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

Mémoire de fin d'étude de master Professionnel

Domaine: Mathématique et informatique

Filière: Informatique

Spécialité : Ingénierie des Systèmes d'Information

Thème

Implémentation et évaluation d'une approche de RI basée sur la position du terme

Présenté par :

M^{elle} : AHMIL Siham

M^{elle} : HANSAL Amina

Devant les jurys composés de :

Président : Mr SADOU Samir

Encadreur : Mr HAMMACHE Arezki

Examinateur : Mr CHEBOUBA Lokmane

Promotion: 2019/2020

Remerciement

A l'issue du cycle de notre formation nous tenons
à remercier « DIEU » le tout puissant
pour nous avoir donné la force, capacité,
volonté et courage afin de mener
à bien et à terme ce travail
Nos remerciements les plus sincères vont à:
Notre promoteur Mr HAMMACHE Arezki
Pour ses conseils précieux son dévouement,
pour son suivi et pour nous avoir aussi
bien encadrées tout au long de
la réalisation de ce projet.
Nous tenons aussi à lui adresser notre
gratitude pour tout le temps qu'il nous a
consacrée, sa disponibilité ainsi
que ses encouragements.
Nos vifs remerciements vont aux membres
De jury pour avoir accepté de juger
Notre présent travail
En fin toute personne qui a participé de prés
Ou de loin à l'accomplissement de ce mémoire
Soit sincèrement remerciée.

Dédicaces

Je dédie ce travail à :

En premier lieu ceux que personne ne peut compenser les sacrifices qu'ils ont consentis pour mon éducation et mon bien-être à mes parents qui se sont sacrifiés pour me prendre en charge tout au long de mes études et qui sont l'origine de ma réussite que DIEU les garde et les protèges.

A la mémoire de mon oncle, qui nous a quitté tôt.

A ma chère sœur Celina, qui sait toujours comment procurer la joie et le bonheur pour toute la famille.

A mes chers frères Mourad et Yacine.

A ma tante Nadia et son petit ange Rahimou.

A mes grands-parents, que DIEU leur donne la santé et longue vie.

A toutes les personnes de ma grande famille.

A ma très chère binôme Siham, pour les merveilleux moments passés ensemble, et à sa famille.

A tous mes collègues de la promotion 2019/2020.

AMINA

Dédicaces

Je dédie ce modeste travail :

A ceux qui ont fait de moi ce qui je suis et ne cessent pas de me soutenir et de me faire confiance : mes très chers parents : AHMIL Djaffar et CHAABA Ouiza pour l'amour et le soutien que m'avez offert tout le long de mon cursus, je vous dis merci……

Un jet d'encre ne suffira jamais à vous remercier

A mes chers frères Rafik et Azouaou.

A ma chère sœur Alicia.

A ma petite ange Tafat.

A toutes les personnes de ma grande famille.

A ma très chère binôme Amina et sa famille.

A toute personne qui me connais.

A tous mes collègues de la promotion 2019/2020.

SIHAM

Sommaire

Sommaire

Chapitre II : Les facteurs de pondération

Chapitre III : Description et évaluation de notre approche

Sommaire

Liste des figures

Liste des tableaux

Introduction générale

Introduction générale

Avec l'augmentation rapide du volume documentaire stocké sous format numérique et l'avènement du web, la quantité d'information disponible ne cesse de croitre au cours de ces dernières années, il est devenu alors très difficile de trouver une information ou un document qui répond à un besoin utilisateur. Il a fallu donc envisager le développement des outils automatiques qui permettent l'accès ciblé et efficace à cette masse d'information. Ces difficultés ont donné naissance à une nouvelle discipline appelée « La Recherche d'Information » (RI).

L'objectif principal de la RI est de fournir des modèles, des techniques et des outils pour stocker et organiser des masses d'informations et localiser celles qui seraient pertinentes relativement à un besoin en information d'un utilisateur, souvent, exprimé à travers une requête. Ces outils sont appelés des Systèmes de Recherche d'Information (SRI). De manière générale, le fonctionnement d'un SRI consiste à construire une représentation des documents et de la requête et d'établir une comparaison entre ces deux représentations (requête, documents) pour retourner les documents pertinents. Cette comparaison est réalisée au moyen d'un modèle de recherche. Afin d'obtenir un SRI performant, il est nécessaire de construire une bonne représentation du document et de la requête et de développer un modèle de RI qui supporte ces représentations.

Le modèle de recherche d'information (MRI) joue un rôle central dans la RI. C'est lui qui détermine le comportement clé d'un SRI. Le fonctionnement d'un MRI est basé sur la fonction de pondération. Cette dernière combine généralement deux facteurs. Ces facteurs sont : la fréquence du terme dans le document TF *(Term Frequency)*, et la fréquence inverse en documents l'IDF (*Inverse document Frequency*). Ces facteurs sont principalement basés sur la représentation en sac de mots des documents. Cette représentation facilite grandement les calculs. Cependant, elle ignore l'ordre d'apparition des termes dans le document. L'une des pistes poursuivies pour aller au-delà de cette représentation est l'utilisation des positions des termes dans le document.

Notre travail s'insère dans cadre de cette piste. Il consiste en l'extension des modèles de RI suivants : TF_IDF et BM25 pour la prise en compte de facteur de position des termes dans un document. Précisément, en surpondérant le poids des termes qui apparaissent au début des documents. Cette idée est basée sur l'intuition suivante : « les auteurs des documents placent les termes les plus importants dans leurs premières parties».

Introduction générale

Pour atteindre cet objectif, le présent mémoire, comporte outre l'introduction générale, la conclusion et la bibliographie, les trois chapitres suivants :

Dans le premier chapitre intitulé « Recherche d'Information », l'objectif est de présenter le domaine de la recherche d'information. Dans un premier temps, nous présentons les concepts de base de la recherche d'information puis nous passerons aux modèles de la recherche d'information et nous finirons par traiter de l'évaluation des systèmes de recherche d'information (SRI).

Le deuxième chapitre intitulé « Facteurs de pondération », est consacré à la présentation de différents facteurs de pondération. En premier lieu, nous allons présenter les facteurs de pondération classiques. En second lieu, nous allons présenter les facteurs de pondérations supplémentaires utilisés pour réévaluer le poids d'un terme dans un document.

Le troisième chapitre intitulé **«** Description et évaluation de notre approche **»,** est dédié à la description de notre approche, son implémentation ainsi que les différents outils utilisés pour la réaliser et nous terminerons avec la présentation et la discussion des résultats obtenus.

Chapitre I
La Recherche d'Information

Chapitre I : La recherche d'information

I.1 Introduction

La recherche d'information (RI) est un domaine qui remonte au début des années 1950, peu après l'invention des ordinateurs. La RI est la science qui s'intéresse à la représentation, le stockage, l'organisation et l'accès à l'information **[1]**.

La recherche d'information est un domaine historiquement lié aux sciences de l'information et à la bibliothéconomie qui ont toujours eu le souci d'établir des représentations des documents dans le but d'en récupérer des informations à travers la construction d'index. L'informatique a permis le développement d'outils pour traiter l'information et établir la représentation des documents au moment de leur indexation, ainsi que pour rechercher l'information. Donc la RI peut être étudié par plusieurs disciplines utilisant des approches qui devraient permettre de trouver des solutions pour améliorer son efficacité.

Ce chapitre a pour objectif de présenter les principaux concepts de la RI les principaux modèles de recherche ainsi que les approches d'évaluation des SRI

I.2 Les concepts de la RI

La tâche principale d'un système de recherche d'information (SRI) est de trouver à partir d'une collection de documents ceux qui sont susceptibles de répondre à une requête utilisateur. Son but est de retourner à l'utilisateur une liste de documents ordonnés contient le maximum de documents pertinents pouvant satisfaire son besoin et le minimum de documents non pertinents.

Cette définition fait ressortir trois notions clés : document, requête, pertinence

Un document : peut-être un texte, une page web, une image, une bande vidéo, etc… On appelle document toute unité qui peut constituer une réponse à une requête utilisateur.

Une requête : exprime le besoin d'information d'un utilisateur est un ensemble de mots clés, mais elle peut être exprimée en langage naturel, booléen ou graphique.

La pertinence : est une notion centrale en RI. Elle définit le degré de correspondance entre un document et une requête.

Cette correspondance peut être considérée du point de vue de l'utilisateur (on parle alors de pertinence utilisateur), ou du point de vue système (on parle de pertinence système)

- La pertinence système : c'est l'évaluation par le SRI, de l'adéquation entre des documents et une requête.
- La pertinence utilisateur : c'est l'évaluation par l'utilisateur, de la pertinence, vis-à-vis de son besoin en information, des documents retrouvés par le SRI **[1]**.

I.3 Processus de la recherche d'information

Afin d'être en mesure d'offrir aux utilisateurs les informations correspondants à leurs attentes, Le processus de recherche d'information met en œuvre un certain nombre de processus qui permettent de mettre en relation l'ensemble des informations disponibles dans le fond permettent de mettre en relation l'ensemble des informations disponibles dans le fond documentaire d'une part et les besoins en information des utilisateurs d'une autre par

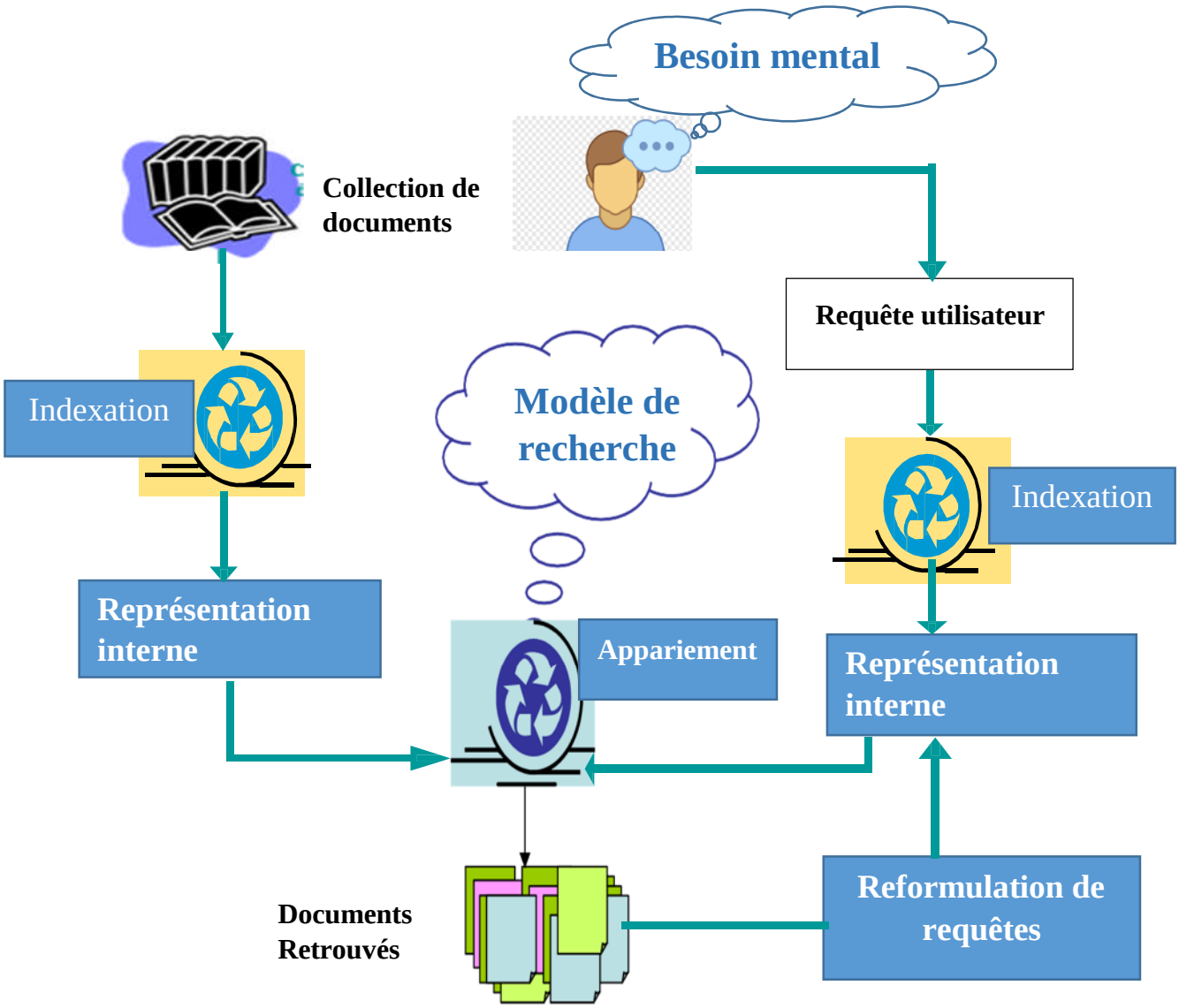


Figure I.1: Processus de recherche d'information [2]

Le fonctionnement général d'un SRI fait ressortir trois mécanismes de base :

I.3.1 Indexation des documents et des requêtes

L'indexation est une technique de passer d'un document textuel a une représentation exploitable par un modèle de RI, Elle a pour but d'extraire à partir d'un document ou d'une requête une représentation paramétrée qui couvre au mieux son contenu sémantique. Le résultat de l'indexation est une liste de termes significatifs pour l'unité textuelle correspondante, auxquels sont associés généralement des poids pour différencier leur degré d'importance.

I.3.2 Appariement requête-document et ordonnancement des résultats

Les résultats d'un SRI doit s'appuyer sur un modèle de pertinence des documents. Celui-ci va ainsi permettre de réaliser, pour chaque requête, un calcul de score de pertinence pour chacun des documents, ce score est généralement calculé à partir d'une fonction, qui utilise plusieurs sources de pertinence, notée RSV (Q, D) (Retrieval Slatus Value), où Q est une requête et D un document.

Les documents sont ensuite présentés aux utilisateurs par ordre décroissant et ceux qui auront le meilleur score de pertinence seront alors au début de la liste.

I.3.3 Reformulation de requêtes

Consiste, à partir d'une requête initiale formulée par l'utilisateur, des résultats initiaux fournis en réponse à cette requête et des jugements de pertinence utilisateur sur ces résultats, à construire une nouvelle requête qui répond mieux à son besoin informationnel.

I.4 Processus d'indexation

La stratégie de recherche séquentielle ne peut pas être réalisable pour accéder à une grande quantité d'informations dans des délais acceptables. Pour résoudre ce problème d'accès afin d'optimiser le cout de la recherche, une étape primordiale doit s'effectuer sur les documents avant l'étape de recherche effective de l'information. Cette étape est appelée indexation.

I.4.1 Types d'indexation

Selon Salton et McGill l'indexation est un processus qui « transforme les documents en substituts capables de représenter leurs contenus » **[3]**.

L'objectif de l'indexation est de créer une représentation interne (l'index) des documents en vue de faciliter la recherche.

L'indexation peut être :

Manuelle : chaque document est analysé par un spécialiste du domaine ou par un documentaliste afin d'extraire les termes significatifs des documents. C'est une approche subjective, puisque le choix des termes d'indexation dépend de l'indexeur et de ses connaissances du domaine, et pratiquement reste inapplicable aux corpus de textes volumineux.

Semi-automatique : Le processus automatique extraire les termes des documents.

Cependant, un spécialiste ou un documentaliste pour le choix final des termes significatifs.

