République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université Ahmed Draia - Adrar

Faculté des Sciences et de la Technologie

Département des Mathématiques et Informatique

Mémoire de fin d'étude, en vue de l'obtention du diplôme de Master en informatique

Option : Réseaux et Systèmes Intelligents

Thème

Outil de détection de plagiat dans un document

Préparé par

Rima ROUIBIA et Imane BELHADJ

Président : Mr. OMARI Mohammed

Examinateur : Mr. CHOGUEUR Djilali

Examinateur : Mr. DEMRI Mohammed

Encadreur : Mr. CHERAGUI Mohamed Amine

Année Universitaire 2015/2016

Remerciements

Il est de coutume de dire qu'un mémoire n'est pas le fruit du seul travail de son auteur, mais le résultat de nombreuses et étroites collaborations; celle-ci ne déroge pas à la règle. Nous remercions avant tout le Bon Dieu de nous avoir donné la volonté de finir ce mémoire. Ce travail a pu voir le jour avec énormément d'aide et encouragement des personnes autour de nous. Ce court remerciement ne sera pas suffisant pour récompenser leurs efforts mais tout de même ... A l'issue de deux agréables années au sein de département mathématique et informatique de l'université d'ADRAR nous tenons à remercier l'ensemble des enseignants pour leur dévouement, toutes nos pensées de gratitude se dirige vers notre encadreur Mr.CHERAGUI Mohamed Amine, qui est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'il a bien voulu nous consacrer. Nous tenant aussi à remercier les membres du jury qui ont accepté d'examiner notre mémoire. Nous adressons nos plus sincères remerciements à tous nos collègues et nos amis qui partagent avec nous les bons moments de l'étude pendant les deux années. Nous exprimons nos gratitude à tous nos proches qui nous ont toujours soutenue et encouragée au cours de la réalisation de ce mémoire. Enfin, tous ceux qui ont Contribués, de près ou de loin à la réalisation de cet mémoire et que nous ne pouvons malheureusement citer, trouvent ici l'expression de notre profonde gratitude.

Dédicace

Je dédie ce modeste travail :

À l'homme de ma vie, mon exemple éternel, mon soutien moral et source de joie et de bonheur, école de mon enfance, qui a été mon ombre durant toutes les années des études, et qui a veillé tout au long de ma vie à m'encourager, à me donner l'aide et à me protéger

À toi mon père.

À celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, à la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma vie et mon bonheur

À toi maman que j'adore

Que dieu les gardes et les protège.

A mes chers frères et sœurs, à mes adorable neveux et nièces, En témoignage de mon affection fraternelle, de ma profonde tendresse, je vous souhaite que Dieu, le tout puissant, vous protège et vous garde.

Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagnaient durant mon chemin d'études supérieures .

mon collègue « Imane » qui a partagé avec moi cette recherche.

Melle .***Rima ROUIBIA***

Dédicace

Je dédie ce modeste travail

A mes parents pour leur amour inestimable, leur confiance, leur soutien, leurs sacrifices et toutes les valeurs qu'ils ont su m'enseigner.

Pour ceux qui aime à ma réussite et j'espère qu'ils glorifiaient

Pour tous mes frères et sœurs et mes neveux et mes niéces

Pour ceux et mes chers amis et mon collègue « Rima » qui ont partagé avec moi troubler cette recherche.

Melle. ***Imane BELHADJ***

Résumé

Avec l'avènement de la transcription orale vers du texte et le perfectionnement de la reconnaissance de l'écriture, la production humaine en information textuelle s'est exponentiellement étendue. Plusieurs voix de recherches on été menée pour un accès précis et automatisé à l'information avec comme contrainte l'exploitation optimale de ces volumineuses bases d'information.

Ce travail s'inscrit dans le cadre général de la recherche d'information. Dans un objectif de réalisation d'un outil de détection de plagiat dans un ensemble de documents.

Le développement de notre outil est passé par deux phases complémentaires, qui sont: la phase d'indexation et la phase de recherche. Cette dernière permet de sélectionner, dans une collection de documents préalablement enregistrée, les informations (documents) pertinentes répondant à un besoin en information. Ce besoin est formellement exprimé par un utilisateur sous forme de requête. Pour accomplir cette tâche nous avons proposé trois techniques fondamentales, à savoir :

- Technique basé sur le modèle vectoriel.
- Technique basé sur les opérateurs logiques.
- Technique basé sur la couverture (requête/document).

Nous avons adopté un dictionnaire pour préserver les informations utiles, ce dictionnaire est sous forme d'une base de données qui contient une seule table. Cette dernière contient toutes les informations concernent la collection des documents.

