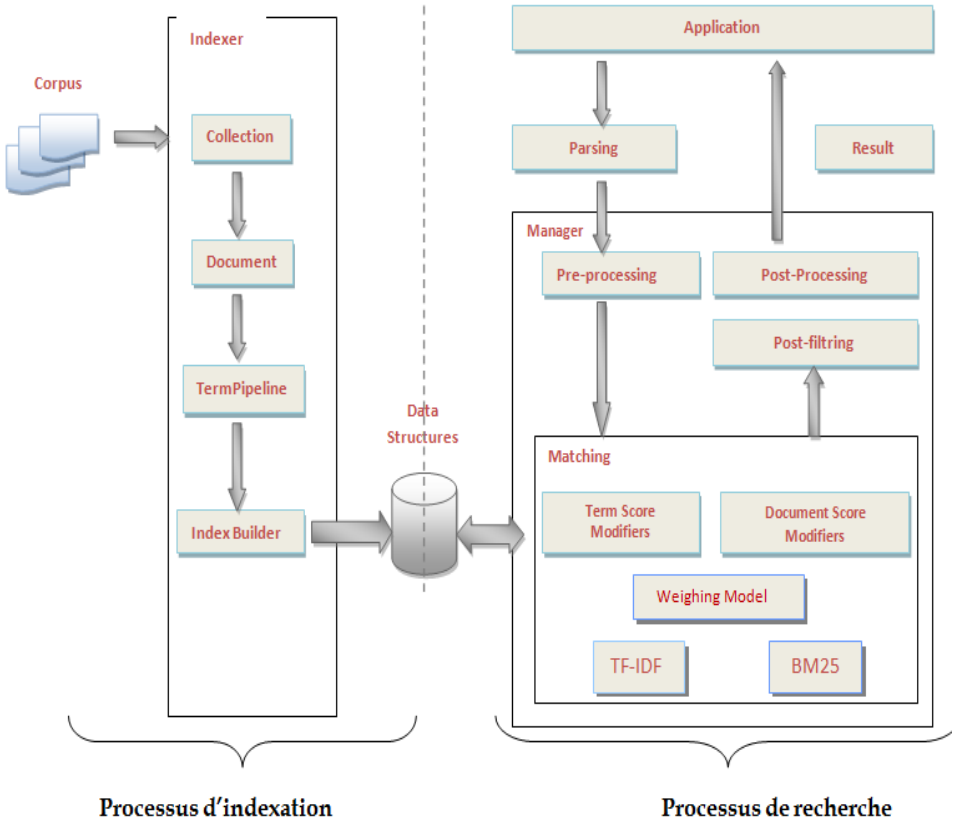

Figure III.1 : Architecture de Terrier

- **Le processus d'indexation de Terrier :**

L'indexation dans Terrier est divisée en quatre procédures :

1) La première procédure consiste à prendre en entrée le corpus à indexer et d'envoyer document à document pour la seconde procédure.
2) Le second traitement consiste à dé-baliser le document reçu (enlever les balises) et le passer au Tokenizer qui extrait les termes du document, et les envois à la troisième procédure
3) Dans la troisième étape, les termes reçus sont traités par le TermPipeline (l'élimination des mots vides et la lemmatisation des termes).
4) La dernière étape consiste à construire les structures de l'index.

- **Le processus de recherche de Terrier:**

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

1) **Parsing :** qui se charge d'extraire les tokens de la requête.
2) **Pre-processing** : qui applique le TermPipeline à la requête : éliminer les mots vides et lemmatiser les termes.
3) **Matching** : qui est responsable de l'initialisation de WeightingModel et du calcul des scores entre la requête et les documents.
4) **Post-filtrage** : filtre les documents pertinents pour la requête.
5) **Post-traitement** : reclasse les documents pertinents s'il y a une extension de la recherche comment l'ajout d'un nouveau modèle.

III.3.2 Le langage Java

Afin de réaliser notre application, nous avons utilisé le langage java, qui est un langage de programmation moderne développé par Sun Microsystems (aujourd'hui racheté par Oracle).

Java est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du langage C. Ses caractéristiques ainsi que la richesse de son écosystème et de sa communauté lui ont permis d'être très largement utilisé pour le développement d'applications de types très disparates. Java est largement utilisée pour le développement d'applications d'entreprises et mobiles. C'est un langage très utilisé, notamment par un grand nombre de programmeurs professionnels, ce qui en fait un langage incontournable actuellement.

Le Java est langage très développé doté d'une riche bibliothèque de classes, comprenant la gestion des interfaces graphiques (fenêtres, boites de dialogues, contrôles, menus, graphismes), et la gestion des exceptions.

Une de ses plus grandes forces est son excellente portabilité, java est un langage multi-plate forme, qui permet aux concepteurs d'écrire un code capable de fonctionner dans tous les environnements. Pour cela, il suffit que l'environnement possède une JVM (Java

Virtual Machine). En plus, il permet d'interagir aisément avec le Web et d'Accéder et fournir des services simplement au travers du réseau.

Le code source Java est compilé en byte code correspondant aux instructions de la machine virtuelle(JVM). La JVM est un programme source compilé pour chaque type de couple (machine physique + système d'exploitation).

La machine virtuelle fournit un environnement de programmation standard caractérisée par :

- Gestion de la mémoire (ramasse-miette)
- Gestion des entrées / sorties (disque, écran, clavier, etc.)
- Interface avec le système d'exploitation

III.3.3.Présentation de l'environnement de développement(NetBeans)

NetBeans est un environnement de développement intégré (EDI), placé en open source par sun. En plus de java, NetBeans permet également de supporter différents autres langages ,comme Python ,C++, javaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristique d'un IDE moderne (éditeur en couleur, projet multi-langage, refactoring, éditeur graphique d'interface et de page web).
Conçu en java, NetBeans est disponible sous Windows, LINUX ,Solaris (sur *86 et SPARC),Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java).Un environnement Java Développent Kit JDK est requis pour les développements en Java NetBeans constitué par ailleurs une plate forme qui permet le développement d'applications spécifique (bibliothèque Swing (Java)).L'IDE NetBeans s'appuie sur cette plate forme **[48].**

III.4. L'implémentions du modèle CTR et ses extension sous terrier

Afin d'implémenter le modèle CTR et ses extensions sous Terrier nous avons exploité certaines classes existantes comme nous avons étendu d'autres classes afin de supporter ce modèle.
Nous avons passé par plusieurs étapes pour obtenir la position du terme dans le document, ensuite nous avons étendu, notamment, les classes de terrier qui sont : « Matching », « WeightingModels ».
Une fois les positions d'un terme de la requete sont obtenues on les passe de la classe Matching à la classe (abstraite) WeightingModels;
Nous avons ensuite créer trois classes qui étendent le WeightingModels: la première réalise le modèle CTR de base; la seconde réalise la première extension du modèle CTR et en fin la dernière réalise la deuxième extension du modèle CTR.

Nous illustrons dans le schéma suivant les classes étendues et développées pour la mise en œuvre de notre travail sous Terrier.

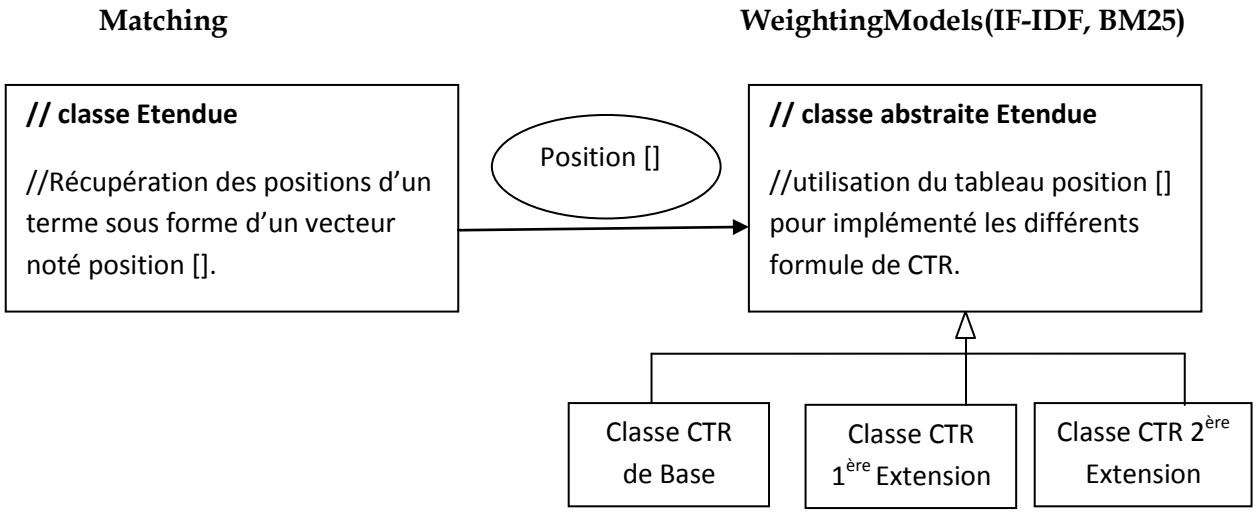

Figure III.2 : Implémentation du modèle CTR et extension sous Terrier

III.5 Résultats et expérimentations

III.5.1 Collections de test utilisées

Nous avons mené nos expérimentations en utilisant les collections de test TREC AP88 (Associated Press newswire) et WT10g.

Pour la recherche nous avons utilisé 100 requêtes des topics numérotées « 51-151 » sur la collection AP88, où le champ « *title* » est considéré, et aussi 100 requêtes des topics numérotées de « 451-550 » sur la collection WT10g.

Collection	Nombre de documents Dans la collection	Nombre de terme dans la collection	Taille moyenne d'un document
AP88	79919	144186	235 .085
WT10g	1692096	3575321	

Tableau III.2 : Description des collections de test utilisées.

III.5.2.Résultats d'évaluation

L'étape évaluation se déroulera de la façon suivante :
En premier lieu nous effectuons l'évaluation des deux modèles de pondération de base (BM25 et TF-IDF). Puis, nous évaluons le modèle CTR en l'intégrant aux deux modèles. Et Enfin, nous évaluons les extensions du modèle CTR.

III.5.2.1.Résultats obtenus avec les modèles de pondération (TF-IDF) et (BM25)

Pour chacun des deux modèles de pondération, TF-IDF et BM25, nous avons effectué une évaluation des résultats de recherche obtenus ces modèles en utilisant la mesure MAP (Mean Average Precision) et nous avons obtenu les résultats résumés dans le tableau suivant :

Collection	AP88	WT10g
MAP TF-IDF	0.1184	0.1469
MAP BM25	0.1296	0.1485

Tableau III.3 : Résultats des modèles TF-IDF et BM25.

Nous remarquons à partir de ce tableau que le modèle BM25 donne de meilleurs résultats que le modèle TF-IDF, ce qui indique que la modélisation probabiliste de la RI est bien réussie que la modélisation vectorielle.

III.5.2.2 Résultats du modèle CTR

Dans cette étape, nous allons présenter les résultats obtenus du modèle CTR et cela en l'intégrant aux deux modèles de pondération TF-IDF et BM25.

Nous allons tester les deux formule des deux modèles en rajoutant un paramètre R soit en addition ou en multiplication au deux formule de pondération, dont R (les formules de calcul de R sont données dans le tableau III.1). Les deux paramètres essentiels sont H qui prend des valeurs de 0 jusqu'à 1 avec un pas de 0.1 et le paramètre D détermine le pourcentage de H qui prend des valeurs de 0 jusqu'à 1 avec un pas de 0.1 (Uniquement pour les formules h, i et j), les autres formules nécessitent uniquement le paramètre D.