Automatique : C'est un processus complètement automatisé qui se charge d'extraire les termes caractéristiques du document. Elle est réalisée par un programme informatique et elle passe par un ensemble d'étapes pour créer d'une façon automatique l'index.

Comme l'indexation automatique est la technique la plus répandue dans le domaine de recherche d'information, à cause de la taille des collections de documents utilisées (indexation manuelle très coûteuse), nous allons décrire les étapes essentielles de cette technique qui regroupe plusieurs traitements automatiques sur un document, de l'extraction des mots jusqu'à la création de l'index.

I.4.2 Etape de l'indexation automatique

I.4.2.1 Analyse lexicale

C'est le processus qui transforme une suite de caractères en une suite de mots, dit aussi tokens. Cette analyse permet de reconnaitre les espaces de séparation, les chiffres et les ponctuations.

I.4.2.2 Elimination des mots vides

L'élimination des mots vides est l'une des étapes du processus d'indexation permettant d'améliorer la fiabilité d'un SRI au sens de qualité logiciel et de performance.

Elle consiste a évité les mots vides (pronom personnel, prépositions, articles, Les déterminants, Les adverbes) et choisir seulement les termes significatifs qui représentent au mieux un document donné. Afin d'éliminer ces mots de force, on utilise une liste appelée, stop-liste (ou parfois anti-dictionnaire) qui contient tous les mots qu'on ne veut pas garder **[4]**.

I.4.2.3 La normalisation

L'idée qui conduit à utiliser la normalisation est de pouvoir indexer un ensemble de mots par un seul mot qui représente le même concept, (généralement le masculin pour les noms, l'infinitif pour les verbes, le masculin-singulier pour les adjectifs **[5]**. Plusieurs types stratégiques de lemmatisation ont été proposés dans la littérature : la table de consultation (dictionnaire), l'élimination des affixes (algorithme de Porter), la troncature, les variétés de successeurs ou encore la méthode des n-grammes **[6]**.

I.4.2.4 Création de l'index

Le résultat final du processus d'indexation est de créer l'index, c'est une structure de stockage pour mémoriser les informations sélectionnées lors de ce processus, qui permet de répondre plus rapidement à une requête et de sélectionner pour n'importe quel terme les documents où il appartient. Plusieurs techniques de stockage ont été développées parmi lesquelles : les fichiers inverses, les tableaux de suffixes et les fichiers de signatures. Les fichiers inverses sont actuellement les plus utilisés pour la plupart des applications. La structure d'un fichier inverse associé à un document est composée de deux éléments:

- Le vocabulaire : est l'ensemble des termes d'index de la collection, auquel on peut éventuellement rajouter l'information sur le nombre de documents de la collection où le terme apparaît, et sa fréquence d'occurrence totale.
- Le postings : est l'ensemble de toutes les listes de documents contenant chaque terme du vocabulaire. Ces listes peuvent éventuellement inclure la fréquence d'occurrence du terme dans le document qui le contient.

I.5 Les modèles de recherche d'information

Un modèle de RI a pour rôle de fournir une formalisation du processus de la recherche d'information. Il doit accomplir plusieurs rôles dont le plus important est de fournir un cadre théorique pour la modélisation de la mesure de pertinence **[7]**. Il existe un grand nombre de modèles de RI textuelle développé dans la littérature, ils s'appuient sur des cadres théoriques différents, théorie des ensembles, algèbre, probabilités, etc. On distingue trois principaux modèles :

Les modèles ensemblistes : reposent sur la théorie des ensembles, les termes de la requête sont séparés par des opérateurs logiques.

Les modèles algébriques : Se basent sur la théorie algébrique. La pertinence est définie par des mesures de similarité dans un espace vectoriel.

Les modèles probabilistes : se basent sur la théorie des probabilités. La pertinence est vue comme une probabilité de pertinence document/requête.

Dans cette section, nous décrivons pour chacun de ces courants le modèle le plus représentatif (à savoir : le modèle booléen, le modèle vectoriel et le modèle probabiliste).

I.5.1 Modèle booléen ou ensembliste

Le modèle booléen est l'un des premiers modèles utilisés en recherche d'information **[8]**. Dans ce modèle, chaque document d est représenté par une conjonction logique des termes non pondérés qui constitue l'index du document. La requête q de l'utilisateur est représentée par une expression logique, composée de termes reliés par des opérateurs logiques : ET (∧), OU (∨), et SAUF (¬).

La fonction de correspondance est basée sur l'hypothèse de présence/absence des termes de la requête dans le document et Vérifie si l'index de chaque document dj implique l'expression logique de la requête q.

Le résultat de cette fonction est donc binaire est décrit Comme suit :

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

Avantages et inconvénients

Le principal avantage du modèle est sa transparence. Le système sélectionne les documents qui répondent exactement à la requête formulée par l'utilisateur, un document est soit pertinent soit non pertinent. L'inconvénient de ce modèle est qu'il rend la tâche de formulation de la requête par l'utilisateur plus complexe, à cause de sa manière d'appariement. En outre, il est incapable de fournir une liste ordonnée de documents car la perception de la pertinence selon le modèle booléen est très différente de celle de l'utilisateur. Afin de tenir compte de la pondération des termes dans les documents et requêtes, des extensions du modèle booléen standard sont proposés, parmi elles on trouve : le modèle booléen basé sur la théorie des ensembles flous, le modèle booléen étendu.

I.5.2 Modèle vectoriel

Le modèle vectoriel est un modèle statistique qui consiste à représenter les documents et les requêtes sous forme de vecteurs de termes pondérés.

Gérard Salton **[9]** et son équipe dans le projet SMART ont proposé de représenter les documents et les requêtes sous forme de vecteurs dans l'espace vectoriel engendré par les termes extraits de tous les documents de la collection. La pertinence d'un document par rapport à la requête est évaluée par le degré de similarité entre le vecteur du document D et celui de la requête Q

La représentation formelle du document et de requête est la suivante :

- Le document $\overrightarrow{dj} = (w1j, w2j, \ldots, wtj)$ où wij représente le poids des termes dans le document D
- La requête $\vec{q} = (w1q, w2q, \ldots, wtq)$

Une des plus simples mesures de similarité est celle du produit scalaire :

$$RSV(\overrightarrow{dj}, \vec{q}) = \sum_{i=1}^{t} wij * wiq \qquad (I.2)$$

Où t est le nombre total de termes de l'index.

Plusieurs fonctions de similarité ont été proposées dans la littérature. Parmi lesquelles on peut citer les mesures de Cosinus, Jaccard, Dice et Overlap.

Indice de Cosinus :

$$RSV(\overrightarrow{dj}, \vec{q}) = \frac{\sum_{i=1}^{t} wij * wiq}{\sqrt{\sum_{i=1}^{t} wiq^2} * \sqrt{\sum_{i=1}^{t} wij^2}} \qquad (I.3)$$

Indice de Jaccard :

$$RSV(\overrightarrow{dj}, \vec{q}) = \frac{\sum_{i=1}^{t} wij * wiq}{\sum_{i=1}^{t} wiq + \sum_{i=1}^{t} wij - \sum_{i=1}^{t} wij * wiq} \qquad (I.4)$$

Indice de Dice :

$$RSV(\overrightarrow{dj}, \vec{q}) = \frac{2\sum_{i=1}^{t} wij * wiq}{\sum_{i=1}^{t} wiq + \sum_{i=1}^{t} wij} \qquad (I.5)$$

Overlap:

$$\text{RSV}(\overrightarrow{dJ},\vec{q}) = \frac{\sum_{i=1}^{t} wij * wiq}{Min(\sum_{i=1}^{t} wiq \ , \ \sum_{i=1}^{t} wij)} \quad \text{(I.6)}$$

Le modèle vectoriel est l'un des modèles les plus influents, les plus étudiés et les mieux acceptés en domaine de RI. Il a trouvé son succès dans sa simplicité conceptuelle et de mise en œuvre. En fait, les mesures de similarité employées par ce modèle permettent à l'utilisateur d'avoir une liste triée des documents pertinents, en plaçant en tête les documents jugés les plus similaires à la requête. Cependant, ce modèle a plusieurs inconvénients. A titre d'exemple, il ne permet pas de modéliser les dépendances entre les termes d'indexation. Chacun des termes est considéré comme indépendant des autres. On reproche aussi à ce modèle l'absence de base théorique forte dans la représentation des documents et des requêtes, et de la fonction de correspondance

I.5.3 Modèles probabilistes

I.5.3.1 Le modèle probabiliste de base

La modélisation probabiliste dans le domaine de recherche d'information, consiste à utiliser un modèle qui classe les documents dans un ordre décroissant de leurs probabilités de pertinence à un besoin d'information d'un utilisateur. L'objectif est de répondre, pour chaque document et chaque requête, à la question : Quelle est la probabilité que le document soit pertinent pour la requête ? **[10]** Le premier modèle probabiliste de RI a vu le jour vers les débuts des années 60. Il a été proposé par Maron et Kuhn **[11]** Depuis, plusieurs modèles probabilistes ont été proposés dont la plupart se basent sur le principe de classement probabiliste (Probability Ranking Principle, PRP) défini par Robertson **[12]**. L'idée de ce principe est de classer l'ensemble des documents en deux classes notamment, la classe des documents pertinents (notée R) et la classe des documents non pertinents (notée NR) à un besoin utilisateur.

Étant donnée une requête Q, la tâche du modèle probabiliste consiste alors à estimer la probabilité qu'un document D appartient à la classe des documents pertinents (non pertinents). Un document est alors sélectionné si sa probabilité de pertinence à Q, notée P(R|D), est supérieure à la probabilité qu'il soit non pertinent à Q, notée P(NR|D). Le score d'appariement entre le document D et la requête Q, noté RSV (D, Q), est donné par :

$$\text{RSV (D, Q)} = \frac{P(R|D)}{P(NR|D)} \quad \text{(I.7)}$$

En appliquant le théorème de Bayes et après simplification, la formule devient :

$$\text{RSV (d, Q)} = \frac{P(R|D)}{P(NR|D)} \approx \frac{P(D|R)}{P(ND|R)} \quad \text{(I.8)}$$

Tel que P(D|R) (respectivement P(ND|R)) est la probabilité que le document appartienne à l'ensemble R des documents pertinents (respectivement à l'ensemble NR des documents non pertinents).

Différentes méthodes sont utilisées pour estimer ces probabilités :

*La variable document D (t1 = x1, t2 =x2, ..., tn = xn) est représentée par un ensemble d'événements indépendants qui dénotent la présence (xi = 1) ou l'absence (xi = 0) d'un terme dans D.

*Les probabilités P (D/R) et P (D/NR) sont données par :

$$\mathrm{P(D|R)} = (\mathrm{t_1} = \mathrm{x_1}, \mathrm{t_2} = \mathrm{x_2}, \mathrm{t_3} = \mathrm{x_3}, \ldots |\mathrm{R})$$

$$= \prod_i P(t_i = 1|R)^{x_i} * P(t_i = 1|R)^{x_{i-1}} \quad \text{(I. 9)}$$

$$P(D|NR) = (t_1 = x_1, t_2 = x_2, t_3 = x_3, \ldots |R) = \prod_i P(t_i = x_i|NR)$$

$$= \prod_i P(t_i = 1|NR)^{x_i} * P(t_i = 1|NR)^{x_{i-1}} \quad \text{(I. 10)}$$

- En posant :

$$\mathrm{P(t_i = 1|R)} = \mathrm{pi}\ ,\ \mathrm{P(t_i = 1|NR)} = \mathrm{qi}$$

$$\text{RSV (D, Q)} = \frac{{p_i}^{x_i} * (1-p_i)^{(1-x_i)}}{{q_i}^{x_i} (1-q_i)^{(1-x_i)}} \quad \text{(I.11)}$$

En se ramenant à la fonction log et après un petit développement, la fonction RSV s'écrit alors:

$$RSV(D, Q) = \sum_{i;x_i} log \frac{p_i(1-q_i)}{q_i(1-p_i)} \quad \text{(I. 12)}$$

Les premières approches d'application de la théorie de probabilité dans le domaine de recherche documentaire se sont basées sur le principe de PRP. D'autres approches probabilistes adoptant une perspective différente de la notion de pertinence existent. Parmi ces approches nous pouvons citer les modèles de langue que nous décrivons ci-dessous.

I.5.3.2 Modèle de langue

Les modèles statistiques de langue sont utilisés dans plusieurs applications du traitement automatique de la langue : la reconnaissance de la parole, la traduction automatique, et la recherche d'information etc **[2]**.

L'utilisation des modèles de langue en RI remonte à 1998 **[13]**. Le principe de ce modèle est de déterminer la probabilité de générer la requête Q à partir du document D, alors que dans les modèles probabilistes, on évalue la probabilité de pertinence d'un document vis-à-vis d'une requête. Le modèle de langue utilisé est souvent le modèle uni-gramme, Etant donné une requête Q = $(t_1,t_2, \ldots,t_n)$, M_d est le modèle de langue du document D, la pertinence est alors mesurée ainsi :

$$P(Q|M_d) = \prod_{i=1}^{n} P(T_i|M_d) \qquad \text{(I. 13)}$$

$$P(T_i|M_d) = \frac{tf(T_i|D)}{\sum_T tf(T,D)} \qquad \text{(I. 14)}$$

Où $tf(T_i|D)$ est la fréquence du terme Ti dans le document D.

On constate que si un terme de la requête est absent du document, le modèle lui assigne une probabilité nulle.

Afin de résoudre ce problème, des techniques de lissage (smoothing techniques) peuvent être utilisées pour assigner des probabilités non nulles aux termes qui n'apparaissent pas dans le document. Différentes techniques de lissage existent dont le lissage de Laplace, le lissage de Good-turing, le lissage Backoff, le lissage par interpolation **[2][13]**.

I.6 L'évaluation des systèmes de recherche d'information

L'évaluation constitue une étape très importante dans la mise en œuvre d'un SRI, car elle permet de mesurer les caractéristiques du système en termes de qualité de service et de facilité d'utilisation, elle permet également de paramétrer le modèle, d'estimer l'impact de chacune de ses caractéristiques et de fournir des éléments de comparaison entre modèles. Cleverdon en1970 avait défini plusieurs mesures de la qualité d'un SRI, dont : le temps de réponse, la présentation des résultats, l'effort requis de l'utilisateur pour retrouver parmi les documents retournés ceux qui répondent à son besoin autrement dit la pertinence qui est évaluée grâce aux deux facteurs : le taux de rappel du système et la précision du système**.**

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 **[13].**

I.6.1 Les mesures d'évaluation d'un SRI

Un SRI idéal est celui qui est capable de trouver tous les documents pertinents et de rejeter tous les documents non pertinents pour une requête utilisateur. Cet objectif est évalué à l'aide de différentes mesures d'évaluation **[13].** On présente ci-dessous les plus utilisées.