Mots clés : Recherche d'Information, Requête, Corpus, Modèles de Recherche, Plagiat.

Abstract

With the appearance of the transformation of the oral transcription into the written form and exponentially broad. So many ways of searching have been brought for exact and automatic access to the information although the pressure of optimal operating of large data base.

This work is included within the information retrieval scope. Its aim is to create a plagiarism detection tool consisting of a collection of documents.

The development of our tool has passed through two complimentary phases, which are: the indexing phase and the search phase. The latter allows us select, within an already saved collection of documents, pertinent information (documents) providing the information we need. This need is formally expressed in a form of a request by the user. To accomplish this process we have suggested three principle techniques, which are:

- A technique based on the vector model.
- A technique based on logic operators.
- A technique based on the covering.

We have adopted a dictionary to preserve useful information, this dictionary is in the form of a database that contains one single table. The latter contains all the information concerning the collection of documents.

Key words: information retrieval, Request, Corpus, Search models, Plagiarism.

ملخص

مع اكتشاف إمكانية تحويل كلام شفهي الى نص والتحكم الجيد في الكتابة أصبح الإنتاج البشري في مجال المعلومات النصية جد متطور, الكثير من البحوث قد تم انجازها من اجل دقة وأوتو ماتيكية الوصول الى المعلومة مع وجود صعوبة في الاستغلال الأمثل لهذه القواعد المعلوماتية.

هذا العمل أنجز في الإطار العام للبحث عن المعلومات .والهدف هو انجاز وسيلة للكشف عن السرقة العلمية في مجموعة من الوثائق.

تطور هذه الوسيلة قد مر بمرحلتين متكاملتين: مرحلة الفهرسة ومرحلة البحث.هذه الأخيرة تسمح باختيار معلومات مهمة في مجموعة من الوثائق المسجلة مسبقا والتي تلبي الحاجة الى المعلومات. هذه الحاجة معبر عنها من طرف مستخدم عن طريق طلب عملية بحث.

ومن اجل انجاز هذا العمل اقترحنا ثلاث تقنيات أساسية :

- تقنية ترتكز على النموذج الشعاعي.
- تقنية ترتكز على العوامل المنطقية.
- تقنية ترتكز على التغطية (طلب عملية بحث \وثيقة).

اعتمدنا على القاموس من أجل الحفاظ على المعلومات المفيدة، هذا القاموس هو في شكل قاعدة بيانات تحتوي على جدول واحد. وهذا الأخير يحتوي على كافة المعلومات المتعلقة بالوثائق.

الكلمات المفتاحية : البحث عن المعلومات,طلب عملية بحث,عينة, نموذج البحث, السرقة العلمية.

Tables des Matières

Liste des figures

Liste des tableaux

Introduction Générale

1. Contexte et But du mémoire

Avec l'augmentation rapide du volume documentaire stocké sous format numérique, et l'avènement du Web, la quantité d'informations disponible ne cesse de croître 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 de données.

Le phénomène de triche par plagiat est un problème qui s'est rapidement développé au cours de ces dernières années. Au lieu de produire un travail original, certains chercheurs préfèrent copier directement le contenu trouvé dans des livres, des articles de journaux ou des travaux antérieurs rédigés par d'autres personnes.

L'accès facile et rapide à l'information sur le web a mis en péril le droit d'auteur. Pour luter contre ce phénomène, appelé également plagiat, plusieurs techniques ont été mises en place pour détecter un éventuel plagiat. Les solutions les plus simples consistent à chercher des mots ou des phrases clés du texte en question, afin de voir si l'on retrouve un texte potentiellement plagié.

L'objectif principal de nos travaux est de concevoir et développer un outil basé sur un ensemble de technique de recherche d'information pour aider les enseignants à détecter les documents plagiés dans un corpus.

2. Plan de travail

Ce mémoire est organisé en quatre (04) chapitres principaux, comme suit :

- Le premier chapitre présente les concepts de base de la RI. Nous commençons par donner une définition de la RI et nous illustrons également le processus de recherche d'information en présentant les étapes d'indexation, la recherche et de mise en correspondance, ainsi que les techniques de reformulation des requêtes.
- Le deuxième chapitre décrit les différents modèles servant de cadre théorique pour la modélisation du processus de RI.
- Le troisième chapitre représente l'objectif de notre projet, nous donnons une description détaillée de notre outil « PlagZoom » pour détection de plagiat.
- le quatrième chapitre présente notre outil « PlagZoom » par le biais de captures d'écrans, et quelques exemples des testes et résultats, ainsi que l'évaluation de notre outil.

Et enfin, nous terminons par une conclusion générale, en évoquant de nouvelles perspectives de recherche.