Les résultats obtenus sont résumés dans les deux tableaux suivants :

Collection		**AP88**	**WT10g**
Formule « a » « Addition »	**MAP CTR et TF-IDF**	0.1472	0.1703
	MAP CTR et BM25	0.1039	0.1582
	Valeur Paramètre H TF-IDF	H =0.4	H=0.8
	Valeurs de H BM25	H =0.5	H=0.8 et 1.0
	Taux d'amélioration % TF-IDF	+24.32%	+15.929
	Taux d'amélioration % BM25	-24.73%	+1.077%
Formule « a» «Multiplication»	**MAP CTR et TF-IDF**	0.1423	0.1509
	MAP CTR et BM25	0.1252	0.1533
	Valeur Paramètre H TF-IDF	H=0.1…1.0	H=0.1…1.0
	Valeurs de H BM25	H=0.1…1.0	H=0.1…1.0
	Taux d'amélioration % TF-IDF	+20.18%	+2.722%
	Taux d'amélioration % BM25	-3.514%	+3.232%
Formule « b » « addition »	**MAP CTR et TF-IDF**	0.0764	0.1658
	MAP CTR et BM25	0.1886	0.1548
	Valeur Paramètre H TF-IDF	H=0.5	H=0.7et0.8
	Valeurs de H BM25	H=0.9	H=1.0
	Taux d'amélioration % TF-IDF	-54.97%	+12.865%
	Taux d'amélioration % BM25	+37.577%	+4.242%

Formule « b» «multiplication»	**MAP CTR et TF-IDF**	0.1418	0.1597
	MAP CTR et BM25	0.1218	0.1686
	Valeur Paramètre TF-IDF	H=1.0	H=0.7
	Valeurs de H BM25	H=1.0	H=0.9
	Taux d'amélioration % TF-IDF	+19.763%	+8.713%
	Taux d'amélioration % BM25	-6.403%	+13.535%
Formule « c» « addition »	**MAP CTR et TF-IDF**	0.0934	0.1699
	MAP CTR et BM25	0.1779	0.1582
	Valeur Paramètre TF-IDF	H=0.8	H=0.8
	Valeurs de H BM25	H=0.9	H=1.0
	Taux d'amélioration % TF-IDF	-26.766%	+15 .656%
	Taux d'amélioration % BM25	+37.268%	+6.531%
Formule « c » «multiplication»	**MAP CTR et TF-IDF**	0.1444	0.1596
	MAP CTR et BM25	0.1262	0.1683
	Valeur Paramètre H TF-IDF	H=0.7	H=0.4
	Valeurs de H BM25	H=0.9	H=0.4
	Taux d'amélioration % TF-IDF	+21.959%	+8.645%
	Taux d'amélioration % BM25	-2.623%	+13.333%
Formule « d» « addition »	**MAP CTR et TF-IDF**	0.1241	0.1529
	MAP CTR et BM25	0.1843	0.1521
	Valeur Paramètre H TF-IDF	H=1.0	H=0.3…1.0
	Valeurs de H BM25	H=0.7 ,0.8 ,0.9 ,1.0.	H=0.5et 0.7
	Taux d'amélioration % TF-IDF	+4.814%	+4.084%
	Taux d'amélioration % BM25	+55.65%	+2.424%
Formule «d» «multiplication»	**MAP CTR et TF-IDF**	0.1015	0.0258
	MAP CTR et BM25	0.0987	0.0268
	Valeur Paramètre H TF-IDF	H=0.1…1.0	H=0.2…1.0
	Valeurs de H BM25	H=0.1…1.0	H=0.2…1.0
	Taux d'amélioration % TF-IDF	-16.65%	-469.379
	Taux d'amélioration % BM25	-31.306%	-454.104
Formule « e» « addition »	**MAP CTR et TF-IDF**	0 .1439	0.1531
	MAP CTR et BM25	0.1848	0.1553
	Valeur Paramètre H TF-IDF	H=0.2 D=0.0…1.0	H=0.3
	Valeurs de H BM25	H=0.1….1.0	H=1.0
	Taux d'amélioration % TF-IDF	+21.53%	+4.220%
	Taux d'amélioration % BM25	+42.59%	+4.579%
Formule« e» «multiplication»	**MAP CTR et TF-IDF**	0 .1439	0.0545
	MAP CTR et BM25	0.1362	0.0581
	Valeur Paramètre H TF-IDF	H=0.0…1.0	H=0.2…1.0
	Valeurs de H BM25	H=0.0…1.0	H=0.2…1.0
	Taux d'amélioration % TF-IDF	+21.537%	-169.541%
	Taux d'amélioration % BM25	+5.092%	-155.593%
Formule « f» «addition»	**MAP CTR et TF-IDF**	0.1167	0.1391
	MAP CTR et BM25	0.1864	0.1472
	Valeur Paramètre H TF-IDF	H=1.0	H=0.3
	Valeurs de H BM25	H=1.0	H=0.2
	Taux d'amélioration % TF-IDF	+1.456%	-5.607%
	Taux d'amélioration % BM25	+43.382%	+0.883%
Formule « f»	**MAP CTR et TF-IDF**	0.1263	0.1127
	MAP CTR et BM25	0.1146	0.1149
	Valeur Paramètre H TF-IDF	H=0.0…1.0	H=0.1…1.0

«multiplication»	Valeurs de H BM25	H=0.0…1.0	H=0.2…1.0
	Taux d'amélioration % TF-IDF	+6.672%	-30.346%
	Taux d'amélioration % BM25	-13.089%	-29.242%
Formule «g» «addition »	**MAP CTR et TF-IDF**	0.1285	0.1648
	MAP CTR et BM25	0.1872	0.1586
	Valeur Paramètre H TF-IDF	H=0.4	H=0.8
	Valeurs de H BM25	H=1.0	H=0.9 et 1.0
	Taux d'amélioration % TF-IDF	+8.53%	+12.185%
	Taux d'amélioration % BM25	+44.444%	+6.801%
Formule « g» «multiplication»	**MAP CTR et TF-IDF**	0.1393	0.1103
	MAP CTR et BM25	0.1262	0.1138
	Valeur Paramètre H TF-IDF	H=0.0…1.0	H=0.2…1.0
	Valeurs de H BM25	H=1.0…0.0	H=0.2…1.0
	Taux d'amélioration % TF-IDF	+17.652%	-33.182%
	Taux d'amélioration % BM25	-2.623%	-30.492%
Formule « h» «addition»	**MAP CTR et TF-IDF**	0.1295	0.1648
	MAP CTR et BM25	0.1872	0.1586
	Valeur Paramètre H TF-IDF	H= 0.9 D=0.1	H=0.8 D=1.0
	Valeurs de H BM25	H=1.1 D=0.0…1.0	H=0.9 et 1.0 D=1.0
	Taux d'amélioration % TF-IDF	+9.375%	+12.185%
	Taux d'amélioration % BM25	+44.444%	+6.801%
Formule « h» «multiplication»	**MAP CTR et TF-IDF**	0.1400	0.1103
	MAP CTR et BM25	0.1265	0.1143
	Valeur Paramètre H TF-IDF	H=0.0…1.0 D=0.5	H=0.2…1.0 D=1.0
	Valeurs de H BM25	H=1.0…0.0 D=0.6 ,0.7	H=0.2…1.0 D=0.8
	Taux d'amélioration % TF-IDF	+18.243%	-33.182%
	Taux d'amélioration % BM25	-2.45%	-29.921%
Formule «i» «addition»	**MAP CTR et TF-IDF**	0.1229	0.1484
	MAP CTR et BM25	0.1639	0.1519
	Valeur Paramètre H TF-IDF	H=0.3 D=1.0	H=0.2 D=0.2
	Valeurs de H BM25	H=1.0 D=0.1	H=0.5 D=0.2
	Taux d'amélioration % TF-IDF	+3.8%	+1 .021%
	Taux d'amélioration % BM25	+26.466%	+2.289%
Formule « i» «multiplication»	**MAP CTR et TF-IDF**	0.1416	0.0989
	MAP CTR et BM25	0.1871	0.1049
	Valeur Paramètre H TF-IDF	H=0.1…1.0 D=0.1	H=0.2…1.0 D=0.1
	Valeurs de H BM25	H=0.1…1.0 D=0.9	H=0.2…1.0 D=0.2
	Taux d'amélioration % TF-IDF	+19.594%	-48.533%
	Taux d'amélioration % BM25	+44.36%	-41.563%
Formule « j» «addition»	**MAP CTR et TF-IDF**	0.1228	0.1740
	MAP CTR et BM25	0.1616	0.1584
	Valeur Paramètre H TF-IDF	H=0.3 D=0.3	H=0.9 D=0.7

	Valeurs de H BM25	H=0.1…1.0 D=0.8	H=0.9 et1.0 D=1.0et0.9
	Taux d'amélioration % TF-IDF	+3.716%	+18.447%
	Taux d'amélioration % BM25	+24.691%	+6.666%
Formule « j» «multiplication»	**MAP CTR et TF-IDF**	0.1443	0.1686
	MAP CTR et BM25	0.1216	0.1693
	Valeur Paramètre	H=0.1…1.0 D=0.8	H=0.2…1.0 D=0.4
	Valeurs de H BM25	H=0.0…1.0 D=0.8	H=0.2…1.0 D=0.4
	Taux d'amélioration % TF-IDF	+21.875%	+14.771%
	Taux d'amélioration % BM25	+43.441%	+14.006%

Tableau III.4. Résultats du modèle CTR.

Le tableau ci-dessus résume les résultats obtenu de l'implémentation du modèle CTR avec les deux modèles TF-IDF et BM25, nous constatons que la meilleure formule c'est la formule numéro « j » avec multiplication en l'intégrant au modèle TF-IDF, et la formule « h » avec l'addition en l'intégrant au modèle BM25 sur la collection AP88.

Sur la collection WT10g la meilleure formule dans le modèle TF-IDF c'est la formule « j » avec addition, et la formule « h » avec l'addition dans le modèle BM25.

III.5.2.3. Résultats des extensions du modèle CTR

Pour l'évaluation des extensions du modèle CTR nous allons évaluer les résultats des meilleures précisions obtenues précédemment. Les résultats obtenus sont résumés dans les tableaux suivants sur la collection AP88 et WT10g.

Nous avons adopté la notation suivante:
B= modèle de Base (TF-IDF ou BM25).
I (D) = modèle CTR de base .
I (M) = modèle CTR avec la première extension.
I (F) = modèle CTR avec la seconde extension.

Modèle	**MAP B**	**MAP I (D)**	**MAP I (M)**	**MAP I (F)**	**Taux d'amélioration (M) % B**	**Taux d'amélioration (F)%B**
Modèle TF_IDF	**0.1184**	**0.1228**	**0.1223**	**0.1184**	**+3.71%**	**0%**
Modèle BM25	**0.1296**	**0.1872**	**0.1845**	**0.1841**	**+44.44%**	**+42.36%**

Tableau III.5 : Taux d'amélioration des extensions du modèle CTR sur AP88

Modèle	**MAP (de base)**	**MAP I(D)**	**MAP I (M)**	**MAP I (F)**	**Taux d'amélioration (M)% B**	**Taux d'amélioration (F)%B**
Modèle TF_IDF	**0.1469**	**0.1740**	**0.1630**	**0.1538**	**+10.95%**	**+4.69%**
Modèle BM25	**0.1485**	**0.1586**	**0.1495**	**0.1498**	**+0.67%**	**+0.87%**

Tableau III.6 : Taux d'amélioration des extensions du modèle CTR sur WT10g

D'après les tableaux ci-dessus nous constatons que la première extension du modèle CTR a apporté une amélioration de 3.71 % en utilisant le modèle TF-IDF et un taux de 44 .44% en utilisant le modèle BM25 sur la collection AP88, et cela par rapport au modèles de base (TF_IDF et BM25). Cependant, par rapport au modèle CTR de base aucune amélioration n'est constatées.

Sur la collection WT10g nous constatons aussi une amélioration avec les deux extensions du modèle CTR sur les modèles de base (TF-IDF et BM25), mais aucune amélioration par rapport au modèle CTR de base.

III.5.2.4 Evaluation requête par requête

Afin d'avoir une estimation plus précise des résultats obtenus précédemment, nous avons évalué, les résultats des meilleures précisions obtenues précédemment, et cela requête par requête.

Les tableaux et les graphes suivants montrent les résultats obtenus sur la collection AP88, basé sur le modèle TF-IDF.