I.6.1.1 Rappel et Précision

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

$$rappel = DPR \ / \ \text{DP} \qquad \text{(I. 15)}$$

La précision : est le rapport du nombre de documents pertinents restitués par le système (DPR) sur le nombre total de documents restitués (DR). Elle mesure la capacité du système à rejeter tous les documents non pertinents à une requête donnée, exprimée ainsi :

$$précision = DPR \ \ DR \qquad \text{(I. 16)}$$

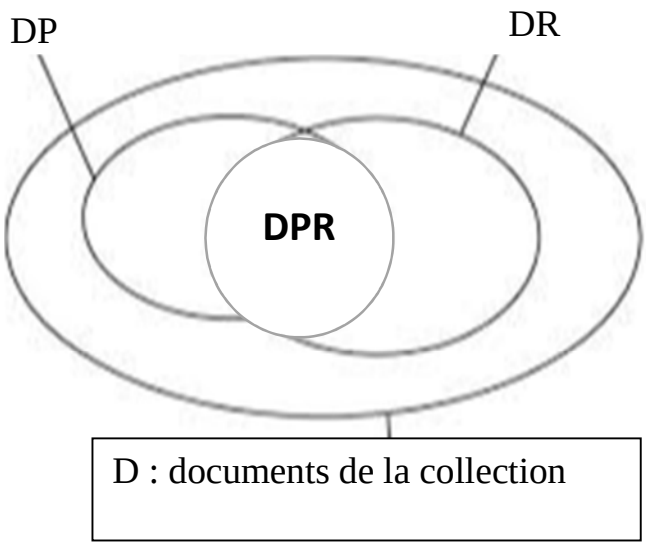


Figure I. 2: Rappel et Précision

I.6.1.2 Mesure harmonique (F-mesure)

La mesure Harmonique F est une fonction qui combine les deux valeurs de précision (P) et de rappel (R), elle est donnée comme suit :

$$F = \frac{2*P*R}{P+R} \quad \text{(I. 17)}$$

La fonction F garantie le compromis entre les deux mesures de rappel et précision **[15].**

I.6.1.3 Courbe Rappel / Précision

Un système idéal devrait retourner tous les documents pertinents et que les documents pertinents ; c'est à dire un taux de précision et de rappel égal à 100%. Cette situation ne se produit pas dans un système réel car le taux de précision et de rappel sont antagonistes. En effet, Lorsque la précision augmente, le rappel diminue et inversement. Ainsi, pour mesurer les performances d'un système il faut utiliser les deux mesures conjointement. Cela est réalisé en calculant la paire des mesures (taux de rappel, taux de précision) à chaque document restitué.

Nous considérons par exemple une requête pour laquelle il existe dix (10) documents pertinents dans le corpus. **Le tableau 1.1** illustre le calcul de la précision et de rappel pour les dix (10) premiers documents retournés par un SRI.

Rang du document renvoyé	Pertinent	Rappel	Précision
Document 1	Oui	1/10=0.1	1/1=1
Document 2	Oui	2/10=0.2	1/1=1
Document 3	Non	2/10=0.2	2/3=0.66
Document 4	Oui	3/10=0.3	3/4=0.75
Document 5	Non	3/10=0.3	3/5=0.60
Document 6	Oui	4/10=0.4	4/6=0.66
Document 7	Non	4/10=0.4	4/7=0.57
Document 8	Oui	5/10=0.5	5/8=0.62
Document 9	Non	5/10=0.5	5/9=0.55
Document 10	Oui	5/10=0.6	6/10=0.6

Tableau I.1: Calcul de précision et de rappel

La figure **I.3** suivante illustre la courbe de rappel et précision correspondante aux résultats du **Tableau I.1**. Pour rendre la courbe lisible on ne garde que la précision calculée à chaque point de rappel (c'est à dire à chaque document pertinent restitué).

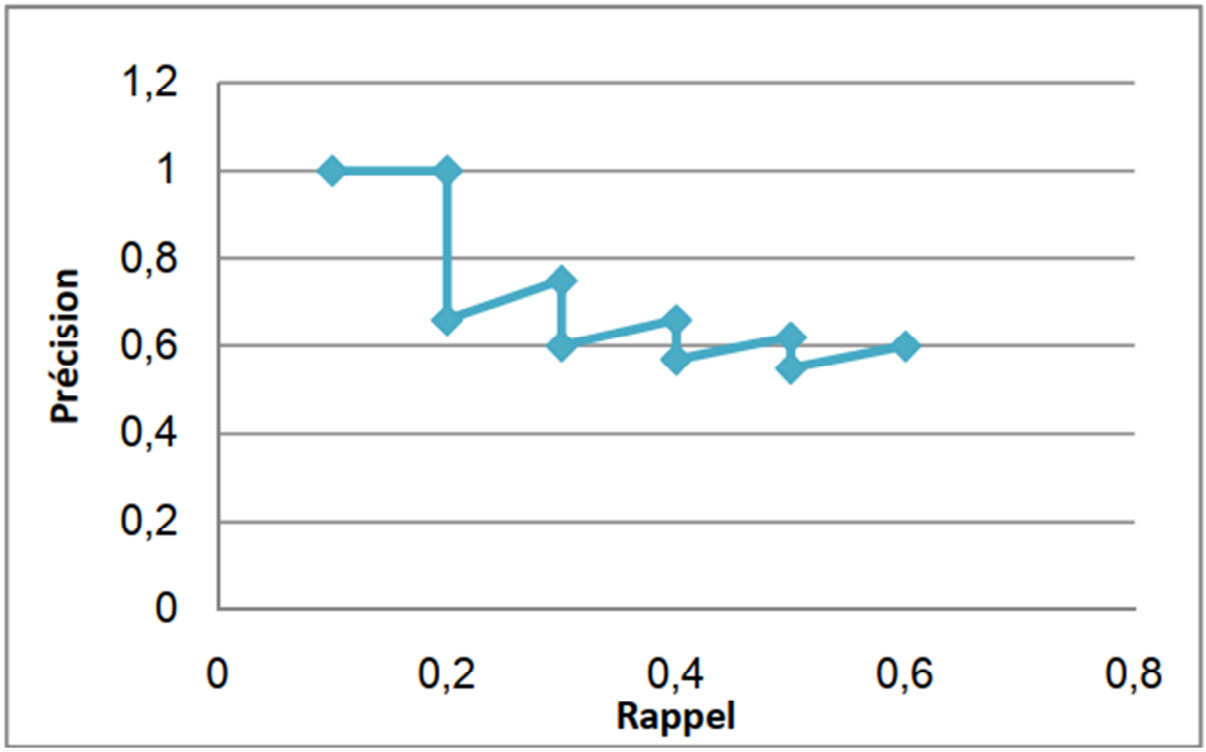


Figure I. 3: courbe rappel/précision

En général, la courbe rappel/précision est basée sur 11 points de rappel standards qui sont varié de 0% à100% avec un pas de 10%. Vu qu'en pratique les niveaux de rappel de chaque requête peuvent être distincts des niveaux standards de rappel, l'utilisation d'une procédure d'interpolation est nécessaire.

Formellement, les mesures de précision aux 11 niveaux standards de rappel sont interpolées comme suit :

Si r_j est la référence au j^{ieme} niveau standard de rappel :

$$P(r_j) = \max_{r_j \leq r < r_j + 1} P(r) \qquad \text{(I.18)}$$

P (r_j) est la précision interpolée au niveau standard de rappel r_j **[3].**

I.6.1.4 La précision exacte

Notée aussi R-précision, elle est calculée sur les « R » premiers documents retournés par le système, sachant que la requête admet « R » documents pertinents.

I.6.1.5 La précision moyenne non interpolée (MAP)

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

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

$$AP_q = \frac{1}{N} \sum_{i=1}^{N} (pr(d_i)) \quad \text{(I.19)}$$

Avec

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

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

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

$$MAP = \frac{1}{M} \sum_{j=1}^{M} AP_{q_j} \quad \text{(I.20)}$$

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

I.6.1.6 La précision à N document

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

$$P@N = \frac{DPR}{N} \quad \text{(I.21)}$$

I.6.2 Collections de test

Une collection (ou un corpus) de test constitue un moyen d'évaluation des SRI. Elle comprend un ensemble de documents à indexer sur le système qui sera évaluée, une liste de requêtes prédéfinies ainsi que les jugements de pertinence, manuellement

établis pour chaque requête (liste de documents jugés pertinents pour cette requête). L'évaluation d'un SRI consiste à comparer les résultats retournés par ce dernier par rapport aux jugements de pertinence. Les collections de test sont les résultats de projets d'évaluation qui se sont multiplié depuis les années 1970. La popularité des collections de tests dans l'évaluation a prospéré en grande partie grâce à des campagnes telles que : TREC, NTCIR, CLEF, Amaryllis, INEX **[16].** Dans ce qui suit nous détaillerons la collection TREC.

I.6.2.1 TREC

Le projet TREC est un programme qui financé par la DARPA (Defense Advanced Research Projects Agency) et le NIST (National Institute of Standards and Technology). La première compagne de TREC voit le jour en 1992 avec 25 participants issus du monde académique.

Ce programme met à la disposition des participants un ensemble de documents et de requêtes. Pour chaque requête, l'ensemble des documents pertinents est déterminé par de juges humains.

Dans ce qui suit, nous allons définir les différents éléments qui constituent le projet TREC :

- **Tâches** : l'objectif est de permettre l'évaluation d'approches spécifiques en recherche d'information concentrant le filtrage, le croisement de langues et la recherche.
- **Les participants** : 25 groupes ont participé à la première édition de TREC et 66 groupes à TREC8

- **Source d'informations** : les documents de la collection sont issus de la presse écrite en 1999
- **Structure et principe de construction de la** collection : un document TREC est identifié par un numéro et décrit par un auteur, une date de production et un contenu textuel. Une requête est également identifiée par un numéro et décrite par un sujet générique **[17].**

La figure suivante illustre un exemple d'un document TREC.

```
<DOC>
<DOCNO> AP880218-0117 </DOCNO>
<FILEID>AP-NR-02-18-88 1130EST</FILEID>
<FIRST>r a PM-LemonadeStands     02-18 0157</FIRST>
<SECOND>PM-Lemonade Stands,0160</SECOND>
<HEAD>House Passes Bill to Allow Lemonade Stands Without
License</HEAD>
<DATELINE>NASHVILLE, Tenn. (AP) </DATELINE>
<TEXT>
   A bill to allow youngsters to have
lemonade stands and sell snacks without a license was
prompted
after the ``big, bad Health Department'' shut down several
children's stands at a jamboree.
   The bill cleared the House today on a 94-0 vote and now
goes to
the Senate. It would allow children age 16 or younger to sell
baked
goods, soft drinks and other refreshments at public events
without
a license or permit as long as they didn't run the stands
more than
three times a year.
   The bill was amended to include a provision to allow
inspections
of the stands.
   The sponsor, Frank Buck, said some children, including his
own,
had lemonade stands and sold cookies at Smithville's annual
fiddlers jamboree, but were shut down.
   ``The big, bad Health Depeartment came by and put them out
of
business,'' he said.
</TEXT>
</DOC>
```

Figure I. 4: exemple d'un document TREC

Les requêtes représentent le besoin en information de l'utilisateur, mais dans TREC elle représente aussi un moyen pour évaluer un SRI. La forme des requêtes a notamment évolué. Au début la forme était très structurée, elle comportait des champs permettant la structuration des requêtes (un titre, le thème, une description et l'objet de la recherche). Pour alléger la forme des requêtes, seuls le titre et une description très brève (d'une phrase ou deux) sont conservés. Un exemple d'une requête TREC est montré dans la figure suivante :

```
<top>
<head> Tipster Topic Description
<num> Number:  054
<dom> Domain:  International Economics
<title>  Satellite Launch Contracts
<desc> Description:

Document will cite the signing of a contract or preliminary agreement, or the
making of a tentative reservation, to launch a commercial satellite.

<smry> Summary:

Document will cite the signing of a contract or preliminary agreement, or the
making of a tentative reservation, to launch a commercial satellite.

<narr> Narrative:

A relevant document will mention the signing of a contract or preliminary
agreement , or the making of a tentative reservation, to launch a commercial
satellite.

<con> Concept(s):

1. contract, agreement

2. launch vehicle, rocket, payload, satellite

3. launch services, commercial space industry, commercial launch industry

4. Arianespace, Martin Marietta, General Dynamics, McDonnell Douglas

5. Titan, Delta II, Atlas, Ariane, Proton
<fac> Factor(s):
<def> Definition(s):
</top>
```

Figure I. 5: exemple d'une requête TREC

I.7 Conclusion

Dans ce chapitre nous avons présenté les principales notions et concepts de la recherche d'information. Nous avons, particulièrement, introduit des notions de base, telles que le besoin en information, la requête, le document et la pertinence. Nous avons aussi décrit les processus de base de la RI, à savoir l'indexation, la recherche et la reformulation des requêtes. Ensuite, nous avons étudié le processus d'indexation et les différents types d'indexation et nous avons traité les différents modèles de la RI. Enfin, l'évaluation des systèmes de recherche d'information. Le chapitre suivant est consacré à la présentation des facteurs de pondérations.

Chapitre II

Les Facteurs de Pondération

Chapitre II : Les facteurs de pondération

II.1 Introduction

Un système de recherche d'information (SRI) a pour but de sélectionner, dans une collection de documents préalablement enregistrée, l'ensemble des documents pertinents pour une requête utilisateur. Le fonctionnement de la plupart des SRI est basé sur des Modèles de Recherche d'Information (MRI) et ces derniers attribuent un score qui estime la correspondance entre les documents et les requêtes. Ce score est obtenu en utilisant le poids des termes de la requête dans le document.

Le but de ce chapitre est de présenter les différents facteurs de pondérations. Dans un premier temps, nous présentons les facteurs de pondération classiques à savoir la fréquence de terme dans le document, la fréquence inverse en documents et la longueur de document. Dans un second temps, nous présentons les facteurs de pondération supplémentaires utilisés pour réévaluer le poids d'un terme dans un document, parmi ces facteurs on peut citer : les modèles thématiques, l'information temporelle, les résultats de recherches antérieures, le word embedding et la position de termes.

II.2 La pondération des termes

La pondération est une fonction fondamentale en RI. Tous les modèles de recherche d'information, excepté le modèle booléen, se basent sur la pondération des termes. Elle consiste à affecter un poids numérique à chaque terme d'un document ou une requête, cette technique permet la caractérisation de ces derniers. Il existe une relation linéaire entre l'importance d'un terme et son poids, plus un terme est important plus son poids doit être élevé, et inversement, plus un terme est insignifiant, plus son poids doit être faible. Ceci dit que le poids est une mesure statistique de l'importance du terme dans un document. Dans la littérature, plusieurs schémas de pondération ont été proposés. La majorité de ces schémas prennent en compte la pondération locale (Term Frequency) et la pondération globale (Inverse Document Frequency). D'autres facteurs ont été proposés pour y'aller au-delà de facteurs classiques.

II.3 Les facteurs de pondération classiques

Le score d'un document vis-à-vis d'une requête est basé sur le poids des termes de la requête dans ce document et il combine généralement trois statistiques sur les termes (facteurs) : Fréquence des termes dans le document (TF), fréquence dans la collection (CF) ou fréquence inverse des documents (IDF), et la longueur de document (dl).

II.3.1 Fréquence de terme (TF)

La fréquence d'un terme dans un document, souvent utilisée dans la recherche d'informations, nous indique l'importance du terme dans le document. Elle prend en compte les informations locales du terme qui ne dépendent que de document. Elle correspond en général à une fonction de la fréquence d'occurrence du terme dans le document notée TF (Term Frequency) exprimée ainsi :

$$\boldsymbol{TF(t) = 1 + log\ tf(t)} \qquad \textbf{(II. 1)}$$

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

Une autre variante de **TF** consiste à normaliser par la fréquence maximale dans un document, est généralement calculée comme suit :

$$\boldsymbol{TF(t) = \frac{ntf(t)}{max\_freq}} \qquad \textbf{(II. 2)}$$

Où $\boldsymbol{ntf}$(**t**) est la fréquence du terme brut et $\boldsymbol{max\_freq}$ est la fréquence du terme le plus courant dans le document.