Chapitre I

La Recherche d'information

1. Introduction

La recherche d'information est un domaine historiquement lié aux sciences de l'information et à la bibliothéconomie qui consiste à chercher sur une grande masse d'informations les documents qui satisfont les besoins d'utilisateur. Hors la quantité d'information stockée au format électronique ne cessant de croître, il devient de plus en plus difficile de retrouver un ensemble d'information contenu dans un document, au sein d'une base de documents, appelée corpus. De plus, l'information disséminée dans un document n'est pas structurée et donc difficilement accessible voire identifiable. Outre le problème d'identifier l'information contenue dans un document, la recherche d'information doit également permettre à l'utilisateur de formuler sa demande, son besoin d'information, le plus exactement possible, sous la forme d'une requête normalement en langage naturel.

Ce chapitre est organisé en deux parties : la première présente les concepts de base de la RI[1]. La deuxième partie décrit le processus de RI, à savoir les étapes d'indexation, la recherche et de mise en correspondance, ainsi que les techniques de reformulation des requêtes.

2. Bref historique sur la recherche d'information

La recherche d'information n'est pas un domaine récent :

- **1940 :** Avec la naissance des ordinateurs, la RI se concentrait sur les applications dans des bibliothèques. Depuis le début de ces études, la notion de pertinence a toujours été un objet **[2]**.
- **1950 :** Début de petites expérimentations en utilisant des petites collections de documents (références bibliographiques). Le modèle utilisé est le modèle booléen **[2]**.
- **1960 :** Expérimentations plus larges ont été menées. On a développé une méthodologie d'évaluation du système qui est aussi utilisée maintenant dans d'autres domaines (des corpus de test ont été conçus pour évaluer des systèmes différents) **[2]**.
- **1970 :** Développement du système SMART. Les travaux sur ce système ont été dirigés par G. Salton **[2]**. Certaines nouvelles techniques ont été implantées et expérimentées pour la première fois dans ce système (par exemple, le modèle vectoriel et la technique de relevance feedback).Il y a aussi de développements sur le modèle probabiliste **[1]**.
- **1980 :** Les travaux sur la RI ont été influencés par l'avènement de l'intelligence artificielle. Ainsi, on tentait d'intégrer des techniques de l'IA en RI, par exemple, système expert pour la RI.

1990 : Internet à propulser la RI en avant-scène avec beaucoup d'applications. La venue d'internet a aussi modifié la RI. La problématique est élargie. Par exemple, on traite maintenant plus souvent des documents multimédia qu'avant. Cependant, les techniques de base utilisées dans les moteurs de recherche sur le web restent identiques **[1] [2]**.

[1] Est un acronyme de la recherche d'information

3. Définitions de la recherche d'information

Il existe plusieurs définitions pour la recherche d'information, qui sont plus au moins proches :

Définition 1 : la recherche d'information est un domaine qui étudie la structure, l'analyse, l'organisation, le stockage, la recherche et la récupération d'informations **[3]**.

Définition 2 : La RI consiste à restituer les documents qui peuvent être pertinents par rapport au besoin d'information exprimé dans la requête **[4]**.

Définition 3 : La recherche d'information est une discipline de recherche qui intègre des modèles et des techniques dont le but est de faciliter l'accès à l'information pertinente pour un utilisateur ayant un besoin en information **[5]**.

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

4. Objectifs de la recherche d'information

La recherche d'information a pour objectifs **[2]**:

- Identifier en vue d'exploiter de l'information contenue dans des documents et des bases de données (son, texte, image) par rapport à une requête formulée par un utilisateur.
- Le SRI[2] devra nous retourner le moins possible de documents non pertinents
- Les contenus des documents peuvent être non structurés ou semi structurés.

5. Concepts de base de la recherche d'information

La recherche d'information est considérée comme l'ensemble des techniques permettant de sélectionner à partir d'une collection de documents, ceux qui sont susceptibles de répondre aux besoins de l'utilisateur. Plusieurs concepts clés s'articulent autour de la notion de la Recherche d'Information :

5.1. Document

Le document est constitué par un texte, un morceau de texte, une image, une bande de vidéo…etc., qui peut être retourné en réponse à une requête/ besoin en information d'utilisateur **[8]**.

5.2. Collection de documents

La collection de documents constitue l'ensemble des informations exploitables et accessibles. Elle est constituée d'un ensemble de documents. Dans le cas général et pour un souci d'optimalité, la base constitue des représentations simplifiées mais suffisantes pour ces documents. Ces représentations sont étudiées de telles sortes que la gestion (ajout suppression d'un document) ou l'interrogation (recherche) de la base se font dans les meilleures conditions de coût **[7]**.