Num Requête	MAP B TF-IDF	MAP I (D) TF-IDF	MAP E I (M) TF-IDF	MAP I (F) TF-IDF	Taux d'amélioration %		
					I (D)	I (M)	I (F)
51	0,4903	0.4242	0.4242	0.4242	-13.481541	-13.481541	-13.481541
52	0.5206	0.4982	0.4982	0.4982	-4.3027276	-4.3027276	-4.3027276
53	0.1715	0.1679	0.1679	0.1679	-2.0991253	-2.0991253	-2.0991253
54	0.4239	0.3886	0.3886	0.3886	-8.3274357	-8.3274357	-8.3274357
56	0.2528	0.2249	0.2249	0.2249	-11.036392	-11.036392	-11.036392
57	0.6125	0.5852	0.5852	0.5852	-4.4571428	-4.4571428	-4.4571428
58	0.1306	0.0885	0.0885	0.0885	-32.235834	-32.235834	-32.235834
59	0.0181	0.017	0.017	0.017	-6.0773480	-6.0773480	-6.0773480
60	0.0083	0.0074	0.0074	0.0074	-10.843373	-10.843373	-10.843373
61	0.0011	0.0011	0.0011	0.0011	0	0	0
62	0.505	0.4775	0.4775	0.4775	-5.4455445	-5.4455445	-5.4455445
63	0.0034	0.0031	0.0031	0.0031	-8.8235294	-8.8235294	-8.8235294
64	0.0754	0.0405	0.0405	0.0405	-46.286472	-46.286472	-46.286472
65	0.0199	0.0167	0.0167	0.0167	-16.080402	-16.080402	-16.080402
66	0	0	0	0	0	0	0
67	0	0	0	0	0	0	0
68	0.001	0.001	0.001	0.001	0	0	0
69	0.0548	0.0593	0.0593	0.0593	8.21167883	8.21167883	8.21167883
70	0.0578	0.0564	0.0564	0.0564	-2.4221453	-2.4221453	-2.4221453
71	0.2102	0.1935	0.1935	0.1935	-7.9448144	-7.9448144	-7.9448144
72	0.0424	0.0368	0.0368	0.0368	-13.207547	-13.207547	-13.207547
73	0.001	0.0009	0.0009	0.0009	-10	-10	-10
74	0.0003	0.0003	0.0003	0.0003	0	0	0
75	0.0005	0.0004	0.0004	0.0004	-20	-20	-20
76	0.0124	0.0113	0.0113	0.0113	-8.8709677	-8.8709677	-8.8709677

77	0.0071	0.0069	0.0069	0.0069	-2.8169014	-2.8169014	-2.8169014
78	0.0372	0.0321	0.0321	0.0321	-13.709677	-13.709677	-13.709677
79	0.1358	0.1117	0.1117	0.1117	-17.746686	-17.746686	-17.746686
80	0	0	0	0	0	0	0
81	0.0023	0.0028	0.0028	0.0028	21.7391304	21.7391304	21.7391304
82	0.1467	0.1366	0.1366	0.1366	-6.8847989	-6.8847989	-6.8847989
83	0.1555	0.1306	0.1306	0.1306	-16.012861	-16.012861	-16.012861
84	0.0074	0.007	0.007	0.007	-5.4054054	-5.4054054	-5.4054054
85	0.0626	0.0846	0.0846	0.0846	35.14377	35.14377	35.14377
86	0.0631	0.0581	0.0581	0.0581	-7.9239302	-7.9239302	-7.9239302
87	0.0529	0.038	0.038	0.038	-28.166351	-28.166351	-28.166351
88	0.01	0.0083	0.0083	0.0083	-17	-17	-17
89	0.0916	0.0902	0.0902	0.0902	-1.5283842	-1.5283842	-1.5283842
90	0.0061	0.0062	0.0062	0.0062	1.63934426	1.63934426	1.63934426
91	0.4002	0.3828	0.3828	0.3828	-4.3478260	-4.3478260	-4.3478260
92	0.0003	0.0003	0.0003	0.0003	0	0	0
93	0.0005	0.0003	0.0003	0.0003	-40	-40	-40
94	0.0665	0.0663	0.0663	0.0663	-0.3007518	-0.3007518	-0.3007518
95	0.0087	0.0081	0.0081	0.0081	-6.8965517	-6.8965517	-6.8965517
96	0.0023	0.0021	0.0021	0.0021	-8.6956521	-8.6956521	-8.6956521
97	0.0182	0.0208	0.0208	0.0208	14.2857143	14.2857143	14.2857143
98	0.0797	0.0767	0.0767	0.0767	-3.7641154	-3.7641154	-3.7641154
99	0.2494	0.2478	0.2478	0.2478	-0.6415397	-0.6415397	-0.6415397
100	0.2419	0.2306	0.2306	0.2306	-4.6713518	-4.6713518	-4.6713518
101	0.0085	0.0066	0.0066	0.0066	-22.352941	-22.352941	-22.352941
102	0.2595	0.2422	0.2422	0.2422	-6.6666666	-6.6666666	-6.6666666
103	0.2859	0.2299	0.2299	0.2299	-19.587268	-19.587268	-19.587268
104	0.0707	0.0576	0.0576	0.0576	-18.528995	-18.528995	-18.528995
105	0.219	0.2046	0.2046	0.2046	-6.5753424	-6.5753424	-6.5753424
106	0.0001	0.0001	0.0001	0.0001	0	0	0
107	0.1844	0.1435	0.1435	0.1435	-22.180043	-22.180043	-22.180043
108	0.4874	0.4429	0.4429	0.4429	-9.1300779	-9.1300779	-9.1300779
109	0.0211	0.0186	0.0186	0.0186	-11.848341	-11.848341	-11.848341
110	0	0	0	0	0	0	0
111	0.2714	0.2579	0.2579	0.2579	-4.9742078	-4.9742078	-4.9742078
112	0.3817	0.3314	0.3314	0.3314	-13.177888	-13.177888	-13.177888
113	0.1484	0.1332	0.1332	0.1332	-10.242587	-10.242587	-10.242587
114	0.033	0.0356	0.0356	0.0356	7.87878788	7.87878788	7.87878788
115	0.0882	0.0774	0.0774	0.0774	-12.244898	-12.244898	-12.244898
116	0.1254	0.1043	0.1043	0.1043	-16.826156	-16.826156	-16.826156
117	0.0182	0.0156	0.0156	0.0156	-14.285714	-14.285714	-14.285714
118	0.105	0.0885	0.0885	0.0885	-15.714285	-15.714285	-15.714285
119	0.0255	0.0244	0.0244	0.0244	-4.3137254	-4.3137254	-4.3137254
120	0.0302	0.0272	0.0272	0.0272	-9.9337748	-9.9337748	-9.9337748
121	0.0055	0.005	0.005	0.005	-9.0909090	-9.0909090	-9.0909090
122	0.0017	0.0016	0.0016	0.0016	-5.8823529	-5.8823529	-5.8823529
123	0.0433	0.0339	0.0339	0.0339	-21.709006	-21.709006	-21.709006
124	0.039	0.0145	0.0145	0.0145	-62.820512	-62.820512	-62.820512
125	0.1014	0.0965	0.0965	0.0965	-4.8323471	-4.8323471	-4.8323471
126	0.0778	0.0598	0.0598	0.0598	-23.136246	-23.136246	-23.136246
127	0.0604	0.0594	0.0594	0.0594	-1.6556291	-1.6556291	-1.6556291
128	0.0503	0.0448	0.0448	0.0448	-10.934393	-10.934393	-10.934393
129	0.1613	0.1478	0.1478	0.1478	-8.3694978	-8.3694978	-8.3694978

130	0.0169	0.0296	0.0296	0.0296	75.147929	75.147929	75.147929
131	0.1961	0.1831	0.1831	0.1831	-6.6292707	-6.6292707	-6.6292707
132	0.0295	0.0275	0.0275	0.0275	-6.7796610	-6.7796610	-6.7796610
133	0.3226	0.2668	0.2668	0.2668	-17.296962	-17.296962	-17.296962
134	0.7095	0.7095	0.7095	0.7095	0	0	0
135	0.9167	1	1	1	9.08694229	9.08694229	9.08694229
136	0.0555	0.0438	0.0438	0.0438	-21.081081	-21.081081	-21.081081
137	0	0	0	0	0	0	0
138	0.0172	0.0146	0.0146	0.0146	-15.116279	-15.116279	-15.116279
139	0.044	0.0423	0.0423	0.0423	-3.8636363	-3.8636363	-3.8636363
140	0.0674	0.0498	0.0498	0.0498	-26.112759	-26.112759	-26.112759
141	0.0031	0.003	0.003	0.003	-3.2258064	-3.2258064	-3.2258064
142	0.0807	0.0584	0.0584	0.0584	-27.633209	-27.633209	-27.633209
143	0.0202	0.0183	0.0183	0.0183	-9.4059405	-9.4059405	-9.4059405
144	0.0099	0.0056	0.0056	0.0056	-43.434343	-43.434343	-43.434343
145	0.024	0.0218	0.0218	0.0218	-9.1666666	-9.1666666	-9.1666666
146	0.1303	0.1142	0.1142	0.1142	-12.356101	-12.356101	-12.356101
147	0.1399	0.1304	0.1304	0.1304	-6.7905646	-6.7905646	-6.7905646
148	0.0204	0.0183	0.0183	0.0183	-10.294117	-10.294117	-10.294117
149	0.015	0.0136	0.0136	0.0136	-9.3333333	-9.33333333	-9.3333333
150	0.0114	0.0098	0.0098	0.0098	-14.035087	-14.0350877	-14.035087
151	0.0112	0.0099	0.0099	0.0099	-11.607142	-11.6071429	-11.607142

Tableau III.7 : Evaluation requête par requête entre le modèle TF-IDF et les différentes extensions du modèle CTR, sur la collection AP88.

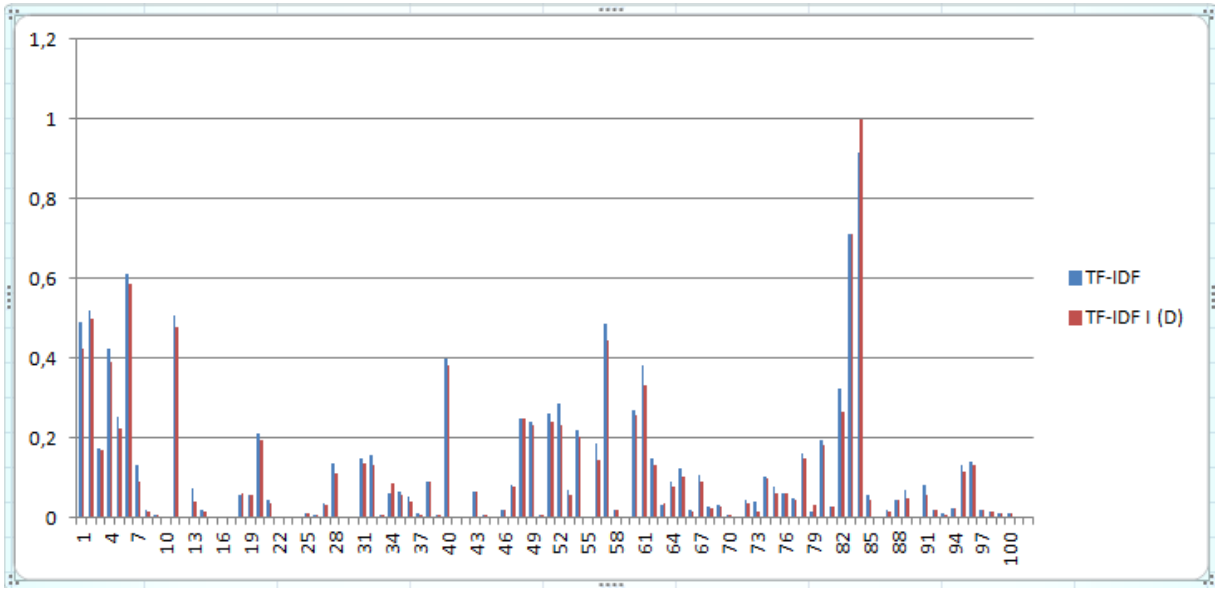

Figure III.3 : Comparaison entre les meilleures précisions obtenues avec le modèle TF –IDF et l'implémentation de CTR de base sur la collection AP88.

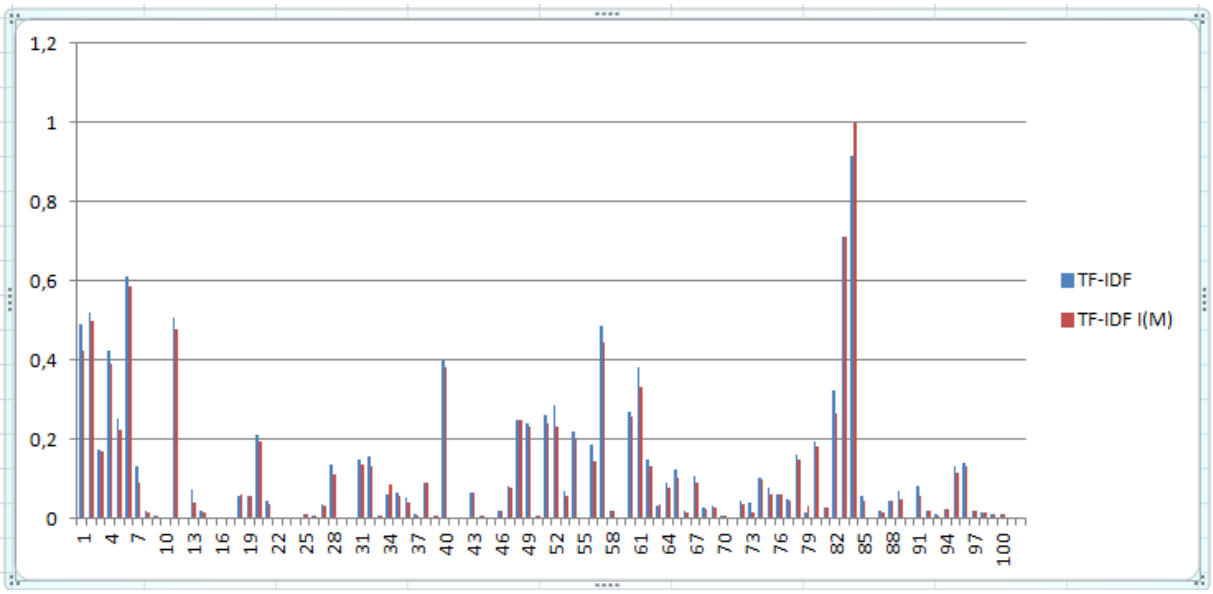

Figure III.4 : Comparaison entre les meilleures précisions obtenues avec le modèle TF-IDF de base et la première extension du modèle CTR sur la collection AP88.