II.3.2 La Fréquence Inverse en Documents (IDF)

Quant à la Fréquence Inverse des Documents (IDF), elle prend en compte les informations concernant le terme dans la collection. Un poids plus important doit être assigné aux termes qui apparaissent moins fréquemment dans la collection. Car les termes qui apparaissent dans de nombreux documents de la collection ne permettent pas de distinguer les documents pertinents des documents non pertinents.

Par exemple, le mot "the" apparaît dans presque tous les textes anglais et aurait donc un score IDF très faible car il contient très peu d'informations sur le "sujet". En revanche, si vous prenez le mot "coffee", bien qu'il soit courant, il n'est pas utilisé aussi largement que le mot "the". Ainsi, le mot "coffee" aurait un score IDF plus élevé que le mot "the". Traditionnellement l'IDF (inverse document frequency) est calculé comme suit :

$$\boldsymbol{IDF(t) = log\ \frac{N}{n}} \qquad \textbf{(II. 3)}$$

Où **N** est le nombre total de documents dans la collection, et **n** est le nombre de documents contenant le terme **t**.

Il existe une autre version qui calcule l'importance d'un terme dans la collection, donné par la formule suivante :

$$\mathbf{CF}(\boldsymbol{t}) = \frac{\mathbf{F(t)}}{\mathbf{|C|}} \qquad \textbf{(II. 4)}$$

Où $\boldsymbol{F(t)}$ est le nombre de fois que le terme **t** apparait dans la collection, et $\boldsymbol{|C|}$ est la taille de collection.

II.3.3 Longueur de document

La normalisation de la longueur des documents ajuste la fréquence des termes ou le score de pertinence afin de normaliser l'effet de la longueur des documents sur le ranking (classement) des documents. Sans normalisation de la longueur, les documents longs auraient tendance à être classés au-dessus des documents courts même si le terme en question est important dans le document court.

Par exemple, le modèle BM25 est parmi les modèles qui prennent en compte cette caractéristique de normalisation de la longueur **avgdl** (average document length)**,** la formule de modèle BM25 est exprimée ainsi :

$$\boldsymbol{Score_{BM25}(Q,D)=\sum_{i=1}^{n} IDF(t_i)} \cdot \frac{\boldsymbol{f(t_i,d).(k_1+b)}}{\boldsymbol{f(t_i,d)+k_1\cdot\left(1-b+b.\frac{dl}{avgdl}\right)}} \qquad \textbf{(II.5)}$$

Où $\boldsymbol{IDF(t_i)}$ est la fréquence inverse de document, $\boldsymbol{f(t_i}$, **d**) est la fréquence du terme dans le document d, $\boldsymbol{dl}$ est la longueur du document **d**, **avgdl** est la longueur moyenne des documents dans la collection considérée, et $\boldsymbol{k_1\ et\ b}$ sont des paramètres du modèle.

II.4 Les facteurs de pondération supplémentaires

Plusieurs signaux ou sources de connaissance ont été utilisés comme facteurs supplémentaires pour réévaluer le poids d'un terme dans un document. Nous analysons ci-dessous ces facteurs.

II.4.1 Modèles thématiques (topic models)

La représentation du contenu des documents textuels est un élément essentiel de toute approche de la recherche d'informations (RI). Généralement, les documents sont représentés comme un "sac de mots", ce qui signifie que les mots sont supposés se produire indépendamment. Pour saisir les relations importantes entre les mots, les chercheurs ont proposé des approches qui regroupent les mots en "sujets". Hoffman **[18]** a décrit la technique probabiliste d'indexation sémantique latente (pLSI). Cette approche utilise un modèle de variable latente qui représente les documents comme des mélanges de sujets. Le pLSI a été exploité à la fois comme un modèle

unigramme pour lisser les distributions empiriques de mots et comme un modèle d'espace latent pour fournir une représentation à faible dimension des documents et des requêtes. Le modèle pLSI lui-même pose un problème dans la mesure où sa sémantique générative n'est pas bien définie, il n'existe donc aucun moyen naturel de prédire un document inédit. Liu et Croft **[19]** ont proposé un modèle de langue incorporant la notion d'agglomération (cluster) de documents, qui s'appuie sur l'hypothèse préconisant que les documents similaires répondent aux mêmes besoins d'informations. Chaque cluster est considéré comme un grand document qui traite un thème (sujet) donné ; cette source (cluster) est utilisée pour étendre le modèle du document de telle sorte que les termes présents dans les documents de même cluster que le document concerné auront une plus grande probabilité. Ces deux approches ont donné des améliorations significatives sur des collections de test TREC par rapport au modèle uni-gramme. Néanmoins, ces approches souffrent d'un inconvénient majeur, qui considère que chaque document traite un seul thème. Pour pallier ce problème, Wei et Croft **[20]** ont proposé un modèle de langue basé sur les modèles LDA (Latent Dirichlet Allocation), qui permet de modéliser un document comme une mixture de thèmes (sujet). La formule générale de ce modèle est exprimée ainsi :

$$\boldsymbol{P(t|D) = \lambda_1 P(t|D) + \lambda_2 P(t|C) + \lambda_3 P_{lda}(t|D)} \qquad \textbf{(II. 6)}$$

Où $\boldsymbol{\lambda_1 + \lambda_2 + \lambda_3}$= 1, $\mathbf{P}(\boldsymbol{t|D})$ est l'estimation de la probabilité maximale du terme **t** dans le document **D,** $\mathbf{P}(\boldsymbol{t|C})$ est l'estimation de la probabilité maximale du terme **t** dans la collection **C** et $\boldsymbol{P_{lda}(t|D)}$ est le modèle basé **LDA**, qui permet de représenter un document par un ensemble de thèmes.

II.4.2 Information temporelle

Les moteurs de recherche d'information traditionnels ne tiennent pas compte de la pertinence temporelle lorsqu'ils répondent aux requêtes des utilisateurs. C'est ce qui a conduit les chercheurs d'identifié le besoin de trouver les périodes de temps importantes pour les requêtes sensibles au temps et d'intégrer la pertinence temporelle dans le modèle de classement. D'autres travaux utilisent l'intuition suivante : « Les documents récents tendent à être plus pertinents que les documents anciens » pour estimer la probabilité a priori d'un document, Li et Croft **[21]** ont proposé un modèle de langue qui permet d'intégrer la notion de « temps » dans l'évaluation de pertinence d'un document vis-à vis d'une requête, où ils assignent une plus grande probabilité de pertinence pour les documents ayant une date de création récente. Ainsi, ils expriment La probabilité de pertinence a priori d'un document sachant sa date de création, comme une distribution exponentielle, exprimée ainsi :

$$P(d|T_d) = \lambda e^{-\lambda(T_c - T_d)} \quad \textbf{(II. 7)}$$

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

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

II.4.3 Les résultats de recherches antérieures

Au début de ce chapitre nous avons présenté la notion de pondération des termes en RI. Plusieurs mesures ont été proposées pour formaliser cette notion. Nous présentons ci-dessous la mesure la plus utilisée en RI

$$W(t) = TF(t) * IDF(t) \quad \textbf{(II. 8)}$$

Où $\mathbf{W(t)}$ est le poids du terme t dans le document.

La disponibilité des résultats des recherches antérieures et leurs pertinences ont conduits à une nouvelle méthode de pondération des termes, qui est basée sur le calcul d'un facteur nommé le pouvoir de discrimination (Discrimination Power) **DP [22]**. La formule de pondération des termes en **(II.8)** a été réécrite de la manière suivante pour prend compte du pouvoir discrimination d'un terme :

$$W(t) = TF(t) * IDF(t) * DP(t) \quad \textbf{(II. 9)}$$

Le pouvoir de discrimination d'un terme peut être obtenu à partir de son rôle dans les requêtes passées. En d'autres termes, le calcul de pouvoir de discrimination consiste à cumuler ses rôles en séparant les documents pertinents des documents non pertinents sur de nombreuses recherches passées. Le pouvoir de discrimination DP d'un terme est donné par la formule suivante :

$$DP(t) \cong \frac{P(t|r)}{P(t|\bar{r})} \quad \textbf{(II.10)}$$

Où r et $\bar{r}$ représentent respectivement l'ensemble des documents pertinents et non pertinents.

L'approche proposée ci-dessus porte des similarités avec la nôtre, la seule différence est que notre contribution est basée sur un autre facteur.

II.4.4 Word Embedding

Dans les modèles de recherche traditionnels les termes sont représentés sous forme de symboles discrets qui ne peuvent pas être comparés directement les uns aux autres. De manière conceptuelle, cela équivaut à une représentation unique dans lequel chaque terme est représenté comme un vecteur épars avec une dimension égale à la taille du vocabulaire (chaque dimension correspond à un mot unique), cette technique est connue sous le nom de la représentation one-hot qui permet de représenter un terme donné **t** avec un vecteur de **0** et définissons l'indice correspondant à **t** à **1**. Dans une telle représentation, tous les termes sont orthogonaux et équidistants les uns aux autres et, par conséquent, il est difficile de reconnaitre la similarité des termes dans l'espace vectoriel. Cela a conduit à un problème de désordre de vocabulaire dans lequel un système de recherche d'information (SRI) ne peut pas reconnaitre lorsque les termes sont distincts mais liés qui se produisent dans la requête et le document.

Pour remédier à ce problème, la technique de word embedding est apparue, elle permet de représenter chaque symbole en tant que vecteur à faible dimension. Le word embedding utilise des modèles de représentation des mots, le modèle le plus connu est word2vec, il englobe deux approches qui ont respectivement comme objectif de prédire un mot cible en fonction des mots qui co-occurrent dans une fenêtre de contexte glissante (modèle CBOW) et de prédire les mots du contexte à partir d'un mot cible (modèle Skip-gram). D'autres modèles s'intéressent à la représentation de mots, dont le modèle Global vector (GloVe), qui exploite la co-occurrence globale des mots. Les words embeddings représentent une similitude sémantique et syntaxique dans la mesure où les words embeddings similaires seront proches l'un de l'autre dans l'espace vectoriel. Cela permet d'effectuer des opérations algébriques simples entre les vecteurs de word embedding qui reflètent la signification du mot. Différentes études **[23]** ont été effectuées en intégrant le word embedding dans les modèles de recherche d'information comme suit :

$$\boldsymbol{Score(D,Q) = (\alpha.IRScore(D,Q) + (1-\alpha).WEScore(D,Q))} \qquad \textbf{(II. 11)}$$

Où $\boldsymbol{\alpha}$ est le coefficient de combinaison déterminé par une double validation croisée selon la métrique MAP, $\boldsymbol{IRScore}$ est le score de document obtenu à l'aide d'un modèle de RI classique et $\boldsymbol{WEScore}$ (la similarité cosinus entre les représentations en word embedding de la requête et du document).

II.4.5 La position de termes

La position de termes est un signal crucial pour la réévaluation du poids des termes dans les documents. Dans le domaine de la RI plusieurs travaux **[24] [25] [26]** ont introduit la position

du terme dans le document selon trois directions différentes :

a) Structure du document

L'utilisation de la structure du document comme source d'évidence consiste à pondérer différemment les parties d'un document. Par exemple, la partie titre du document est généralement surpondérée **[27].**

b) Proximité des termes de la requête

La seconde utilisation de la position du terme dans le document, consiste d'abord à mesurer la proximité (dépendance) des termes de la requête dans le document, ensuite à intégrer ce facteur de proximité dans le calcul de la similarité document-requête **[28]**. Metzler et Croft **[29]** ont élaboré un cadre formel pour la modélisation des dépendances entre termes en utilisant les champs de Markov, nommé (MRF). Une structure de graphe non orienté est utilisée pour modéliser les distributions jointes. Dans ce cadre, ils ont proposé de modéliser deux types de dépendance : dépendance séquentielle (SD : Sequential Dependency), capturant les relations entre les paires de termes adjacents de la requête, et la dépendance complète (FD : Full Dependency), capturant les relations entre toutes les paires de termes de la requête. Ces deux modèles de dépendance ont été interpolés linéairement avec un modèle uni-gramme, selon la formule suivante :

$$P_{\wedge}(D|Q) = \sum_{c\in T} \lambda_T f_T(c) + \sum_{c\in O} \lambda_O f_O(c) + \sum_{c\in U} \lambda_U f_U(c) \qquad \textbf{(II. 12)}$$

Où T est défini comme étant l'ensemble de deux cliques impliquant un terme de requête et un document D, O est l'ensemble des cliques contenant le nœud document et deux ou plusieurs termes de la requête qui apparaissent de façon contiguë dans la requête, U est l'ensemble des cliques contenant le nœud document et deux ou plusieurs termes de requête apparaissant de façon non contiguë dans la requête, et les fonctions $f_X(c)$ tel que $X \in \{T, O, U\}$ sont présentées dans le tableau suivant :

Nom	Type	Formule
$f_T(q_i, D)$	terme	$\log[(1-\alpha_D)\frac{tf_{q_i,D}}{\lvert D\rvert} + \alpha_D \frac{F_{q_i}}{\lvert C\rvert}]$
$f_O(q_i, \dots q_{i+k}, D)$	Phrase ordonnée	$log[(1-\alpha_D)\frac{tf\#1(q_i \dots\dots q_{i+k}),D}{\lvert D\rvert} + \alpha_D \frac{F_{\#1(q_i \dots q_{i+k})}}{\lvert C\rvert}]$
$f_U(q_i, \dots q_j, D)$	Phrase non ordonnée	$log[(1-\alpha_D)\frac{tf\#UtN(q_i \dots\dots q_j),D}{\lvert D\rvert} + \alpha_D \frac{F_{\#UtN(q_i \dots q_j)}}{\lvert C\rvert}]$

Tableau II. 1: Résumé des fonctions des caractéristiques

Où $tf_{t,D}$ est le nombre de fois que le terme **t** apparait dans le document **D**, $tf\#1(_{q_i \ldots\ldots q_{i+k}})$ indique le nombre de fois que la phrase exacte $(q_i \ldots q_{i+k}$)apparaît dans le document **D**, $tf\#UtN(_{q_i \ldots\ldots q_j})$ est le nombre de fois que les deux termes $(q_i .. q_j)$ se produisent (de manière ordonnée ou non ordonnée) dans une fenêtre de **N** positions, $\boldsymbol{F}$ est le nombre de fois que le terme apparait dans la collection, $|D|$ est la longueur de document, $|C|$ est la longueur de collection, enfin $\boldsymbol{\alpha_D}$ agit comme un paramètre de lissage.

c) Positions exactes du terme dans un document

La troisième direction d'exploitation de la position du terme dans le document, consiste à utiliser les positions exactes du terme dans le document comme facteur de pondération. Peu de travaux ont été réalisés dans cette perspective. L'idée principale de cette catégorie d'approches est de surpondérer les termes qui apparaissent au début du document. Troy et al **[30]** ont introduit une extension du modèle BM25 par l'utilisation de la position exacte du terme dans le document. Cette extension exploite le classement chronologique des termes (Chronological Term Rank : CTR). Ils ont introduit ce classement chronologique des termes représenté par $\boldsymbol{\mathcal{R}}$ soit d'une manière multiplicative comme le montre la formule (**II. 13)**, ou d'une manière additive comme le montre la formule (**II. 14).** Ce facteur $\boldsymbol{\mathcal{R}}$ est avéré avec des expérimentations, qui ont obtenues des meilleurs résultats avec la manière additive.

$$\sum_{t \in d \cap q} \boldsymbol{Ln} \frac{\boldsymbol{N-df+0.5}}{\boldsymbol{df+0.5}} \cdot \frac{\boldsymbol{tf}}{\boldsymbol{0.5+1.5.\frac{dl}{avgdl}+tf}} \cdot \boldsymbol{\mathcal{R}} \qquad (\boldsymbol{II.\ 13})$$

$$\sum_{t \in d \cap q} \boldsymbol{Ln} \frac{\boldsymbol{N-df+0.5}}{\boldsymbol{df+0.5}} \cdot \left(\frac{\boldsymbol{tf}}{\boldsymbol{0.5+1.5.\frac{dl}{avgdl}+tf}} + \boldsymbol{\mathcal{R}} \right) \qquad (\textbf{II. 14})$$

Où $\boldsymbol{df}$ est le nombre de document qui contient le terme $\boldsymbol{t}$, $\boldsymbol{avgdl}$ est la longueur moyenne des documents de la collection et $\boldsymbol{tf}$ est la fréquence du terme $\boldsymbol{t}$ dans le document

Plusieurs fonctions ont été testées pour formaliser le facteur $\boldsymbol{\mathcal{R}}$, nous présentons ci-dessous la fonction qui a montré les meilleurs résultats dans l'étude de Troy et al **[30].**

$$\boldsymbol{\mathcal{R} = C - \left(C.D \frac{log\left(\frac{t_r - 1}{20} + 10\right)}{log\left(\frac{dl}{20} + 10\right)} \right)} \qquad \textbf{(II. 15)}$$

Où **C** et **D** sont des constantes $\in[0,1]$, $\boldsymbol{t_r}$ est la position de la première occurrence du terme **t** et $\boldsymbol{dl}$ est la longueur du document.

Notre approche à des similarités avec le modèle CTR. Cependant, au lieu d'utiliser uniquement la position de la première apparition du terme dans le document, notre approche prend en compte aussi toutes les positions du terme dans tous les documents de la collection. De plus, dans ce cas la position est intégrée d'une manière additive dans le modèle BM25, par contre dans notre approche nous avons constaté que l'intégration de ce facteur position est plus efficace en tant que complément multiplicatif dans les deux modèles TF_IDF et BM25.

II.5 Conclusion

Ce deuxième chapitre nous a permet de présenter les facteurs de pondération. Particulièrement les points suivants ont été abordés. En premier lieu, nous avons présenté les facteurs de pondération classiques. Ensuite, nous avons présenté les facteurs de pondération supplémentaires. Notre approche s'insère dans la catégorie des travaux qui exploitent les positions de terme dans les documents de la collection.

Dans le chapitre suivant nous détaillons notre approche, ainsi que les résultats des expérimentations obtenues.

Chapitre III

Description et évaluation de notre approche

Chapitre III : Description et évaluation de notre approche

III.1 Introduction

Un Modèle de Recherche d'Information a pour but de classer les documents. La plupart des modèles de la RI utilisent la combinaison de facteurs classiques dans leur fonction de pondération. Ces facteurs sont : TF (Term Frequency) et IDF (Inverse Document Frequency). Comme il existe aussi d'autres facteurs utilisés pour reévaluer le poids d'un terme dans le document (présenté dans le chapitre II).

En plus des paramètres de pondération classiques, dans notre approche nous prenons en compte le facteur position des termes dans le document sous l'hypothèse suivante : "les termes qui apparaissent au début du document sont les meilleurs représentants du contenu du document, par conséquent, ils doivent être surpondérés". L'importance du terme diminue progressivement en allant vers la fin du document. Nous précisons que nous considérons toutes les positions du terme dans le document.

Ce chapitre est organisé comme suit, dans la première section nous allons présenter les modèles de recherche d'information de base. Dans la seconde section nous allons décrire notre approche et son architecture. Par la suite nous allons introduire l'environnement de développement de notre approche en précisant les outils et le langage utilisés pour sa mise en œuvre. Enfin nous présentons les résultats obtenus sur la collection de test TREC AP88.

III.2 Modèles de recherche d'information de base utilisés

Nous considérons dans notre étude deux modèles de base de RI : TF_IDF et BM25.

A) Le modèle TF_IDF :(Term Frequency et Inverse Document Frequency) est un modèle utilisé pour déterminer la pertinence d'un document pour une requête. La formule prend en compte la fréquence du terme ($\boldsymbol{tf}$) dans un document donné et la fréquence du terme dans la requête ($\boldsymbol{tf_q}$), ainsi que le nombre de documents contenant ce terme dans la collection ($\boldsymbol{IDF}$). La formule du modèle TF_IDF est donnée comme suit :

$$\boldsymbol{score_{TF\_IDF}(D,Q) = \sum_{i=1}^{n} tf_q(t_i) * \frac{k_1.tf(t_i)}{tf(t_i)+k_1\left(1-b+b.\frac{dl}{avgdl}\right)} * IDF(t_i)} \qquad \textbf{(III. 1)}$$

Où $\boldsymbol{dl}$ est la longueur du document **d**, $\boldsymbol{avgdl}$ est la longueur moyenne des documents dans la collection considérée, et $\boldsymbol{k_1}$ *et* $\boldsymbol{b}$ sont des paramètres libres.

B) Le modèle BM25 : est un modèle probabiliste qui ordonne les documents en fonction de la fréquence des termes qui apparaissent dans chaque document, il existe toute une famille de fonctions attribuant un score à chaque document pour une requête donnée. L'une des formes les plus connues de cette famille de fonctions est la suivante :

$$score_{BM25}(D,Q)=\sum_{i=1}^{n} \frac{tf(t_i).(k_3+1).tf_q(t_i)}{tf(t_i)+k_1.\left(1-b+b.\frac{dl}{avgdl}\right).\left(k_3+tf_q(t_i)\right)} * IDF(t_i) \qquad \textbf{(III.2)}$$

Où $IDF(t_i)$ est la fréquence inverse de document, $\mathbf{tf(t_i)}$ est la fréquence du terme $\mathbf{t_i}$dans le document **d**, tf_q la fréquence du terme $\mathbf{t_i}$ dans la requête **Q**, dl est la longueur du document **d**, $avgdl$ est la longueur moyenne des documents dans la collection considérée, et k_1, k_3 *et* b sont des paramètres libres.

III.3 Approche proposée

Dans cette section nous présentons dans un premier temps notre approche, par la suite nous présentons l'extension des modèles de base avec notre approche.

III.3.1 Description de notre approche

La représentation en sac de mots des documents est largement utilisée en recherche d'information car elle est simple à mettre en œuvre. Cependant, cette représentation ignore la position des termes dans le document, ce qui ne permet pas de bien capturer la sémantique du document.

Dans notre approche nous prenons en compte la position du terme dans le document sous l'hypothèse suivante : "les termes qui apparaissent au début du document sont les meilleurs représentants du contenu du document, par conséquent, ils doivent être surpondérés". Afin de mettre en œuvre cette hypothèse, nous proposons une nouvelle technique d'estimation du poids du terme en se basant sur ses positions dans tous les documents de la collection. Ensuite nous intégrons les facteurs obtenus dans les modèles de base : TF_IDF et BM25.

III.3.2 Formalisation de notre approche

Dans cette section nous allons présenter la formalisation du facteur position.

En se basant sur l'hypothèse décrite au-dessus, nous proposons un cas de figure qui exploite toutes les positions du terme dans les documents de la collection. En se basant sur cette intuition nous proposons de formaliser le facteur position en quatre (04) étapes.

Chapitre III : Description et évaluation de notre approche

Dans notre modèle, nous considérons une requête $\boldsymbol{Q}$ et un document $\boldsymbol{D}$ représentés avec le vocabulaire suivant : $V = \{t_1 \dots t_i \dots t_n\}$.

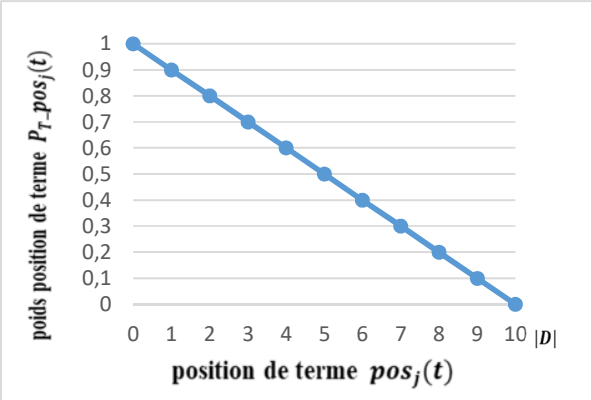
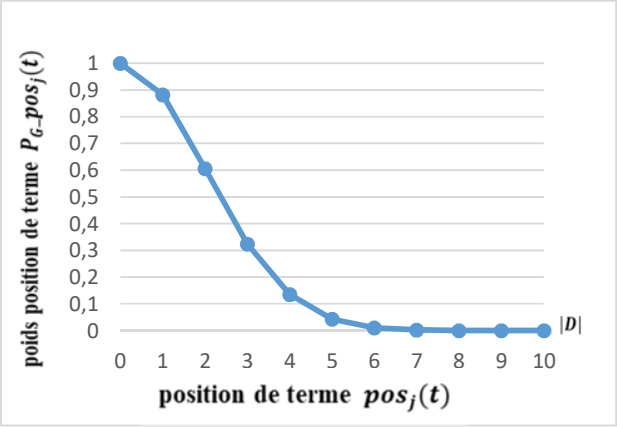
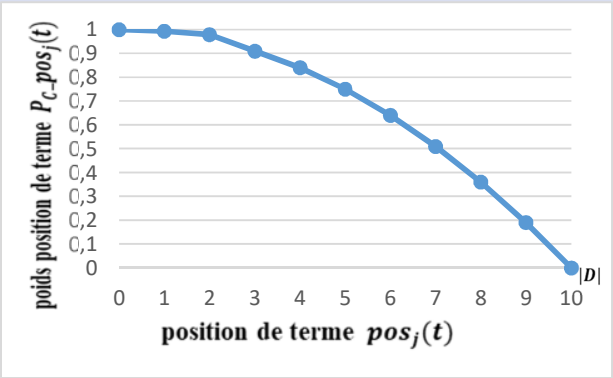
Nous assumons qu'un document D est représenté comme un ensemble de triplets, comme suit :

$$D = \{(t_1; tf(t_1); Pos(t_1) \dots (t_i; tf(t_i); Pos(t_i) \dots (t_n; tf(t_n); Pos(t_n)$$

Chaque triplet contient: l'intitulé du terme $\boldsymbol{t_i}$, la fréquence du terme dans le document $\boldsymbol{tf(t_i)}$ et les positions absolues du terme dans le document $Pos(t_i) =< pos_1(t_i) \dots pos_j(t_i) \dots pos_{tf(t_i)}(t_i) >$

Etape1

En se basant sur cette représentation, nous assignons un poids pour chaque position $\boldsymbol{Pos(t)}$ du terme **t** dans le document $\boldsymbol{D}$. Précisément, nous attribuons un poids élevé aux premières positions du document et ce poids diminue progressivement vers la fin du document. Pour répondre à cette exigence, nous proposons d'utiliser trois fonctions pour estimer le poids du terme $\boldsymbol{t}$ apparaissant à la position $\boldsymbol{Pos(t)}$ dans le document $\boldsymbol{D}$, et cela est donné dans le tableau suivant :

Fonction	Graphe
Triangle $P_{T\_}pos_j(t) = \frac{\|D\| - pos_j(t)}{\|D\|}$ **(III. 3)**	
Gaussienne $P_{G\_}pos_j(t) = e^{-\frac{1}{2}(\frac{pos_j(t)}{\lambda.\|D\|})^2}$ **(III. 4)** Où λ est un paramètre pour contrôler la dispersion du "poids de la position" dans le document.	
Cercle $P_{C\_}pos_j(t) = \sqrt{1 - (\frac{pos_j(t)}{\|D\|})^2}$ **(III.5)**	

Où $P_{X\_}pos_j(t)$ est le poids de la position pos_j du terme t dans le document D, tel que $X \in \{T, G, C\}$, qui représente respectivement les fonctions **{Triangle, Gaussienne, Cercle}**, $|D|$ est la taille du document.

Etape2

Nous prenons en compte toutes les positions du terme dans le document (noté $P_{X\_All_{pos_j(t)}}$), nous formalisons alors les poids cumulés au niveau de document comme suit :

$$P_{X\_All_{pos_j(t)}} = \sum_{t \in D} P_{X\_}pos_j(t) \quad \textbf{(III.6)}$$

Etape3

Afin d'obtenir le score global d'un terme selon les positions dans lesquelles il apparait dans les documents de la collection, nous formalisons les poids cumulés au niveau des documents de la collection comme suit :

$$score_{X\_}pos_j(t) = \sum_{D \in C} P_{X\_All_{pos_j(t)}} \quad \textbf{(III.7)}$$

Etape4

Finalement pour avoir le score final, nous normalisons le $score_{X\_}pos_j(t)$ obtenu. Le score final est noté ($scorefinal_{X\_}pos_j(t)$), il est donné par le ratio suivant:

$$scorefinal_{X\_}pos_j(t) = \frac{score_{X\_}pos_j(t)}{CF(t)} \quad \textbf{(III.8)}$$

Où **CF(t)** est la fréquence du terme **t** dans la collection.

III.3.3 Extension des modèles de base

Dans notre approche nous prenons en compte la position du terme dans tous les documents de la collection. Afin de mettre en œuvre cette approche nous proposons d'intégrer le score final des positions de terme normalisé défini par la formule **(III. 8)** aux modèles de base : TF_IDF et BM25, qui sont respectivement exprimés dans les formules **(III. 1), (III. 2)** comme suit :

A) Le modèle TF_IDF_X :

$$score_{TF\_IDF_X}(D, Q) = \sum_{t \in Q} \mathbf{score}_{\mathbf{TF\_IDF}}(\mathbf{D}, \mathbf{Q}) * scorefinal_{X\_}pos_j(t) \quad \textbf{(III.9)}$$

B) Le modèle $BM25_X$:

$$score_{BM25_X(D,Q)} = \sum_{t \in Q} \mathbf{score_{BM25}}(\mathbf{D}, \mathbf{Q}) * scorefinal_X\_pos_j(t) \quad \textbf{(III.10)}$$

Où **X∈{T, G, C}**, qui représente respectivement les fonctions **{Triangle, Gaussienne, Cercle}**

III.3.4 Architecture de notre approche

Dans notre approche nous utilisons le facteur position pour l'estimation de la pertinence finale (score) dans les modèles de recherche d'information modifié (TF_IDF_X,$BM25_X$). Nous présentons dans la figure suivante l'emplacement de notre approche dans l'architecture d'un SRI, ensuite nous expliquons les étapes de notre approche.