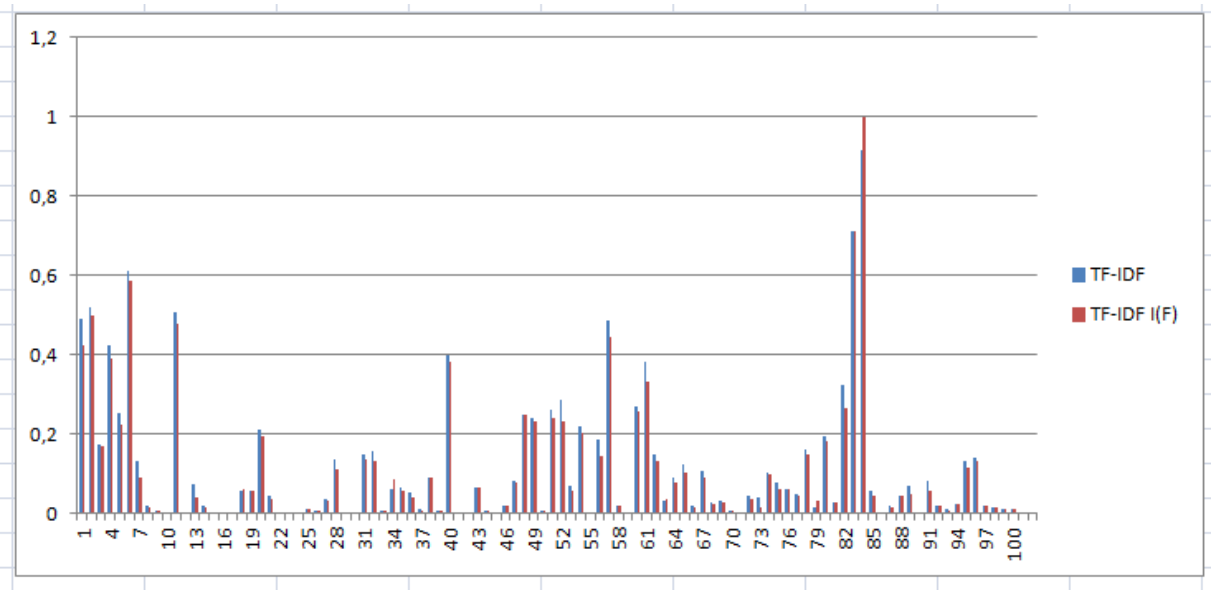

Figure III.5: Comparaison entre les meilleures précisions obtenues avec le modèle TF-IDF de base et la seconde extension du modèle CTR.

Les résultats obtenus dans le tableau ci-dessus et représentés autrement par les graphes ci-dessus, nous donnent les résultats suivants :

- 8 requêtes améliorées par apport au modèle TF-IDF
- 10 requêtes nulles par apport au modèle TF-IDF
- 82 requêtes dégradées par apport au modèle TF-IDF

Les tableaux et les graphes suivants montrent les résultats obtenus sur la collection AP88, basé sur le modèle BM25.

Num Requête	MAP B BM25	MAP I (D) BM25	MAP I (M) BM25	MAP I (F) BM25	Taux d'amélioration %		
					I (D)	I (M)	I (F)
51	0.5351	0.6953	0.6954	0.6954	29.9383293	29.9570174	29.9570174
52	0.5334	0.6497	0.6485	0.6478	21.8035246	21.5785527	21.4473191
53	0.185	0.2775	0.2774	0.2772	50	49.9459459	49.8378378
54	0.4444	0.5228	0.5234	0.5231	17.6417642	17.7767777	17.7092709
56	0.2758	0.3836	0.3816	0.3815	39.0862944	38.3611313	38.3248731
57	0.6325	0.7459	0.7457	0.7458	17.9288538	17.8972332	17.9130435
58	0.1838	0.3905	0.3905	0.3891	112.459195	112.459195	111.697497
59	0.0214	0.0863	0.0865	0.0866	303.271028	304.205607	304.672897
60	0.0087	0.0287	0.0285	0.0285	229.885057	227.586207	227.586207
61	0.0012	0.0037	0.0037	0.0037	208.333333	208.333333	208.33333
62	0.5076	0.4963	0.496	0.496	-2.2261623	-2.2852639	-2.2852639
63	0.0038	0.013	0.0131	0.0131	242.105263	244.736842	244.736842
64	0.2676	0.1826	0.1826	0.1826	-31.763826	-31.763826	-31.763826
65	0.0203	0.0308	0.0307	0.0306	51.7241379	51.2315271	50.7389163
66	0	0	0	0	0	0	0
67	0	0	0	0	0	0	0
68	0.001	0.002	0.002	0.002	100	100	100
69	0.0702	0.1618	0.1613	0.1608	130.48433	129.77208	129.059829
70	0.066	0.2191	0.2191	0.2191	231.969697	231.969697	231.969697
71	0.234	0.4247	0.4238	0.4235	81.4957265	81.1111111	80.982906
72	0.0488	0.0971	0.0971	0.097	98.9754098	98.9754098	98.7704918
73	0.001	0.0035	0.0035	0.0035	250	250	250
74	0.0003	0.0011	0.0011	0.0011	266.666667	266.666667	266.666667
75	0.0004	0.0008	0.0008	0.0008	100	100	100
76	0.0139	0.0631	0.0631	0.0629	353.956835	353.956835	352.517986
77	0.0079	0.0493	0.0493	0.0491	524.050633	524.050633	521.518987
78	0.0455	0.1992	0.1992	0.1987	337.802198	337.802198	336.703297
79	0.1661	0.5338	0.5334	0.5327	221.372667	221.131848	220.710415
80	0	0.0003	0.0003	0.0003	0	0	0
81	0.0025	0.0051	0.0051	0.0051	104	104	104
82	0.1515	0.1716	0.1714	0.1715	13.2673267	13.1353135	13.2013201
83	0.2092	0.4101	0.41	0.41	96.0325048	95.9847036	95.9847036
84	0.0077	0.0153	0.0153	0.0153	98.7012987	98.7012987	98.7012987
85	0.0641	0.1621	0.1622	0.1622	152.886115	153.042122	153.042122
86	0.0658	0.1331	0.1326	0.1318	102.279635	101.519757	100.303951
87	0.0683	0.1827	0.1697	0.1697	167.49634	148.462665	148.462665
88	0.0095	0.0142	0.0142	0.0142	49.4736842	49.4736842	49.4736842
89	0.0924	0.0728	0.0727	0.0727	-21.212121	-21.320346	-21.320346
90	0.0068	0.0235	0.0235	0.0235	245.588235	245.588235	245.588235
91	0.3968	0.3369	0.3369	0.3369	-15.095766	-15.095766	-15.095766
92	0.0003	0.0004	0.0004	0.0004	33.3333333	33.3333333	33.3333333
93	0,0004	0,0014	0,0014	0,0014	250	250	250
94	0,0693	0,2398	0,2401	0,2396	246,031746	246,464646	245,743146
95	0.0088	0.0097	0.0097	0.0096	10.2272727	10.2272727	9.09090909

96	0.0024	0.0065	0.0065	0.0065	170.833333	170.833333	170.833333
97	0.0203	0.0957	0.0957	0.0957	371.428571	371.428571	371.428571
98	0.0921	0.1413	0.141	0.141	53.4201954	53.0944625	53.0944625
99	0.3439	0.4086	0.4085	0.4084	18.8136086	18.7845304	18.7554522
100	0.2487	0.2414	0.2402	0.2394	-2.9352633	-3.4177724	-3.7394451
101	0.0094	0.0402	0.0402	0.0402	327.659574	327.659574	327.659574
102	0.2654	0.307	0.3071	0.3057	15.6744537	15.7121326	15.184627
103	0.2981	0.3237	0.3231	0.3231	8.58772224	8.3864475	8.3864475
104	0.085	0.1871	0.1857	0.1856	120.117647	118.470588	118.352941
105	0.2368	0.3002	0.3002	0.2982	26.7736486	26.7736486	25.9290541
106	0.0001	0.0001	0.0001	0.0001	0	0	0
107	0.2114	0.2841	0.2831	0.282	34.3897824	33.9167455	33.3964049
108	0.495	0.5493	0.54	0.5409	10.969697	9.09090909	9.27272727
109	0.0207	0.0272	0.0266	0.0266	31.4009662	28.5024155	28.5024155
110	0	0	0	0	0	0	0
111	0.2567	0.2831	0.2823	0.282	10.2843787	9.97273081	9.85586287
112	0.4356	0.6601	0.6596	0.6597	51.5381084	51.4233242	51.446281
113	0.1981	0.3822	0.3822	0.382	92.9328622	92.9328622	92.8319031
114	0.0343	0.0328	0.0328	0.0328	-4.3731778	-4.3731778	-4.3731778
115	0.0984	0.1313	0.131	0.131	33.4349593	33.1300813	33.1300813
116	0.1303	0.223	0.2219	0.2214	71.143515	70.2993093	69.9155794
117	0.0208	0.2	0.2	0.2	861.538462	861.538462	861.538462
118	0.1193	0.1636	0.162	0.1636	37.1332775	35.7921207	37.1332775
119	0.0259	0.0543	0.0543	0.0543	109.65251	109.65251	109.65251
120	0.0301	0.0628	0.0629	0.0628	108.637874	108.9701	108.637874
121	0.0063	0.0183	0.0178	0.0178	190.47619	182.539683	182.539683
122	0.0017	0.0066	0.0066	0.0066	288.235294	288.235294	288.235294
123	0.0509	0.0938	0.0932	0.0932	84.2829077	83.1041257	83.1041257
124	0.0402	0.0818	0.0818	0.0818	103.482587	103.482587	103.482587
125	0.1054	0.1352	0.135	0.1349	28.2732448	28.0834915	27.9886148
126	0.0924	0.1576	0.1573	0.1567	70.5627706	70.2380952	69.5887446
127	0.0668	0.1061	0.106	0.106	58.8323353	58.6826347	58.6826347
128	0.0505	0.1016	0.1013	0.1008	101.188119	100.594059	99.6039604
129	0.2011	0.4034	0.4034	0.4035	100.596718	100.596718	100.646445
130	0.0201	0.0441	0.0441	0.044	119.402985	119.402985	118.905473
131	0.2175	0.3231	0.3229	0.3229	48.5517241	48.4597701	48.4597701
132	0.0305	0.0275	0.0274	0.0274	-9.8360655	-10.163934	-10.163934
133	0.3715	0.5619	0.5611	0.5611	51.2516824	51.0363392	51.0363392
134	0.7095	0.7095	0.7095	0.7095	0	0	0
135	0.9167	1	1	1	9.08694229	9.08694229	9.08694229
136	0.0682	0.1361	0.1353	0.1352	99.5601173	98.3870968	98.2404692
137	0	0	0	0	0	0	0
138	0.0191	0.0977	0.0977	0.0977	411.518325	411.518325	411.518325
139	0.0459	0.0484	0.0483	0.0482	5.44662309	5.22875817	5.01089325
140	0.069	0.0708	0.0706	0.0709	2.60869565	2.31884058	2.75362319
141	0.0029	0.0047	0.0047	0.0047	62.0689655	62.0689655	62.0689655
142	0.0975	0.1132	0.1111	0.1133	16.1025641	13.9487179	16.2051282
143	0.0215	0.0413	0.0413	0.0408	92.0930233	92.0930233	89.7674419
144	0.0101	0.0065	0.0065	0.0065	-35.643564	-35.643564	-35.643564
145	0.0182	0.0149	0.014	0.0147	-18.131868	-23.076923	-19.230769
146	0.1375	0.1486	0.1486	0.1491	8.07272727	8.07272727	8.43636364
147	0.1487	0.245	0.2446	0.2445	64.7612643	64.4922663	64.4250168
148	0.0313	0.0479	0.0478	0.0478	53.0351438	52.715655	52.715655

149	0.0197	0.0465	0.0465	0.0465	136.040609	136.040609	136.040609
150	0.0141	0.1409	0.1409	0.1409	899.29078	899.29078	899.29078
151	0.0078	0.0126	0.0123	0.0123	61.5384615	57.6923077	57.6923077

Tableau III.8 : Evaluation requête par requête entre le modèle BM25 et les différentes extensions du modèle CTR, sur la collection AP88.