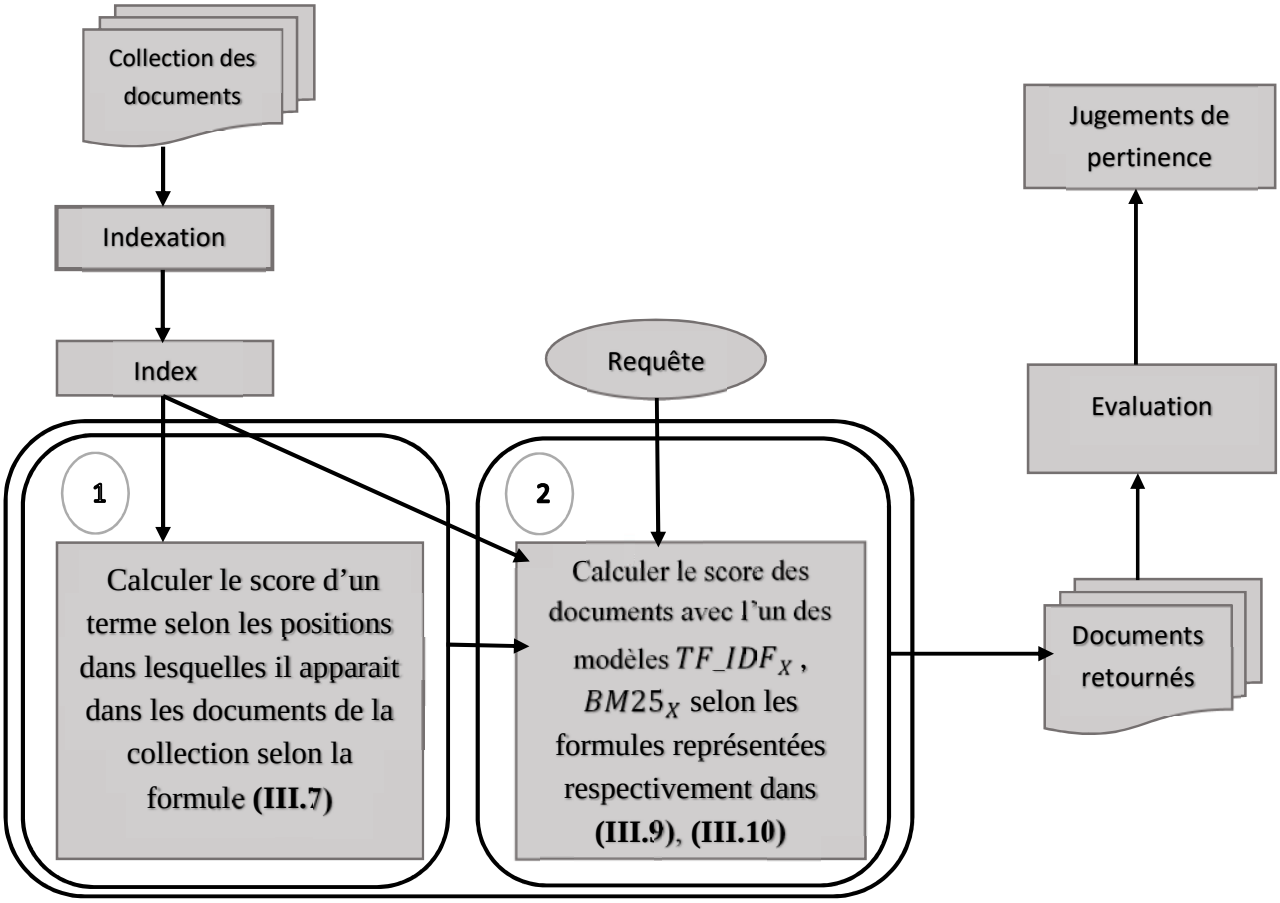


Figure III. 1: Emplacement de notre approche dans un SRI

D'après **la Figure III.1 :**

1 **:** L'index est fourni en entrée pour le calcul de score global du terme selon les positions dans lesquelles il apparait dans les documents de la collection. Le processus de calcul de ce score est détaillé dans l'algorithme suivant

Début

Choisir la fonction qui calcule score position des termes {Triangle, Gaussienne ou Cercle}

Récupérer l'index ;

Pour (i=0 jusqu'à taille de la collection)

Récupérer le terme t_i

Déclarer et initialiser la variable $score_{X\_}pos_j(t_i)$

Récupérer l'ensemble des documents qui contient le terme t_i

Pour (k=0 jusqu'au nombre des documents qui contient le terme t_i)

Récupérer toutes les positions du terme t_i dans le document D_K et leur taille

Pour (j=0 jusqu'au nombre position)

Si (la tailleDoc différente de zéro)

$score_{X\_}pos_j(t_i)$**+= calcule_score_position (tailleDoc, position[j])**

Fin Si

Fin Pour

Fin Pour

Fin Pour

Enregistrer le score final des positions de chaque terme dans un fichier

Fin

Le résultat de cet algorithme est stocké dans un fichier. Nous présentons ci-dessous un extrait de ce fichier :

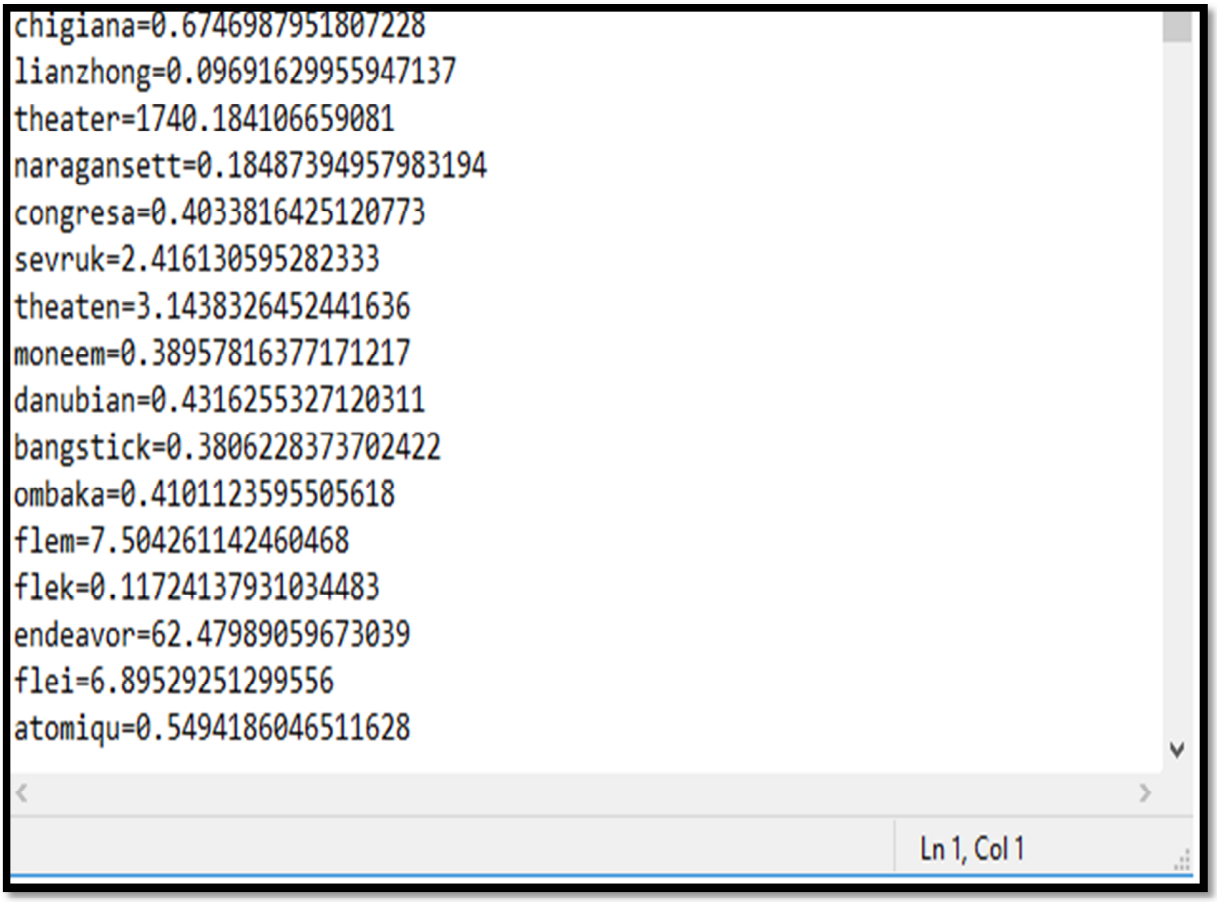


Figure III. 2: Extrait du fichier contenant le score global des positions de chaque terme

2 **:** L'index, la requête et le résultat ressorti en algorithme 1 sont passés au modèle de recherche étendu, afin d'estimer le score final de chaque document.

L'estimation des scores des documents pour une requête donnée est présentée dans l'algorithme suivant :

Début

Récupérer l'index, les termes_requête et le fichier qui contient les scores basés sur les positions des termes

Déclarer et initialiser la variable $\boldsymbol{score\_modèle\_de\_recherche_X(D_k, Q)}$

Pour (i=0 jusqu'au nombre de terme dans la requête)

Récupérer le terme $\boldsymbol{t_i}$

Pour (k=0 jusqu'au nombre de document qui contient le terme $\boldsymbol{t_i}$)

Récupérer le document $\boldsymbol{D_k}$

$\boldsymbol{score\_modèle\_de\_recherche_X(D_k, Q) += modèl\_de\_rechereche * scorefinal_X\_pos_j(t_i)}$

Fin Pour

Fin Pour

Fin

III.4 L'environnement de développement

Dans ce qui suit nous allons présenter les différents outils utilisés pour mettre en œuvre notre approche à savoir la plateforme Terrier, le langage de programmation java, et l'environnement Netbeans.

III.4.1 Terrier [1]

Terrier est une plateforme modulaire pour le développement rapide des applications de recherche d'informations RI à grande échelle. Il peut indexer diverses collections de documents, y compris les collections TREC et Web.

Terrier est un moteur de recherche open source très flexible, efficace et performant, met en œuvre des fonctionnalités d'indexation, de recherche et d'évaluation. Il est écrit en Java, fonctionne sous différentes plateformes comme Windows et Linux dans notre cas on a utilisé la version sous Windows, téléchargé sur le site officiel http://terrier.org/download/ . Dans notre approche on a utilisé la version Terrier 3.5.

Chapitre III : Description et évaluation de notre approche

L'architecture de la plateforme Terrier distingue les deux phases classiques, l'indexation et la recherche. L'indexation décrit le processus au cours duquel Terrier analyse une collection de documents et représente l'information dans la collection sous la forme d'un index qui contient des statistiques sur la fréquence des termes dans chaque document et dans l'ensemble de la collection. La recherche décrit le processus au cours duquel Terrier pondère chaque terme de document et estime la pertinence probable d'un document pour une requête, sur la base de ces pondérations de termes. La figure suivante présente l'architecture générale de Terrier.

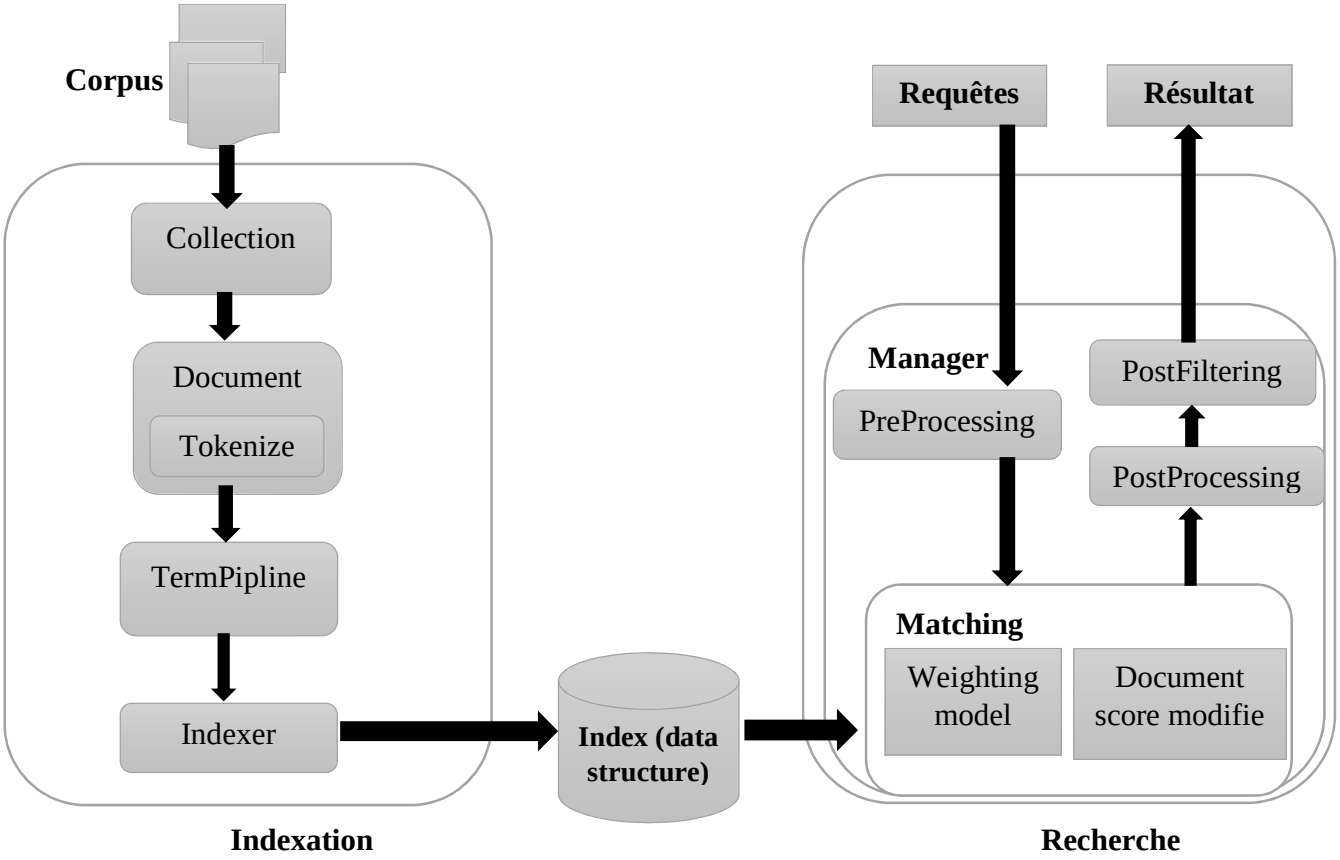


Figure III. 3: L'architecture du Terrier

Terrier prend en entrée un corpus documentaire. Les documents de la collection passent par un ensemble de prétraitements tels que la tokenisation. Les tokens sont ensuite injectés dans une chaîne de traitement TermPipLines, à savoir le StopWords pour l'élimination des mots vides de sens. La phase d'indexation conduit à la construction de l'index (Data structures). Dans la phase de recherche un utilisateur émet une requête vers le cadre Terrier, dans un premier temps, la requête est analysée, ensuite sera transmise au composant PreProcessing qui à son tour la prétraité en

l'appliquant à la TermPipeline configurée. Après le prétraitement, la requête sera transmise au composant Matching. Ce dernier est responsable de l'initialisation du WeightingModel qui représente le modèle de recherche qui est utilisé pour pondérer les termes d'un document, et de DocumentScoreModifiers appropriés, qui est responsable du calcul des scores des documents. Comme dans notre étude, afin de réaliser notre approche, des sous classes ont été créés à partir des classes Matching et weightingModel, ensuite nous avons créé des classes de modèle TF_IDF étendu et BM25 étendu. Et PostProcessing, c'est le ResultSet peut être modifié de n'importe quelle manière. Le PostFiltering est plus simple, il permet d'inclure ou d'exclure des documents. Le résultat renvoyé est la liste des documents jugés pertinents et leurs scores respectifs.