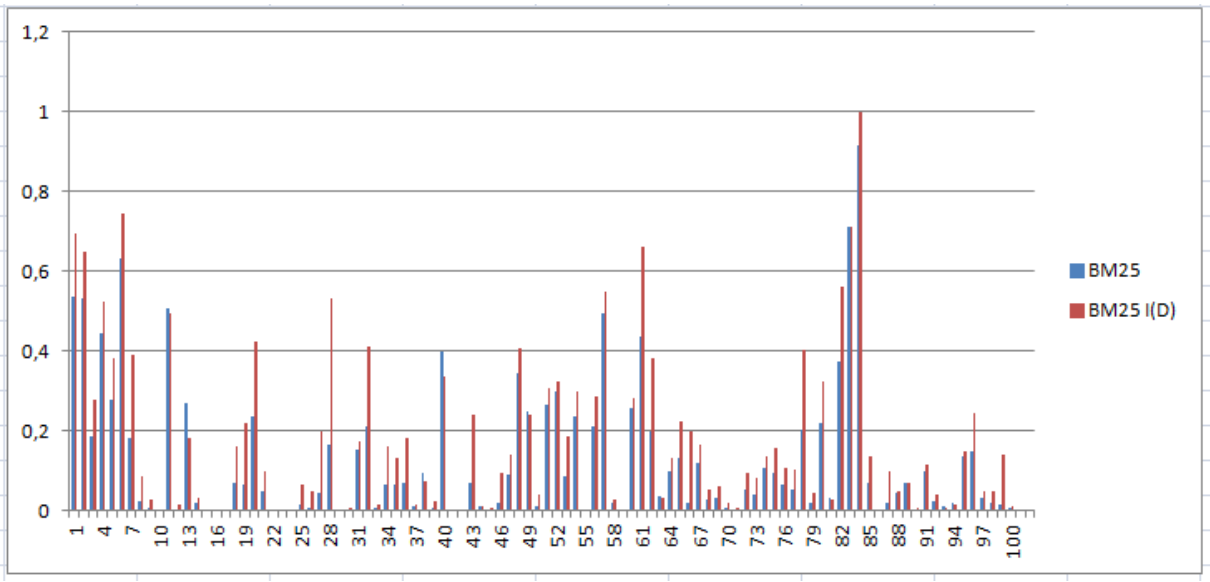

Figure III.6 : Comparaison entre les meilleures précisions obtenues avec le modèle BM25 et l'implémentation du modèle CTR de base sur la collection AP88.

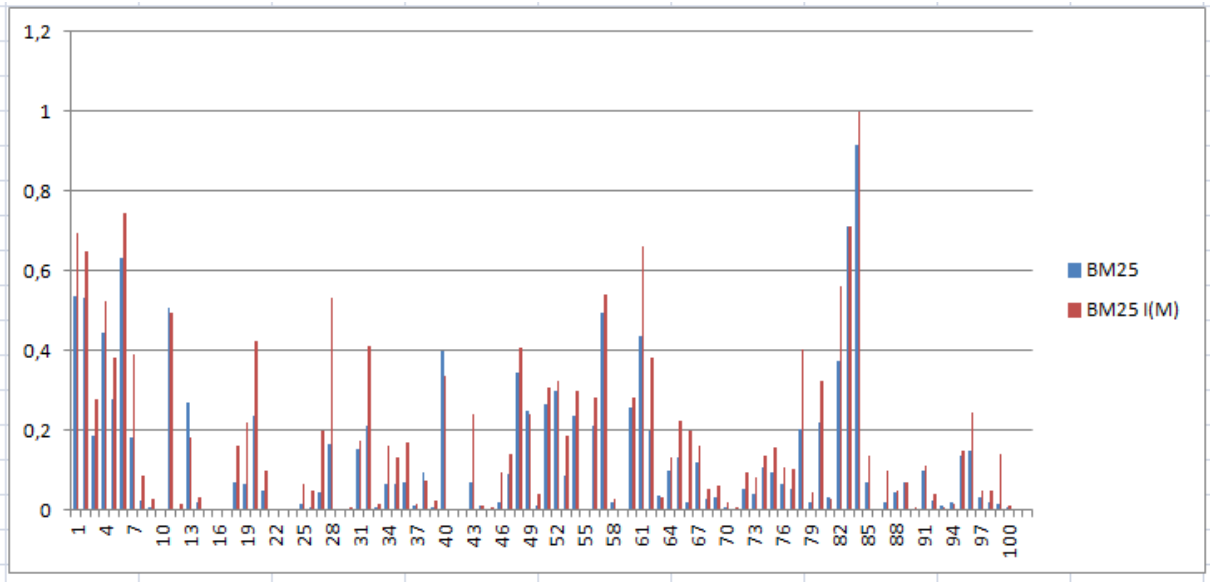

Figure III.7 : Comparaison entre les meilleures précisions obtenues avec le modèle BM25 de et la première extension de CTR sur la collection AP88.

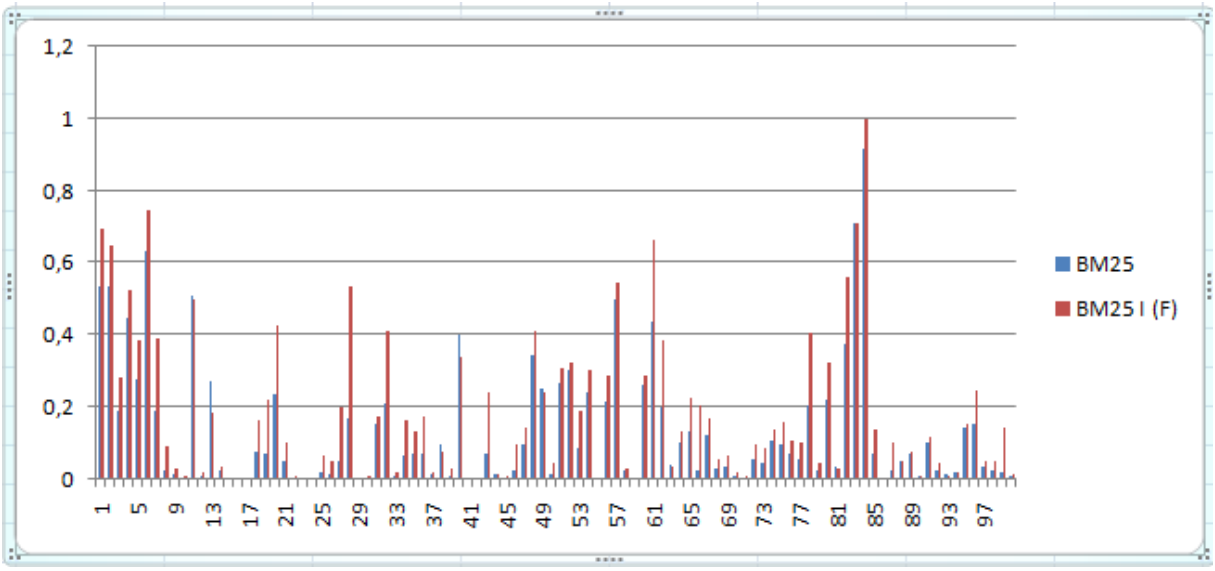

Figure III.8 : Comparaison entre les meilleures précisions obtenues avec le modèle BM25 et la seconde extension de CTR sur la collection AP88.

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

- 86 requêtes améliorées par rapport au modèle BM25.
- 7 requêtes nulles par rapport au modèle BM25 .
- 7 requêtes dégradée par rapport au modèle BM25.

Les tableaux et les graphes suivants montrent les résultats obtenus sur la collection WT10g, basé sur le modèle TF-IDF.

Num Requête	MAP B TF-IDF	MAP TF-IDF I(D)	MAP TF-IDF I(M)	MAP TF-IDF I(F)	Taux d'amélioration %		
					I(D)	I(M)	I(F)
451	0.4624	0.4821	0.4656	0.4658	4 .26	0.69	0.73
452	0.0249	0.0345	0.0272	0.0273	38.55	9.23	9.63
453	0.3465	0.3538	0.3466	0.3465	2.10	0.02	0
454	0.3106	0.3461	0.3191	0.3188	11.49	2.73	2.64
455	0.2304	0.2985	0.2486	0.2492	29.55	7.89	8.15
456	0.0012	0.0030	0.0014	0.0014	150	16.66	16.66
457	0.1183	0.1433	0.1204	0.1200	21.13	1.77	1.43
458	0.0692	0.0716	0.0695	0.0692	3.46	0.43	0
459	0.2040	0.2004	0.2092	0.2092	-1.76	2.54	2.54
460	0.1158	0.1174	0.1148	0.1146	1.38	-0.86	-1.03
461	0.6042	0.6110	0.6059	0.6060	1.12	0.28	0.29
462	0.2010	0.2464	0.2030	0.2036	22.58	0.99	1.29
463	0.0000	0.0000	0.0000	0.0000	0	0	0
464	0.0059	0.0093	0.0060	0.0059	57.62	1.69	0
465	0.2823	0.1200	0.2778	0.2763	-57.49	-1.59	-2.12
466	0.1475	0.1384	0.1441	0.1441	-6.16	-2.30	-2.30
467	0.0275	0.0290	0.0284	0.0284	5.45	3.27	3.27

468	0.0070	0.0507	0.0085	0.0092	624.28	21.42	31.42
469	0.0843	0.0749	0.0842	0.0843	-11.15	-0.11	0
470	0.0206	0.0198	0.0218	0.0200	-3.88	5.82	-2.91
471	0.0466	0.0302	0.0473	0.0473	-35.19	1.50	1.50
472	0.0000	0.0000	0.0000	0.0000	0	0	0
473	0.0002	0.0003	0.0002	0.0002	50	0	0
474	0.1150	0.1217	0.1238	0.1262	5.82	7.65	9.73
475	0.1975	0.2540	0.2183	0.2203	28.60	10.53	11.54
476	0.0073	0.0070	0.0068	0.0068	-4.10	-6.84	-6.84
477	0.0085	0.0105	0.0085	0.0085	23.52	0	0
478	0.2029	0.2590	0.2236	0.2249	27.64	2.07	10.84
479	0.0005	0.0010	0.0005	0.0005	100	0	0
480	0.2325	0.3349	0.2665	0.2675	44.04	14.62	15.05
481	0.0865	0.0746	0.0881	0.0885	-13.75	1.84	2.31
482	0.0312	0.0306	0.0311	0.0311	-1.92	-0.32	-0.32
483	0.0969	0.0803	0.0906	0.0907	-17.13	-6.50	-6.39
484	0.4167	0.3250	0.4167	0.4167	-22.00	0	0
485	0.8611	0.8500	0.8611	0.8611	-1.28	0	0
486	0.0568	0.0636	0.0575	0.0582	11.97	1.23	2.46
487	0.0457	0.0756	0.0446	0.0457	65.42	-2.40	0
488	0.0921	0.1137	0.0999	0.1009	23.45	8.46	9.55
489	0.0009	0.0016	0.0012	0.0012	77.77	33.33	33.33
490	0.3563	0.3341	0.3462	0.3463	-5.47	-2.82	0
491	0.0001	0.0001	0.0001	0.0001	0	0	0
492	0.1514	0.1641	0.1516	0.1514	8.38	0.13	0
493	0.0172	0.0285	0.0204	0.0206	65.69	18.60	19.76
494	0.2331	0.2728	0.2331	0.2331	18.62	0	0
495	0.4404	0.4128	0.4404	0.4404	-6.26	0	0
496	0.0833	0.0833	0.0833	0.0833	0	0	0
497	0.3000	0.5000	0.3000	0.3000	66.66	0	0
498	0.0535	0.0451	0.0519	0.0516	-15.70	-2.99	-3.55
499	0.0228	0.0359	0.0280	0.0283	57.45	22.80	24.12
500	0.0709	0.0406	0.0631	0.0624	-42.73	-11.00	-11.98
501	0.0672	0.0869	0.0704	0.0712	29.31	4.76	5.95
502	0.3335	0.3358	0.3379	0.3384	0.68	1.31	1.46
503	0.2347	0.3062	0.2640	0.2663	30.46	12.48	13.46
504	0.1075	0.0802	0.1075	0.1075	-25.36	0	0
505	0.0592	0.0818	0.0618	0.0618	38.17	4.39	4.39
506	0.0233	0.0285	0.0257	0.0258	22.31	10.30	10.72
507	0.2109	0.2728	0.2313	0.2337	29.35	9.67	10.81
508	0.4899	0.5530	0.5206	0.5215	12.88	6.26	6.45
509	0.1539	0.1762	0.1653	0.1665	14.48	7.40	8.18
510	0.1621	0.1555	0.1540	0.1525	-4.07	-4.99	-5.92
511	0.1126	0.1174	0.1133	0.1126	4.26	0.62	0
512	0.3125	0.3132	0.3124	0.3100	0.22	-0.03	-0.8
513	0.1015	0.1411	0.1140	0.1147	39.01	12.31	13.00
514	0.0530	0.0711	0.0530	0.0530	34.15	0	0
515	0.0967	0.0856	0.0933	0.0925	-11.47	-3.51	-4.34
516	0.0280	0.0259	0.0284	0.0281	-7.5	1.42	0.35
517	0.0917	0.0899	0.0911	0.0905	-1.96	-0.65	-1.30
518	0.0641	0.0709	0.0652	0.0650	10.60	1.71	1.40
519	0.0300	0.0096	0.0179	0.0160	-68	-40.33	-46.66
520	0.0070	0.0071	0.0072	0.0072	1.42	2.85	2.85