III.4.2 Le langage de programmation java [2]

Java est un langage de programmation moderne né en 1995 développé par Sun Microsystems, aujourd'hui racheté par Oracle. C'est un langage orienté objet, inspiré du langage C++ et compilé en bytecode qui est un langage intermédiaire indépendant de la machine, ce dernier est interprété par une machine virtuelle il est également typé (toute variable doit être déclaré avec un type qui est fournis soit par le langage ou par la définition des classes).

- **Java est un langage de programmation orienté objet**

Comme la plupart des langages récents, Java est orienté objet. Chaque fichier source contient la définition d'une ou plusieurs classes qui sont utilisées les unes avec les autres pour former une application. Java n'est pas complètement objet car il définit des types primitifs (entier, caractère, flottant, booléen,...).

- **Java est portable**

Une fois le programme java est créé, ce dernier est portable dont le code peut être exploité dans différents environnements (Windows, Mac, Linux, etc.).

- **Java est interprété**

Un code source doit être traduit dans le langage machine avant d'être exécuté. Le Compilateur java traduit le code source java en bytecode par la suite un interpréteur java spécifique à une machine donnée (JVM : Java Virtuel Machine) traduit et exécute le bytecode.

III.4.3 Netbeans [3]

L'EDI Netbeans est un environnement de développement pour java, placé en open source par Sun en juin 2000. En plus de java, il peut également supporter python, C, C++, XML, et HTML, se concentrant principalement sur simplifier le développement d'applications java. L'EDI est lui-même écrit en java, ce qui permet de le faire tourner sur n'importe quel système d'exploitation.

NetBeans est disponible sous Windows, Linux, Mac OS X ou sous une autre version indépendante des systèmes d'exploitation (requérant une machine virtuelle java). Un environnement java développement Kit JDK est requis pour les développements en java.

La figure suivante illustre l'environnement de développement Netbeans avec l'interface de notre approche :

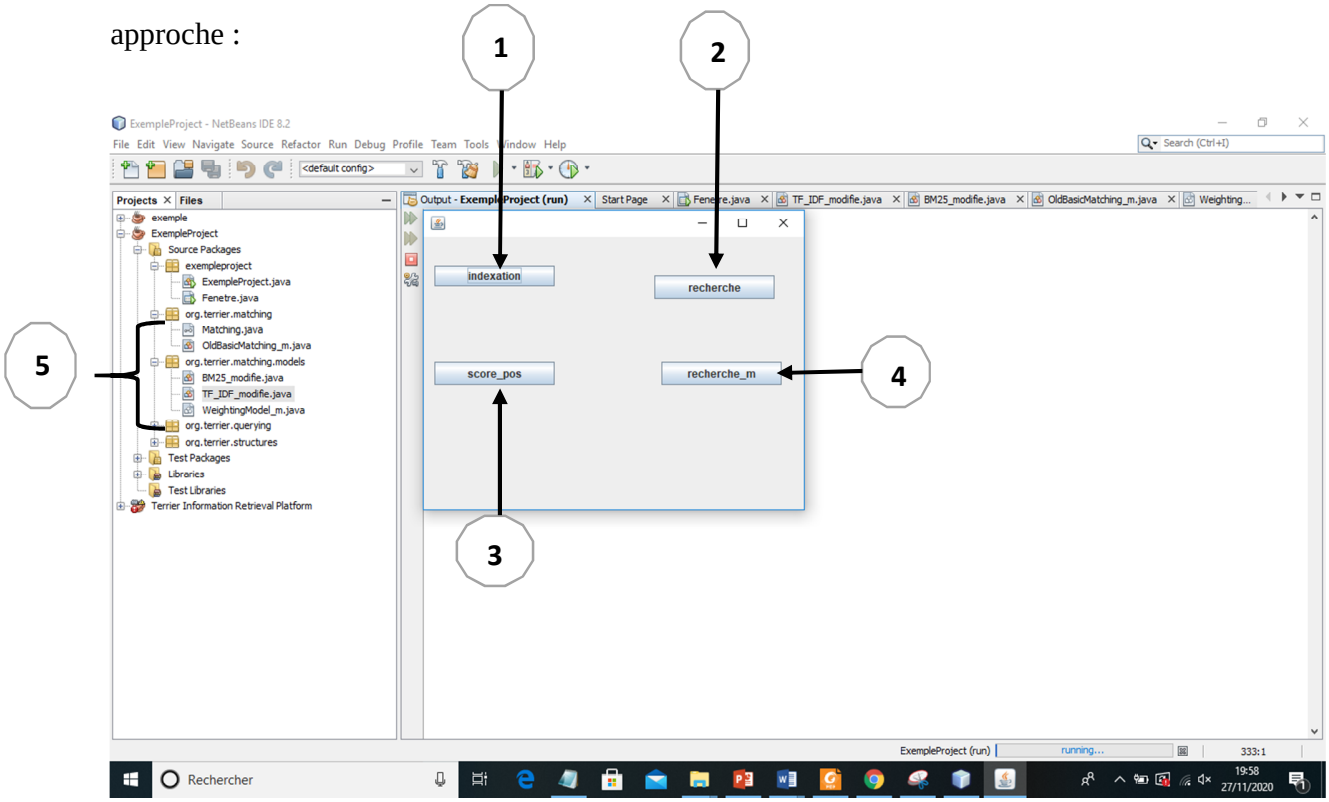


Figure III. 4: Environnement de développement de Netbeans avec l'interface de notre approche

1 : Le bouton « Indexation » permet d'effectuer l'indexation de la collection avec la classe « TRECIndexing ».

2 : Le bouton « recherche » responsable d'effectuer la correspondance entre l'index et les requêtes avec la classe « TRECQuerying ».

3 : Le bouton « score_pos » permet le calcul du score global des positions des tous les termes de la collection.

4 : Le bouton « recherche_m » responsable d'effectuer la correspondance entre l'index et les requêtes après l'extension des modèles de base.

5 : représente les classes crée pour mettre en œuvre notre approche sous Terrier :

- La classe OldBasicMatching_m : est une extension de la classe OldBasicMatching, permet d'effectuer la correspondance des documents avec une requête, en attribuant d'abord des scores aux documents pour chaque terme de requête et en modifiant ces scores avec les modificateurs appropriés. Ensuite, une série de modifications de score de document sont appliquées si nécessaire.
- La classe TF_IDF_modifie : implémente le modèle TF_IDF avec l'extension que nous avons proposée.
- La classe BM25_modifie : implémente le modèle BM25 avec l'extension que nous avons proposée.
- La classe WeightingModel_m: est une extension de la classe WeightingModel, elle doit être étendue par les classes utilisées pour la pondération des termes et des documents.

III.5 Evaluation et résultats

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

III.5.1 La collection de test et les requêtes utilisées

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

Pour la recherche nous avons utilisé 50 requêtes issues des topics numérotés de « 51-100 » de la collection TREC.

III.5.2 Mesures d'évaluation utilisées

Afin d'évaluer les différents modèles, nous avons utilisé trois métriques (voir chapitre I section I.6.1) : la MAP (Mean Average Precision), et les Précisions à N documents (P@5 et P@10).

III.5.3 Présentation des résultats obtenus

Pour évaluer les performances de notre approche, nous devons tout d'abord fixer la fonction qui estime le score des positions des termes. Pour le cas de la fonction Gaussienne nous devons fixer la valeur du paramètre λ, nous avons varié sa valeur de 0 à 1 avec un pas de 0.1.

III.5.3.1 Résultats globaux

Nous présentons dans cette section les résultats de l'évaluation de notre approche par rapport aux modèles de recherche de base : **TF_IDF** et **BM25**. Nous avons noté les modèles : **TF_IDF_X** et **BM25_X** tel que $\mathbf{X \in \{T, G, C\}}$, qui représentent respectivement les fonctions **{Triangle, Gaussienne, Cercle}**, nous avons aussi noté les modèle : **TF_IDF_G_λ** et **BM25_G_ λ** tel que $\boldsymbol{\lambda \in [0.1, 1]}$.

Les tableaux ci-dessous présentent les résultats de cette évaluation, où nous avons mis en gras les meilleurs résultats pour chaque métrique.

Modèles de recherche	MAP	Taux d'amélioration de la MAP	P@5	Taux d'amélioration de la P@5	P@10	Taux d'amélioration de la P@10
TF_IDF	0.2947	_____	0.4000	_____	0.3633	_____
TF_IDF_C	0.2941	-0.20%	0.4041	+1.02%	0.3714	+2.22%
TF_IDF_T	0.2974	+0.91%	0.4041	+1.02%	0.3755	+3.35%
TF_IDF_G_0.1	0.3012	+2.20%	**0.4122**	**+3.05%**	**0.3816**	**+5.03%**
TF_IDF_G_0.2	**0.3016**	**+2.34%**	**0.4122**	**+3.05%**	0.3776	+3.93%
TF_IDF_G_0.3	0.3007	+2.03%	0.4041	+1.02%	0.3714	+2.22%
TF_IDF_G_0.4	0.2976	+0.98%	0.4041	+1.02%	0.3735	+2.80%
TF_IDF_G_0.5	0.2953	+0.20%	0.4041	+1.02%	0.3714	+2.22%
TF_IDF_G_0.6	0.2951	+0.13%	0.4041	+1.02%	0.3673	+1.10%
TF_IDF_G_0.7	0.2947	0%	0.4041	+1.02%	0.3694	+1.67%
TF_IDF_G_0.8	0.2941	-0.20%	0.4041	+1.02%	0.3694	+1.67%
TF_IDF_G_0.9	0.2941	-0.20%	0.4041	+1.02%	0.3694	+1.67%
TF_IDF_G_1	0.2941	-0.20%	0.4041	+1.02%	0.3694	+1.67%

Tableau III. 1: Résultats de l'évaluation de notre approche par rapport au modèle TF_IDF

Modèles de recherche	MAP	Taux d'amélioration de la MAP	P@5	Taux d'amélioration de la P@5	P@10	Taux d'amélioration de la P@10
BM25	0.2937	_____	0.3878	_____	0.3612	_____
BM25_C	0.2926	-0.37%	0.3918	+1.03%	0.3673	+1.68%
BM25_T	0.2953	+0.54%	0.3918	+1.03%	0.3653	+1.13%
BM25_G_0.1	**0.2974**	**+1.25%**	0.3959	+2.08%	**0.3714**	**+2.82%**
BM25_G_0.2	**0.2974**	**+1.25%**	0.3918	+1.03%	0.3673	+1.68%
BM25_G_0.3	0.2959	+0.74%	**0.4000**	**+3.14%**	0.3612	0%
BM25_G_0.4	0.2951	+0.51%	0.3918	+1.03%	0.3612	0%
BM25_G_0.5	0.2937	0%	0.3918	+1.03%	0.3673	+1.68%
BM25_G_0.6	0.2934	-0.10%	0.3918	+1.03%	0.3673	+1.68%
BM25_G_0.7	0.2932	-0.13%	0.3918	+1.03%	0.3673	+1.68%
BM25_G_0.8	0.2927	-0.34%	0.3918	+1.03%	0.3673	+1.68%
BM25_G_0.9	0.2937	0%	0.3918	+1.03%	0.3673	+1.68%
BM25_G_1	0.2937	0%	0.3918	+1.03%	0.3673	+1.68%

Tableau III. 2: Résultats de l'évaluation de notre approche par rapport au modèle BM25

D'après les tableaux ci-dessus, nous pouvons tirer les remarques et conclusions suivantes :

Notre approche améliore les résultats des modèles de base en termes de Map et de précision à N document, selon les deux fonctions Triangle, Gaussienne, contrairement à la fonction Cercle qui a obtenue des améliorations uniquement en terme de précision à N documents.

Nous avons obtenu une meilleure amélioration de précision avec la fonction gaussienne au niveau de deux modèles : TF_IDF_G_0.2 affiche des améliorations de l'ordre de **+2.34% (0.3016)**, BM25_G_0.1 et BM25_G_0.2 affiche des améliorations de l'ordre **+1.25% (0.2974)**.

III.5.3.2 Résultats requête-par-requête

Afin d'avoir une vision plus fine et détaillée des améliorations obtenues par notre approche, nous avons effectué une analyse requête-par-requête entre notre approche et les deux modèles de base TF_IDF et BM25.

Les graphiques ci-dessous montrent les différences entre les averages précision du modèle TF_IDF_X et celles du modèle de base TF_IDF, et entre les averages précision du modèle $BM25_X$ et celles du modèle de base BM25.

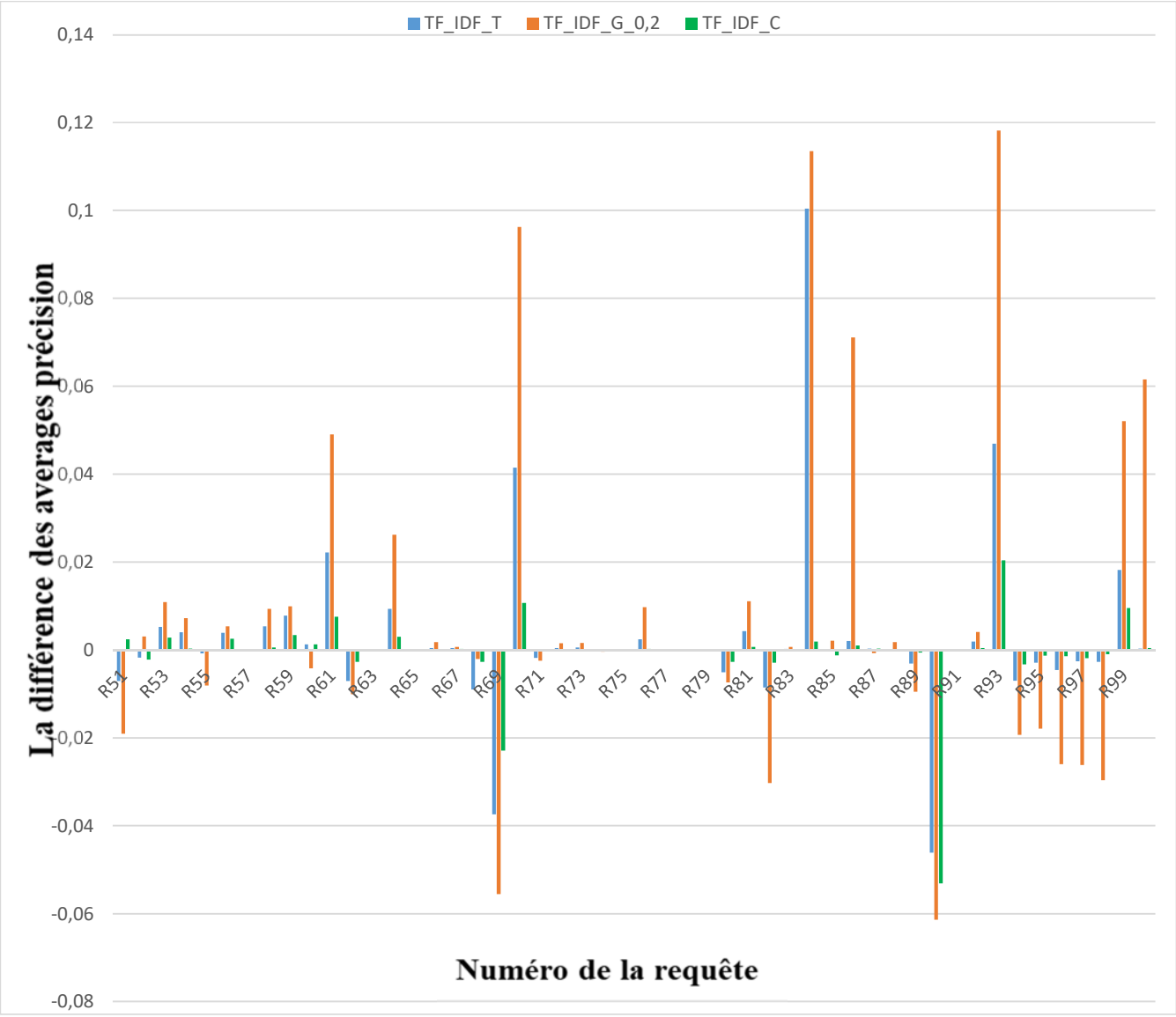


Figure III. 5: Analyse requête-par-requête entre les modèles TF_IDF et TF_IDF_X

À partir des résultats obtenus nous avons noté les remarques suivantes :