521	0.0406	0.0191	0.0331	0.0328	-52.95	-18.47	-19.21
522	0.0033	0.0033	0.0033	0.0033	0	0	0
523	0.0846	0.0771	0.0770	0.0763	-8.86	-8.98	-9.81
524	0.0748	0.0966	0.0748	0.0748	29.14	0	0
525	0.1733	0.2260	0.1892	0.1900	30.40	9.17	9.63
526	0.3676	0.3752	0.3697	0.3698	2.06	0.57	0.59
527	0.2496	0.2706	0.2630	0.2633	8.41	5.36	5.48
528	0.1925	0.2312	0.2101	0.2107	20.10	9.14	9.45
529	0.1078	0.0874	0.1073	0.1078	-18.92	-0.46	0
530	0.1840	0.1969	0.1845	0.1840	7.01	0.27	0
531	0.2069	0.2195	0.2111	0.2125	6.08	2.02	2.70
532	0.0045	0.0056	0.0049	0.0049	24.44	8.88	8.88
533	0.0035	0.0060	0.0037	0.0037	71.42	5.71	5.71
534	0.1790	0.2178	0.1970	0.1974	21.67	10.05	10.83
535	0.0067	0.0162	0.0070	0.0070	141.79	4.47	4.47
536	0.3750	0.5909	0.3750	0.3750	57.57	0	0
537	0.1026	0.0797	0.0986	0.0978	-22.31	-3.89	-4 .67
538	0.0418	0.0311	0.0357	0.0315	25.59	-14.59	-24.64
539	0.1848	0.1924	0.1828	0.1819	4.11	-1.08	-1.56
540	0.0004	0.0002	0.0004	0.0004	-50	0	0
541	0.0001	0.0004	0.0003	0.0003	300	200	200
542	0.4159	0.4443	0.4312	0.4326	6.82	3.67	4.01
543	0.2072	0.1210	0.1727	0.1720	-41.60	16.65	16.98
544	0.0891	0.1162	0.0966	0.0969	30.41	8.41	8.75
545	0.1037	0.0997	0.1072	0.1078	-3.85	3.37	3.95
546	0.4167	0.5000	0.3333	0.3333	19.99	-20.01	-20.01
547	0.1801	0.1820	0.1846	0.1840	1.05	2.49	2.16
548	0.1725	0.1847	0.1781	0.1783	7.07	3.24	3.36
549	0.4624	0.1857	0.4656	0.4666	-59.83	0.69	0.90
550	0.1725	0.1847	0.0272	0.0282	7.07	-84.23	-83.65

Tableau III.9: Evaluation requête par requête entre le modèle TF-IDF et les différentes extensions du modèle CTR, sur la collection WT10g.

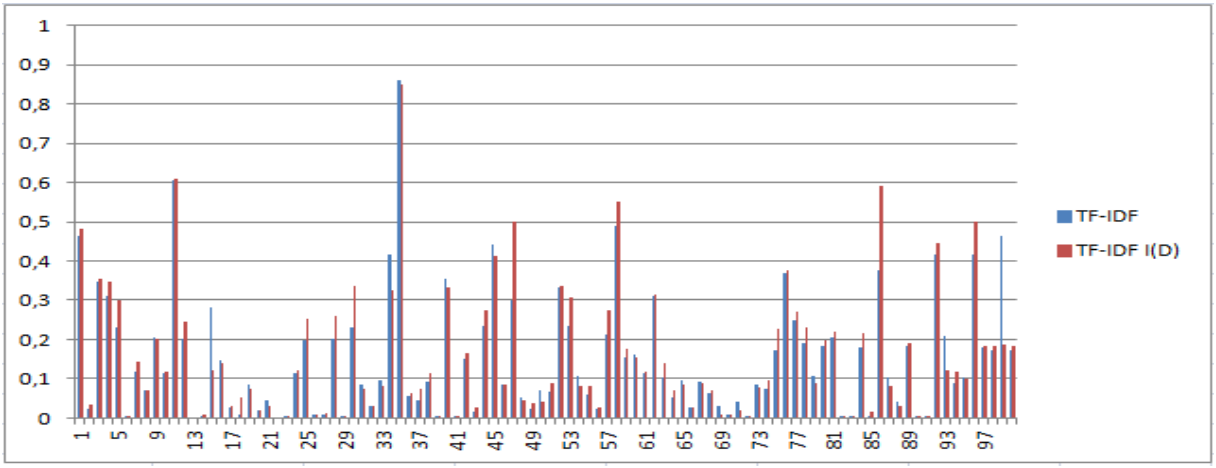

Figure III. 9 : Comparaison entre les meilleures précisions obtenues avec le modèle TF_IDF et l'implémentation du modèle CTR de base sur la collection WT10g.

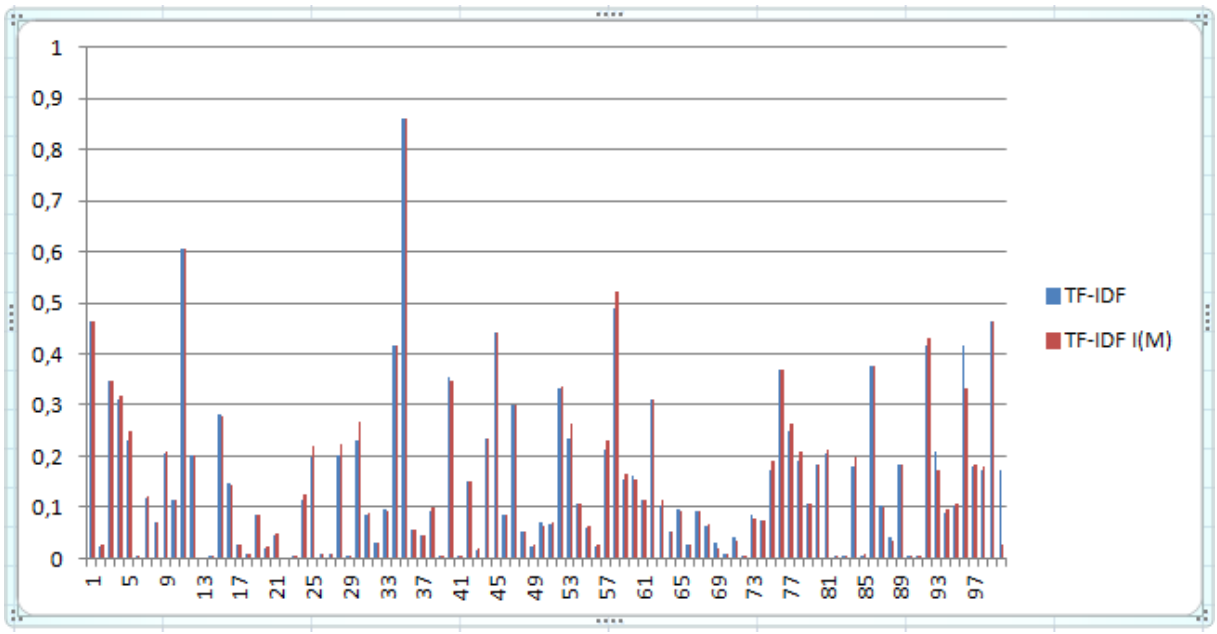

Figure III.10 : Comparaison entre les meilleures précisions obtenues avec le modèle CTR de base et la première extension du modèle CTR sur la collection WT10g .

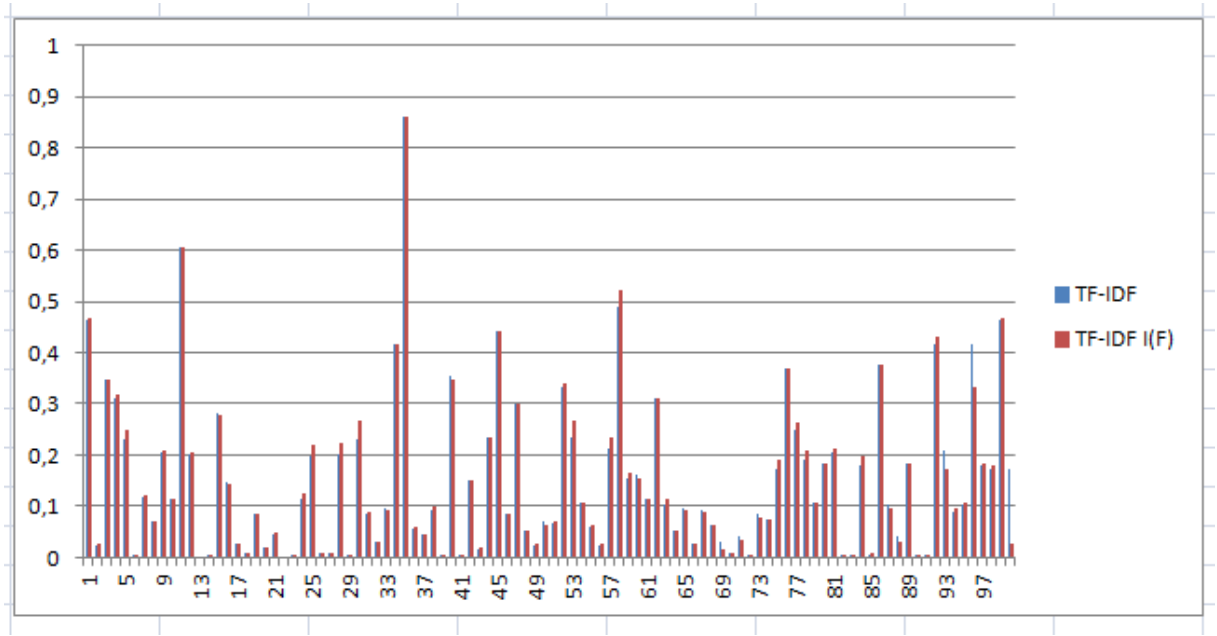

Figure III.11 : Comparaison entre les meilleures précisions obtenues avec le modèle CTR de base et avec la seconde extension du modèle CTR sur la collection WT10g.

Les résultats obtenus dans le tableau et les graphes ci-dessus nous donnent les résultats suivants :

- 63 requêtes améliorées par apport au modèle TF-IDF.
- 5 requêtes nulles par apport au modèle TF-IDF.
- 32 requêtes dégradées par apport au modèle TF-IDF.

Requête	MAP B (BM25)	MAP I D BM25	MAP I M BM25	MAP I F BM25	Taux d'amélioration %		
					I (D)	I (M)	I (F)
451	0.4420	0.4853	0.4863	0.4851	9.79	10.02	9.75
452	0.0257	0.0371	0.0372	0.0374	44.35	44.74	45.52
453	0.3465	0.3453	0.3476	0.3448	-0.34	0.31	-0.49
454	0.3105	0.3293	0.3231	0.3222	6.05	4.05	3.76
455	0.2342	0.3026	0.2964	0.2963	29.20	26.55	26.51
456	0.0011	0.0013	0.0012	0.0012	18.18	9.09	9.09
457	0.1181	0.1328	0.1321	0.1298	12.44	11.85	9.90
458	0.0692	0.0699	0.0695	0.0692	1.01	0.43	0
459	0.2008	0.1940	0.1930	0.1926	-3.38	-3.88	-4.08
460	0.1208	0.1119	0.1101	0.1096	-7.36	-8.85	-9.27
461	0.6042	0.6129	0.6132	0.6131	1.43	1.48	1.47
462	0.2006	0.2052	0.2028	0.2028	2.29	1.09	1.09
463	0.0000	0.0000	0.0000	0.0000	0	0	0
464	0.0059	0.0060	0.0060	0.0058	1.69	1.69	-1.69
465	0.2845	0.1263	0.1263	0.1263	-55.60	-55.60	-55 .60
466	0.1468	0.1337	0.1328	0.1352	-8.92	-9.60	-7.90
467	0.0275	0.0322	0.0334	0.0333	17.09	21.47	21.09
468	0.0070	0.0508	0.0508	0.0508	625.71	625.71	625.71
469	0.0843	0.0811	0.0842	0.0842	-3.79	-0.11	-0.11
470	0.0223	0.0107	0.0107	0.0104	-52.01	-52.01	-53.36
471	0.0466	0.0308	0.0316	0.0315	-98.22	-32.18	-32.40
472	0.0000	0.0000	0.0000	0.0000	0	0	0
473	0.0002	0.0002	0.0002	0.0002	0	0	0
474	0.1165	0.1183	0.1183	0.1182	1.54	1.54	1.45
475	0.1976	0.2553	0.2556	0.2554	29.20	29.35	29.25
476	0.0073	0.0064	0.0064	0.0064	-12.32	-12.32	-12.32
477	0.0085	0.0087	0.0084	0.0081	2.35	-1 .17	-4.70
478	0.1939	0.2479	0.2415	0.2402	27.84	24.45	23.87
479	0.0005	0.0009	0.0009	0.0009	80	80	80
480	0.2408	0.3367	0.3370	0.3374	39.82	39.95	40.11
481	0.0667	0.0602	0.0600	0.0599	-9.74	-10.04	-10.19
482	0.0312	0.0303	0.0296	0.0300	-2 .88	-4.98	-3.84
483	0.0986	0.0831	0.0819	0.0820	-15.72	-16.93	-16.83
484	0.4167	0.4167	0.4167	0.4167	0	0	0
485	0.8611	0.8500	0.8500	0.8500	-1.28	-1.28	-1.28
486	0.0571	0.0572	0.0563	0.0553	0.17	-1.40	-3.15
487	0.0457	0.0499	0.0447	0.0442	9.19	-2.18	-3.28
488	0.0926	0.1140	0.1140	0.1143	23.11	23.11	23.43
489	0.0009	0.0016	0.0016	0.0016	77.77	77.77	77.77
490	0.3705	0.3190	0.3155	0.3134	-13.90	-14.84	-15.41
491	0.0001	0.0001	0.0001	0.0001	0	0	0
492	0.1514	0.1534	0.1512	0.1489	1.32	-0.13	-1.65
493	0.0214	0.0309	0.0308	0.0313	44.39	43.92	46.26
494	0.2331	0.2415	0.2331	0.2331	25.99	0	0
495	0.4404	0.4405	0.4403	0.4625	0.02	-0.02	5.01
496	0.0833	0.0833	0.0833	0.0833	0	0	0
497	0.3000	0.3000	0.3000	0.3000	0	0	0
498	0.0538	0.0445	0.0445	0.0443	-17.28	-17.28	-17.65
499	0.0228	0.0358	0.0357	0.0358	57.01	56.57	57.01

500	0.0711	0.0464	0.0476	0.0472	-34.73	-49.36	-33.61
501	0.0672	0.0846	0.0841	0.0840	25.89	25.14	25
502	0.3402	0.3423	0.3414	0.3369	0.65	-8.46	-0.97
503	0.2347	0.3140	0.3140	0.3147	33.78	33.78	34.08
504	0.1075	0.0631	0.0631	0.0632	-41.30	-41.30	-41.20
505	0.0592	0.0765	0.0776	0.0788	29.22	31.08	33.10
506	0.0245	0.0283	0.0282	0.0283	15.51	15.10	15.51
507	0.2443	0.2737	0.2729	0.2724	12.03	11.70	11.50
508	0.4875	0.5435	0.5413	0.5395	11.48	11.03	10.66
509	0.1542	0.1669	0.1662	0.1659	8.23	7.78	7.58
510	0.1525	0.1503	0.1519	0.1507	-1.44	-0.39	-1.18
511	0.1126	0.1139	0.1118	0.1117	1.15	-0.71	-0.79
512	0.3161	0.2987	0.2977	0.2985	-5.50	-5 .82	-5.56
513	0.1049	0.1256	0.1249	0.1250	19.73	19.06	19.16
514	0.0530	0.0548	0.0522	0.0511	3.39	-1.50	-3 .58
515	0.1150	0.0809	0.0801	0.0800	-29.65	-29.65	-30.43
516	0.0334	0.0270	0.0271	0.0267	-19.16	-18.86	-20.05
517	0.0988	0.0914	0.0912	0.0913	-7.48	-7.69	-7.59
518	0.0656	0.0540	0.0540	0.0538	-17.68	-17.68	-17.98
519	0.0155	0.0085	0.0085	0.0086	-45.16	-45.16	-44.51
520	0.0070	0.0070	0.0070	0.0070	0	0	0
521	0.0521	0.0188	0.0182	0.0181	-63.91	-65.06	-65.25
522	0.0033	0.0033	0.0033	0.0032	0	0	-3.03
523	0.1057	0.0720	0.0703	0.0695	-31.88	-33.49	-34.24
524	0.0748	0.0751	0.0747	0.0745	0.40	-0.13	-0.40
525	0.1745	0.2147	0.2141	0.2139	23.03	22.69	22.57
526	0.3745	0.3768	0.3767	0.3747	0.61	0.58	0.05
527	0.2497	0.2387	0.2320	0.2290	-4.40	-7.08	-8.28
528	0.2010	0.2389	0.2394	0.2421	18.85	19.10	20.44
529	0.1058	0.1054	0.1078	0.1019	-0.37	1.89	-3.68
530	0.1840	0.1850	0.1822	0.1798	0.54	-0.97	-2.28
531	0.2088	0.2201	0.2187	0.2176	5.41	4.74	4.21
532	0.0041	0.0050	0.0050	0.0049	21.95	21.95	19.51
533	0.0034	0.0039	0.0038	0.0039	14.70	11.76	14.70
534	0.1792	0.2203	0.2197	0.2198	22.93	22.60	22.65
535	0.0063	0.0075	0.0072	0.0072	19.04	14.28	14.28
536	0.3750	0.3500	0.3500	0.3611	-6.66	-6.66	-3.70
537	0.1076	0.0819	0.0804	0.0795	-23.88	-25.27	-26.11
538	0.0419	0.0331	0.0330	0.0331	-21.00	-21.24	-21.00
539	0.1853	0.1695	0.1643	0.1661	-8.52	-11.33	-10.36
540	0.0007	0.0002	0.0002	0.0002	-71.42	-71.42	-71.42
541	0.0003	0.0004	0.0004	0.0004	33.33	33.33	33.33
542	0.4217	0.4488	0.4482	0.4476	6.42	6.28	6.14
543	0.2172	0.1195	0.1197	0.1200	-44.98	-44.88	-44.75
544	0.0882	0.1174	0.1176	0.1176	33.10	33.33	33.33
545	0.1058	0.0995	0.0987	0.0983	-5.95	-6.71	-7.08
546	0.4500	0.5000	0.4500	0.4500	11.11	0	0
547	0.1864	0.1666	0.1647	0.1616	-10.62	-11.64	-13 .30
548	0.1693	0.1788	0.1789	0.1807	5.61	5.67	6.73
549	0.1864	0.1666	0.1647	0.1616	-10.62	-11.64	-13 .30
550	0.1693	0.1788	0.1789	0.1807	5.61	5.67	6.73

Tableau III.10 : Evaluation requête par requête entre le modèle BM25 et les différentes extensions du modèle CTR, sur la collection WT10g.

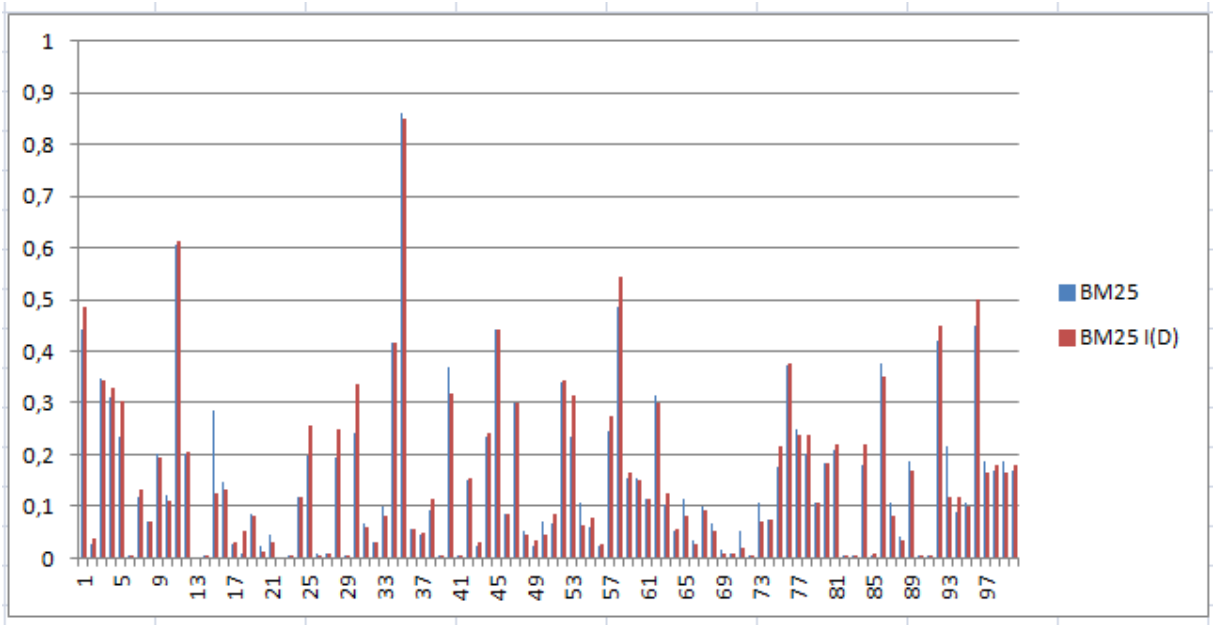

Figure III.12 : Comparaison entre les meilleures précisions obtenues avec le modèle BM25 et l'implémentation du modèle CTR de base sur la collection WT10g.

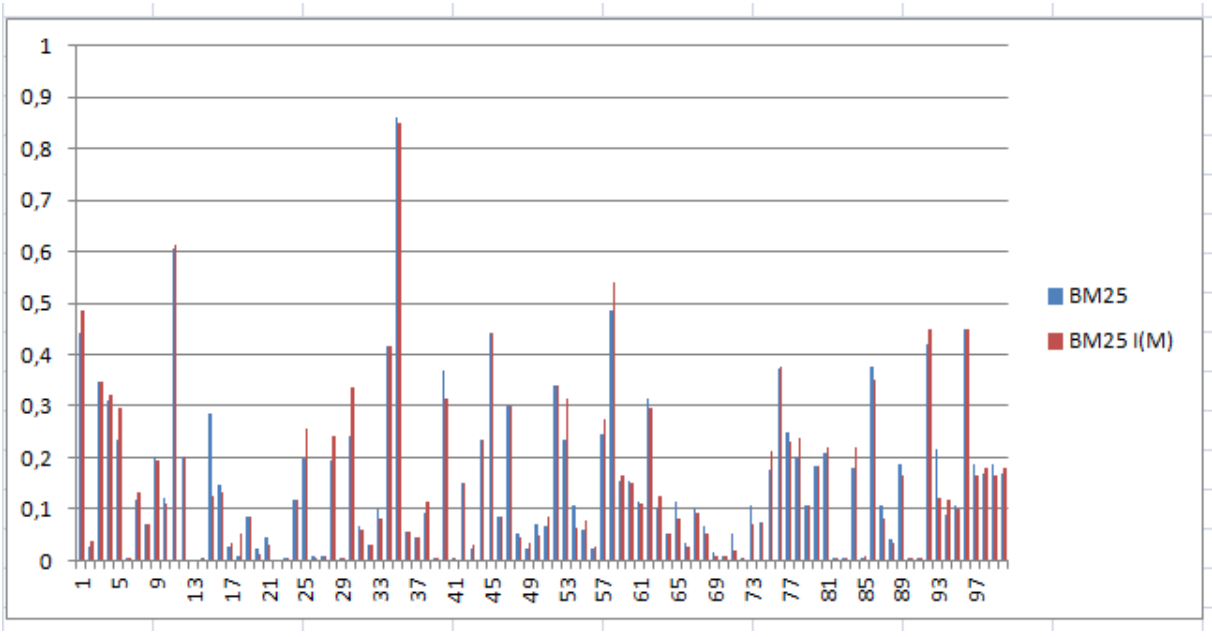

Figure III.13: Comparaison entre les meilleures précisions obtenues avec le modèle CTR de base et la première extension du modèle CTR sur la collection WT10g.