- TF_IDF_T donne de meilleurs résultats par rapport au modèle de base TF_IDF dans **23** requêtes. Ce dernier modèle outrepasse le modèle TF_IDF_T dans **18** requêtes. Les deux modèles présentent les mêmes résultats sur **9** requêtes.
- TF_IDF_G_0.2 donne de meilleurs résultats par rapport au modèle de base TF_IDF dans **24** requêtes. Ce dernier modèle outrepasse le modèle TF_IDF_G_0.2 dans **19** requêtes. Les deux modèles présentent les mêmes résultats sur **7** requêtes.
- TF_IDF_C donne de meilleurs résultats par rapport au modèle de base TF_IDF dans **23** requêtes. Ce dernier modèle outrepasse le modèle TF_IDF_C dans **16** requêtes. Les deux modèles présentent les mêmes résultats sur **11** requêtes.

L'analyse de l'ensemble des remarques ci-dessus montrent que les trois fonctions contribuent à l'amélioration des requêtes. Néanmoins, la deuxième fonction (la fonction gaussienne où $\boldsymbol{\lambda = 0.2}$) offre une meilleure amélioration par rapport aux autres fonctions.

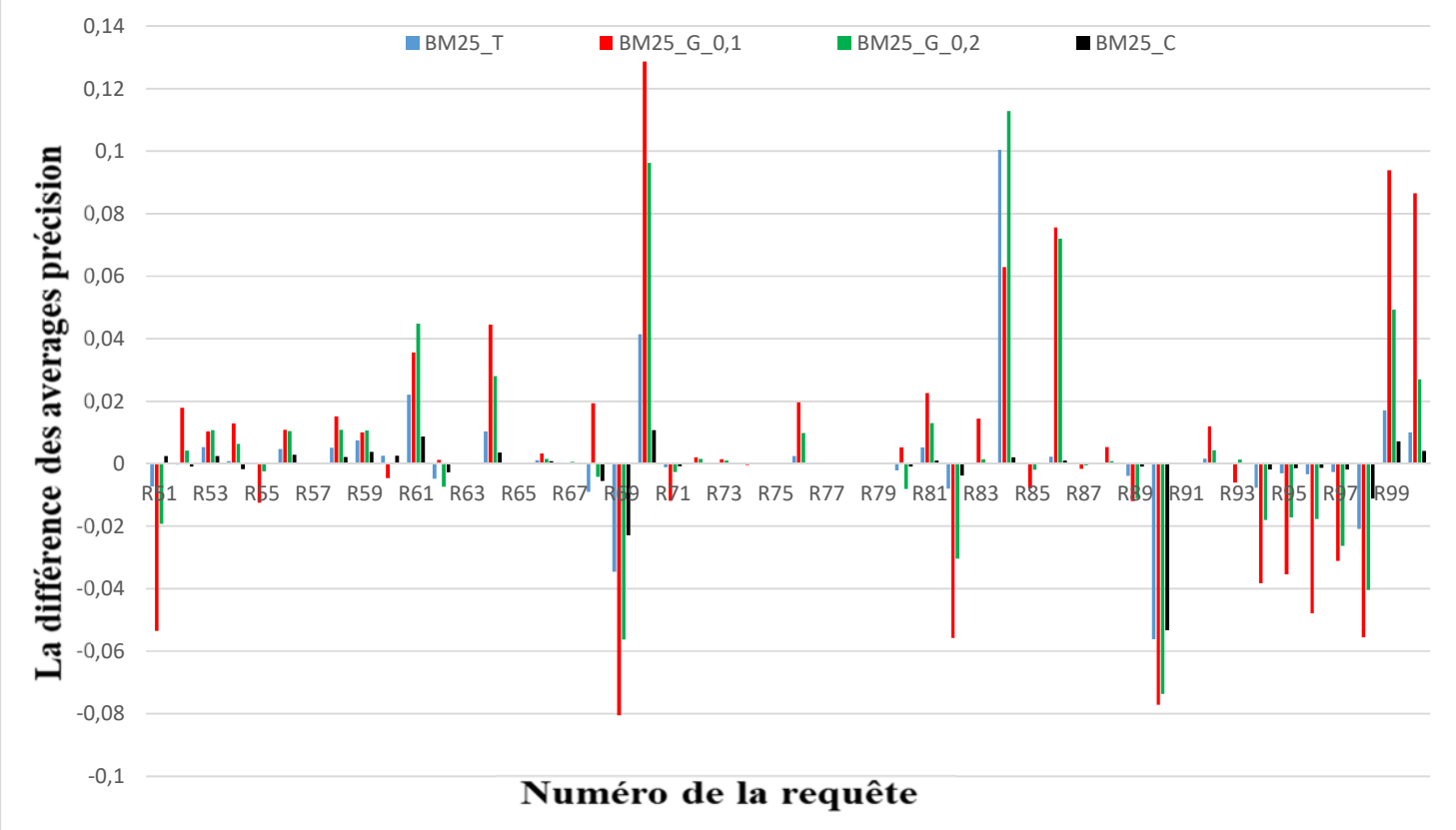


Figure III. 6: Analyse requête-par-requête entre les modèles $BM25$ et $BM25_X$

À partir des résultats obtenus nous notons les points suivants :

- Le modèle BM25_T améliore les résultats du modèle de base BM25 sur **23** requêtes. Ce dernier modèle outrepasse le modèle BM25_T dans **18** requêtes. Enfin, les deux modèles réalisent les mêmes performances sur **9** requêtes.
- BM25_G_0.1 donne de meilleurs résultats par rapport au modèle de base BM25 dans **25** requêtes. Ce dernier modèle outrepasse le modèle BM25_G_0.1 dans **18** requêtes. Les deux modèles présentent les mêmes résultats sur **7** requêtes.
- BM25_G_0.2 donne de meilleurs résultats par rapport au modèle de base BM25 dans **23** requêtes. Ce dernier modèle outrepasse le modèle BM25_G_0.1 dans **19** requêtes. Les deux modèles présentent les mêmes résultats sur **8** requêtes.
- Le modèle BM25_C améliore les résultats du modèle de base BM25 sur **21** requêtes. Ce dernier modèle outrepasse le modèle BM25_C dans **20** requêtes. Les deux modèles donnent des résultats équivalents sur **9** requêtes.

L'analyse de l'ensemble des remarques ci-dessus montrent que les trois fonctions contribuent à l'amélioration des requêtes. Néanmoins, la deuxième fonction (la fonction gaussienne où $\boldsymbol{\lambda = 0.1}$) offre une meilleure amélioration par rapport aux autres fonctions.

III.6 Conclusion

Dans ce chapitre, nous avons décrit une extension des modèles de base : TF_IDF et BM25 avec les positions du terme. L'idée principale est de surpondérer les termes qui apparaissent au début du document. Nous avons proposé une technique pour mettre en œuvre cette idée. Cette technique prend en compte toutes les positions du terme dans les documents de la collection, elle est formalisée en tant que score position, qui est ensuite intégré dans les deux modèles de base.

Les résultats obtenus sur la collection de test TREC AP88 ont montré des améliorations par rapport aux deux modèles de base : TF_IDF et BM25. Cela montre que le facteur position du terme est utile pour la recherche d'information. Nous avons constaté également que le modèle basé sur la fonction gaussienne affiche les meilleurs résultats par rapport aux autres fonctions.

Conclusion générale

Conclusion générale

Notre travail présenté dans le cadre de ce mémoire s'insère dans le domaine de la recherche d'information. Il porte sur l'extension de deux modèles de base : TF-IDF et BM25 par un facteur basé sur les positions du terme dans les documents de la collection où il apparait.

Pour mener à terme notre travail, nous avons donné un aperçu général sur la recherche d'information ainsi que le système de recherche d'information. Nous avons ensuite présenté les facteurs de pondérations à savoir les facteurs de pondération classiques et supplémentaires utilisés pour réévaluer le poids d'un terme dans un document.

Pour mettre en œuvre notre approche « Estimation de score dans les modèles de recherche d'information RI » nous avons utilisé la plateforme Terrier, le langage de programmation Java et l'environnement NetBeans.

L'approche proposée a montré des améliorations par rapport aux deux modèles de base (TF_IDF et BM25). Cela montre que le facteur position du terme est utile pour la recherche d'informations.

Ce travail nous a permis d'approfondir nos connaissances sur la recherche d'information, de mettre l'accent sur la manière dont les systèmes de recherche d'information fonctionnent dans la plate-forme Terrier.

Dans le futur, nous prévoyons d'explorer différents points, comme l'utilisation d'autres collections de tests pour évaluer ce nouveau facteur: le poids de la position du terme dans tous les documents de la collection. Ainsi que, l'intégration de ce nouveau facteur dans d'autres modèles de recherche d'information par exemple le modèle de langue.

Bibliographie

Bibliographie

[1] : Jian-Yun, N. « Le domaine de recherche d'information ». Cours à l'Université de Montréal.

[2] : Amirouche, F. « La recherche d'information ». Cours Master2 ingénierie des systèmes d'informations. Université Mouloud Mammeri Tizi-Ouzou, 2019/2020.

[3] : Salton, G. McGill. « Introduction to Modern Information Retrieval ». Article,1983.

[4] : Ben Aouicha, M. « Une approche algébrique pour la recherche d'information structurée ». Thèse de doctorat à l'université Paul Sabatier,2009.

[5]: Christopher D, M. Prabhakar, R. Hinrich, S. « Introduction to information retrieval » Cambridge University Press, 2008.

[6] : EL CHARIF, R. « Analyse des paramétres de pondération dans le cadre collections volumineuses ». DEA d'informatique, 2006.

[7] : Sauvagnat**,** K. « Modèle flexible pour la Recherche d'Information dans des corpus de documents semi-structurés ». Thèse de doctorat à l'université de Toulouse, 2005.

[8] : Salton, G. « A comparison between manual and automatic indexing methods ». Journal of American Documentation, 1971. pages 61–71.

[9]: Salton, G. «The SMART retrieval system : Experiments in automatic document ». Article, 1970.

[10]: Jones, K. Walker, S. & Robertson, S. « probabilistic model of information retrieval : development and comparative experiments ». Information Processing & Management, Article, 2000.

[11]: Maron, M. Kuhns, J. « On relevance, probabilistic indexing and information retrieval ». Journal of the ACM (JACM), 1960. pages 216–244.

[12]: Robertson, S.E. « The probability ranking principle in IR ». Journal of documentation, 1977. pages 294–304.

[13] : Hammache, A. Recherche d'Information. « Un modèle de langue combinant mots simples et mots composés ». Thèse de doctorat à l'Université Mouloud Mammeri TiziOuzou, 2013.

Bibliographie

[14] : Abbas, N. « Vers une Extension Sémantique de l'Analyse Formelle de Concepts : Application à la Recherche d'informations ». Mémoire de magister, Université Mouloud Mammeri de Tizi-Ouzou, 2014.

[15]: Shaw JR, W. Burgin, R. Hawell, P. « performance standards and evaluations in ir test collection: Cluster-based retrieval models ».Article, 1997.

[16]: Scholer, F. Kelly, D. Carterette, B. « Information retrieval evaluation using test collections ». 2016

[17] : Bouramoul, A. « Recherche d'informations contextuelle et sémantique sur le web ». Thèse de doctorat à l'Université MENTOURI de Constantine, 2011.

[18]: Hofman, T. « Probabilistic latent semantic indexing ». Proceedings of SIGIR '99, Berkeley, CA, USA, 1999.

[19]: Liu, X. and Croft, W. B. « Cluster-based retrieval using language models ». In Proceedings of SIGIR '04, 2004. page 186-193.

[20]: Wei, X., & Croft, W. B. « Lda-based document models for ad-hoc retrieval ». In Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval, 2006. page 178-185.

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

[22]: Song, S. & Myaeng, S. « A Novel Term Weighting Scheme Based on Discrimination Power Obtained from Past Retrieval Results ». Information Processing Manegement,48(5), 2012. page 919-930.

[23]: Zhang, Y. Rahman, M. Braylan, A. & Dany, B. « Neural Information Retrieval: A Literature Review » université du Texas à Austin,2017.

[24]: Robertson, S.E. Zaragoza, H. & Taylor, M. (2004). « Simple bm25 extension to multiple weighted fields ». In Conference on Information and Knowledge Management, 2004, Washington, DC, USA.

Bibliographie

[25]: Zamani, H. Mitra, B. Song, X. Craswell, N. & Tiwary, S. (2018). « Neural Ranking Models with Multiple Document Fields ». In Proceedings of the 11th ACM International Conference on Web Search and Data Mining, 2018, Los Angeles, USA.

[26]: Hammache, A. Boughanem, M. & Ahmed-Ouamer, R. (2014). « Combining compound and single terms under language model framework ». Knowl. Inf. Syst, Vol 39, n° 2, page 329-349.

[27]: Oglivie, P. Callan, J. (2003). « Combining document representations for known-item search ». In ACM International Conference on Research and Development in Information Retrieval, 2003, Toronto, Canada.

[28]: Lv, Y. Zhai, C. (2009). « Positional language models for information retrieval ». In ACM International Conference on Research and Development in Information Retrieval, 2009, Boston, Massachusetts.

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

[30]: Troy, AD. Zhang, GQ. (2007). « Enhancing Relevance Scoring with Chronological Term Rank ». In ACM International Conference on Research and Development in Information Retrieval, 2007, Amsterdam, Pays-Bas.

[31]: Ounis, I. Amati, G. Plachouras, V. He, B, Macdonald, C. Lioma, C. « Terrier: A High Performance and Scalable Information Retrieval Platform », Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006), 2006

[32] : Gauthier, P. Laurent, V. « Initiation à la programmation orienté objet avec le langage java ». Cours, 2013.

[33] : Bernard, D. « Netbeans/PHP ». Cours, 2010.