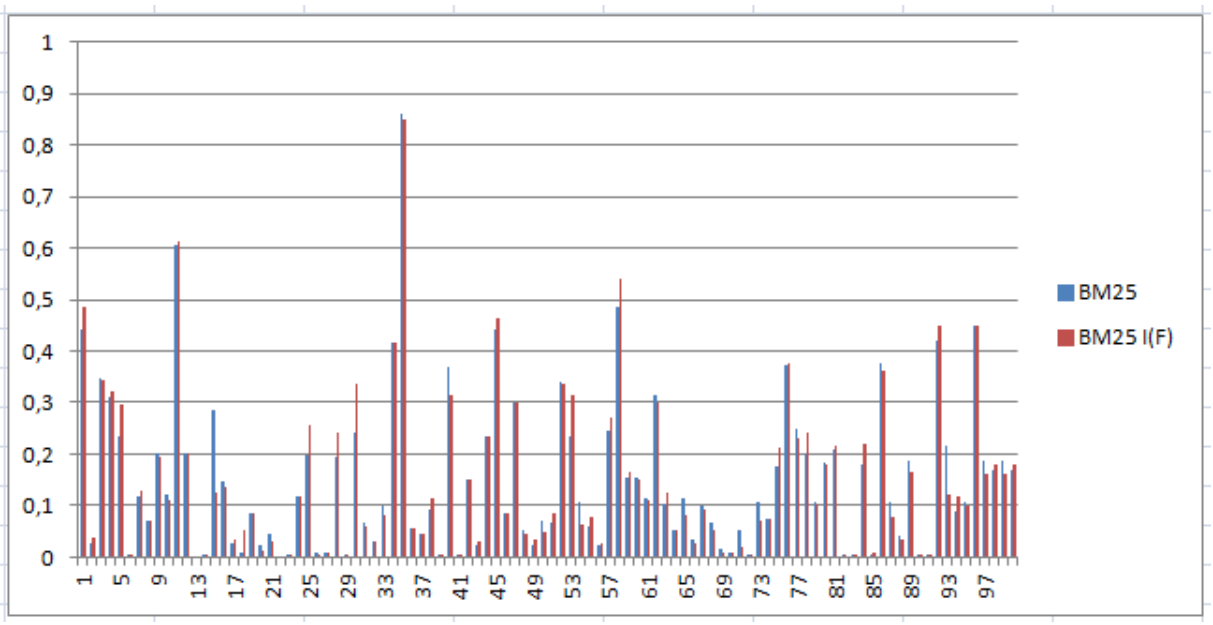

Figure III.13 : Comparaison entre les meilleures précisions obtenues avec le modèle CTR de base et la seconde extension du modèle CTR sur la collection WT10g.

Les résultats obtenus dans ce tableau et les graphes ci-dessus nous donnent les résultats suivant :

- 52 requêtes améliorées par apport au modèle TF-IDF.
- 9 requêtes nulles par apport au modèle TF-IDF.
- 39 requêtes dégradées par apport au modèle TF-IDF.

III.6.Conclusion

Dans ce chapitre nous avons exposé le modèle CTR qui consiste à étendre le modèle TF-IDF et BM25 avec le facteur : « position du terme » dans le document, puis nous avons proposé deux extensions pour ce modèle. En fin nous avons implémenté sous Terrier ces modèles et évalués sur deux collections de test TREC (AP88, WT10G); nous avons aussi présenté les résultats des évaluations sous forme de tableaux et de graphiques.

Les résultats obtenu montrent que la prise en compte de la position des termes de la requête dans les documents apporte un plus pour la RI.

Conclusion Générale

Le travail présenté dans ce mémoire se situe dans le cadre de la recherche d'information, particulièrement au niveau de la pondération des termes.

Notre objectif consistait, en premier à implémenter un modèle "CTR" qui intègre un facteur de pondération qui est la position d'un terme de requête dans un document. Ensuite, à proposer des extensions pour ce modèle et bien sur l'évaluation des ces dernières.
Les expérimentations réalisées sont des expérimentations de grandeur nature. C'est à dire nous avons utilisé les collections utilisés dans le domaine, à savoir les collections TREC. Précisément, nous avons utilisé deux collections: AP88 et WT10G.

La réalisation de ce travail nous a permis tout d'abord de nous imprégner des principaux concepts de cette nouvelle discipline 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 en particulier le SRI terrier.
Enfin sur le plan pratique, nous avons amplifié nos connaissances sur le langage JAVA lors de la réalisation de l'implémentation du modèle CTR.

A l'issu du travail élaboré dans ce mémoire, et d'après les résultats obtenus, nous sommes dans la bonne voix sur les deux collections de tests AP88 et WT10g.
Cependant, des améliorations peuvent être apportées a ce travail, dont nous pouvons citer:

- La combinaisons des trois versions du modèle CTR en pondérant chaque version.
- L'utilisation d'une étape d'apprentissage pour identifier quelle version du modèle à utiliser.

Notre perspective est de tester le modèle CTR et les extensions proposées sur d'autres collections pour une meilleure interprétation des résultats.

[1] Abbadeni .N, Ziou .D, Wang .S « Recherche d'images basée sur leur contenu», Rapport de recherche, université de Sherbrooke, 1998.

[2] Herzallah A., RECHERCHE D'INFORMATION, Cours, Université de Bouira.

[3]Baeza–Yates, R, Ribeiro-Netro, B, A.Modern Information Retrieval. Pearson Education Ltd., Harlow, UK, 2nd edn, 2011.

[4]Hammache A., thèse de doctorat en informatique. « Recherche d'Information : un modèle de langue combinant mots simples et mots composés », Université de Tizi-Ouzou, 2013.

[5]Belkin .N.Jand Croft.W.B.Information filtering and information retrieval: Two Sides of the same coin?Commun.ACM,35(12):29-38,1992.

[6]BORLUND.P, AND INGWERSEN.P, Measures of relative relevance and ranked half-life: performance indicators for interactive ir.In Proc.of the International ACM-SIGIR conference (1998), pp.24-28.

[7]Cleverdon ,C.Progress in Documentation Evaluation of information retrieval systems .Journal of Documentation,26,pp.55-67,1970.

[8]Harter,S.Psychological revelance and information science .Journal of the American Society for Information Science (JASIS),43:602-615,1992.

[9]Calabretto S, Recherche d'Information, LIRIS, INSA Lyon. 2003.

[10]Salton .G and Mc Gill. Introduction to Modern Information Retrieval. McGraw-Hill, New York, 1983.

[11]Le Maitre Jacques « Recherche d'information ».Université du Sud Toulon-Var ; France.

[12]Luhn, 58 : Luhn, H. «The automatic creation of literature abstracts». IBM Journal of Research and Development, 2(2),1958.

[13]Tmar.M, Modèle auto-adaptatif de filtrage d'information : apprentissage incrémental du profil et de la fonction de décision. Thèse de l'Université Paul Sabatier. Toulouse 2002.

[14]Zemirli W.Nesrine « Vers le développement d'un système de recherche d'information personnalisé intégrant le profil utilisateur », Université Paul Sabatier – Toulouse III, Année 2003/2004.

[15]Salton et al., 83 : Salton, G., Fox, E. A. et Wu, H.« Extended boolean information retrieval». Commun. ACM, 1983.

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

[17]Paice , 84 : Paice, C. D. «Soft evaluation of Boolean search queries in information retrieval systems». Inf. Technol. Res. Dev. Appl., 1984

[18]Salton, 71: Salton, G. « A Comparison between manual and automatic indexing methods».Journal of the American Documentation, 20(1), pp. 6171, 1971.

[19]Robertson, 94 : Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M. et Gatford, M. « Okapi at TREC-3. In Proceedings of the Third Text REtrieval Conference», TREC'94.1994.

[20]Lancaster ,F.Information Retrieval Systems:characteristics ,testing,and evaluation ,John Wiley,New York, 1979.

[21]Sanderson,M.Test collection based evaluation of information retrieval systems.Foundation and Trends in Information Retrieval 4,pp.247-375,2010.

[22]Cruzel, 01 ,S.L. Cruzel. "Conception de systèmes de recherche d'informations : accès aux documents numériques scientifiques". HDR, Université Claude Bernard Lyon 1, 2001

[23]Rijsbergen .V. C. J. Information Retrieval. Department of Computing Science University of Glasgow.

[24]G.K Zipf, 1949. « Recherche d'Information. PDF »

[25]Harman .D, Relevance feedback revisited. Proceedings of ACM SIGIR1992

[26]BAZIZ Mustapha, thèse En vue de l'obtention du doctorat de l'université PAUL SABATIER ; Spécialité informatique, «« indexation conceptuelle guidée par ontologie pour la recherche d'information»». Décembre 2005.

[27]Jones, 72 : Jones, K. S. « A statistical interpretation of term specificity and its application in retrieval». Journal of Documentation .1972.

[28]Luhn, 57 : Luhn, H. « A statistical approach to mechanized encoding and searching of literary information». IBM Journal of Research and Development 4, 1 .1957.

[29]Salton & Buckley, 1988: Salton, G., and Buckley, C. «Term-weighting approaches in automatic text retrieval ». Information Processing & Management (IPM) .1988.

[30]G. Salton,1988: Syntactic approaches to automatic book indexing. In *Proc. of the annual meeting on Association for Computational Linguistics (ACL) (1988)*, pages 204–210, Department of Computer Science,Cornell University, Ithaca, New York, 1988.

[31]G. Salton: *A Theory of Indexing*. Society for Industrial and Applied Mathematics, Philadelphia, 1975.

[32]Singhal,1997: Mandar Mitra, Chris Buckley, Amit Singhal, Claire Cardie. *Conference Proceedings of RIAO-97*, pages 200-214, 1997.

[33]M. Maron,1961: Automatic indexing : an experimental enquiry. *Journal of the ACM*, 24(8) :404–417, 1961.

[34]SPARCK Jones, K .1972: "Une interprétation statistique de la spécificité terme et son application dans la récupération". Journal de la documentation, 28 (1). 1972.

[35]V. N. Anh and A. Moffat. Simplified similarity scoring using term ranks. In SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pages 226–233, 2005.

[36] S. E. Robertson, S. Walker, and M. Beaulieu. Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactive track. In NIST Special Publication 500-242: Proceedings of the Seventh Text Retrieval Conference (TREC-7), pages 253–264, July 1999.

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

[38] Yuanhua Lv & ChengXiang Zhai, Positional Language Models for Information Retrieval". In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR'09), pages 299-306, 2009.

[39] Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M. et Gatford, M. « Okapi at TREC-3. In Proceedings of the Third Text REtrieval Conference», TREC'94. 1994.

[40] BRINI ASMA HEDIA « Un Modèle de Recherche d'Information basé sur les Réseaux Possibilistes », Université Paul Sabatier de Toulouse, Décembre 2005.

[41] Porter M., An algorithm for suffix stripping, in Program Volume 14 no. 3, pp 130-137, July 1980.

[42] RASOLOFO Y., SAVOY J., « Term Proximity Scoring for Keyword- based Retrieval Systems », ECIR 03, 2003.].

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

[44]Manning, D, Schutze, H,Fondation of Statistic Natural Processing MIT Press ,2000.

[45]Ponte, TM, Coft,W.B.A langage modeling approach to information retrieval.Proceding of the 21st annual International ACM SIGIK conference research and development in information retrieval .pp 275-281.1998.

[46]Middleton C., Baeza-yates R."A Comparaison of Open Source Search Engines",Technical report,octobre,2007.

[47]Ounis I.,Amati G., plachouras V., He B., Macdonalde C.,Lioma C.,"terrier :A high performance and Scalable Information Retrieval platform",Proceeding of ACM SIGIR' O6 Workshop on Open Source information Retrieval (OSIR 2006),2006.

[48]http:/www.netbeans.org.

[49] Troy A.D, Zhang G.Q. "Enhancing Relevance Scoring with Chronological Term Rank". In SIGIR '07 : Actes de la conférence annuelle internationale ACM 30 SIGIR sur la recherche et le développement dans la recherche d'information page 599-606 .