[2] Est un acronyme de Système de recherche d'information

5.3. Besoin d'information

La notion de besoin en information en recherche d'informations est souvent assimilée au besoin de l'utilisateur. Trois types de besoin utilisateur ont été définis par **[6]** :

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

5.4. Requête

La requête constitue l'expression du besoin en information de l'utilisateur. Elle représente l'interface entre le SRI et l'utilisateur. Une requête est un ensemble de mots clés, mais elle peut être exprimée en langage naturel, booléen ou graphique **[7]**.

5.5. Pertinence

La notion de pertinence est un critère principal pour l'évaluation des systèmes de recherche d'information **[8]**.Cependant, la définition de cette notion complexe n'est pas simple, car elle fait intervenir plusieurs notions. basiquement elle peut être définie comme la correspondant entre un document et une requête ou encours comme une mesure d'informativité du document à la requête. Essentiellement; deux types de pertinence sont définis: la pertinence système et la pertinence utilisateur.

- ✓ **Pertinence système:** qui définit la capacité du système à comparer entre documents et requêtes et à quel point il a réussi à retrouver les documents adéquats à cette requête. Pratiquement, la pertinence système se traduit par un score de pertinence basé sur le degré de similarité entre un document et une requête donnée **[9]**.

- ✓ **Pertinence utilisateur:** quant à elle, se traduit par les jugements de pertinence utilisateur sur les documents fournis par le système de recherche d'information en réponse à une requete.la pertinence utilisateur est subjective; car pour un même document retourné en réponse à une même requête, il peut être jugé différent par deux utilisateurs distincts (qui ont des centres d'intérêt différent).de plus, cette pertinence est évolutive, un document jugé non pertinent à l'instant «t» pour une requête peut être jugé pertinent à l'instant «t+1», car la connaissance de l'utilisateur sur le sujet à évolué **[9]**.

L'objectif de tout système de recherche d'information est de rapprocher la pertinence système de la pertinence utilisateur.

5.6. Système de Recherche d'Information (SRI)

Est un système informatique constitué d'un ensemble de programmes, dont l'objectif principal est de sélectionner, dans une collection de documents préalablement enregistrée, les informations (documents) pertinentes répondant à un besoin en information formellement exprimé par un utilisateur sous forme de requête **[21]**. **La Figure 01** illustre ce fonctionnement.

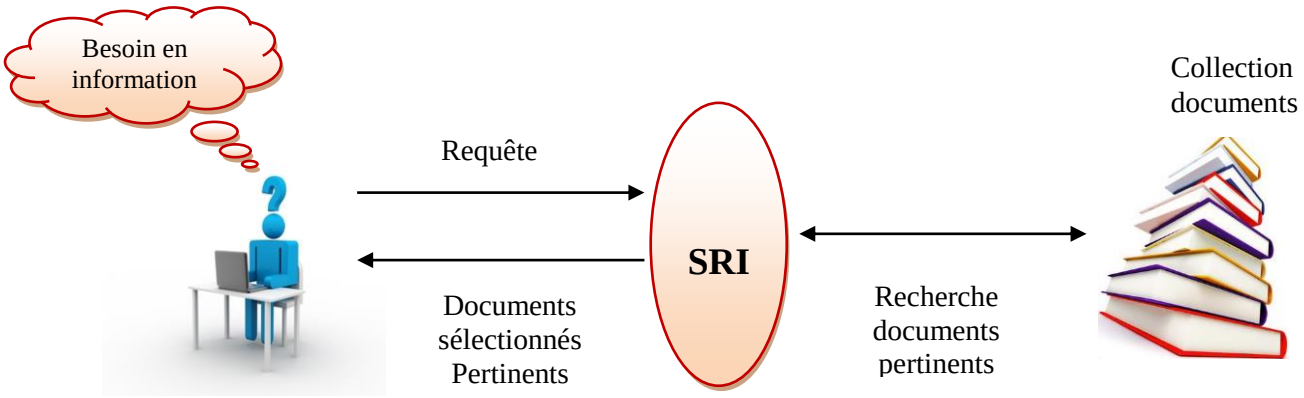


Figure 01 : SRI en réponse à une Requête utilisateur

6. Processus de recherche d'information

Pour répondre aux besoins en information de l'utilisateur, un SRI met en oeuvre un certain nombre de processus pour réaliser la mise en correspondance des informations contenues dans un fond documentaire d'une part, et des besoins en information des utilisateurs d'autre part. Ces processus supposent que la collection de documents est unique **[10]**.

Un système de recherche d'information possède trois fonctions principales : l'indexation, la recherche et la reformulation de la requête, représentées schématiquement par le processus de recherche d'information.

La **Figure 02** illustre l'architecture générale d'un système de recherche d'information.

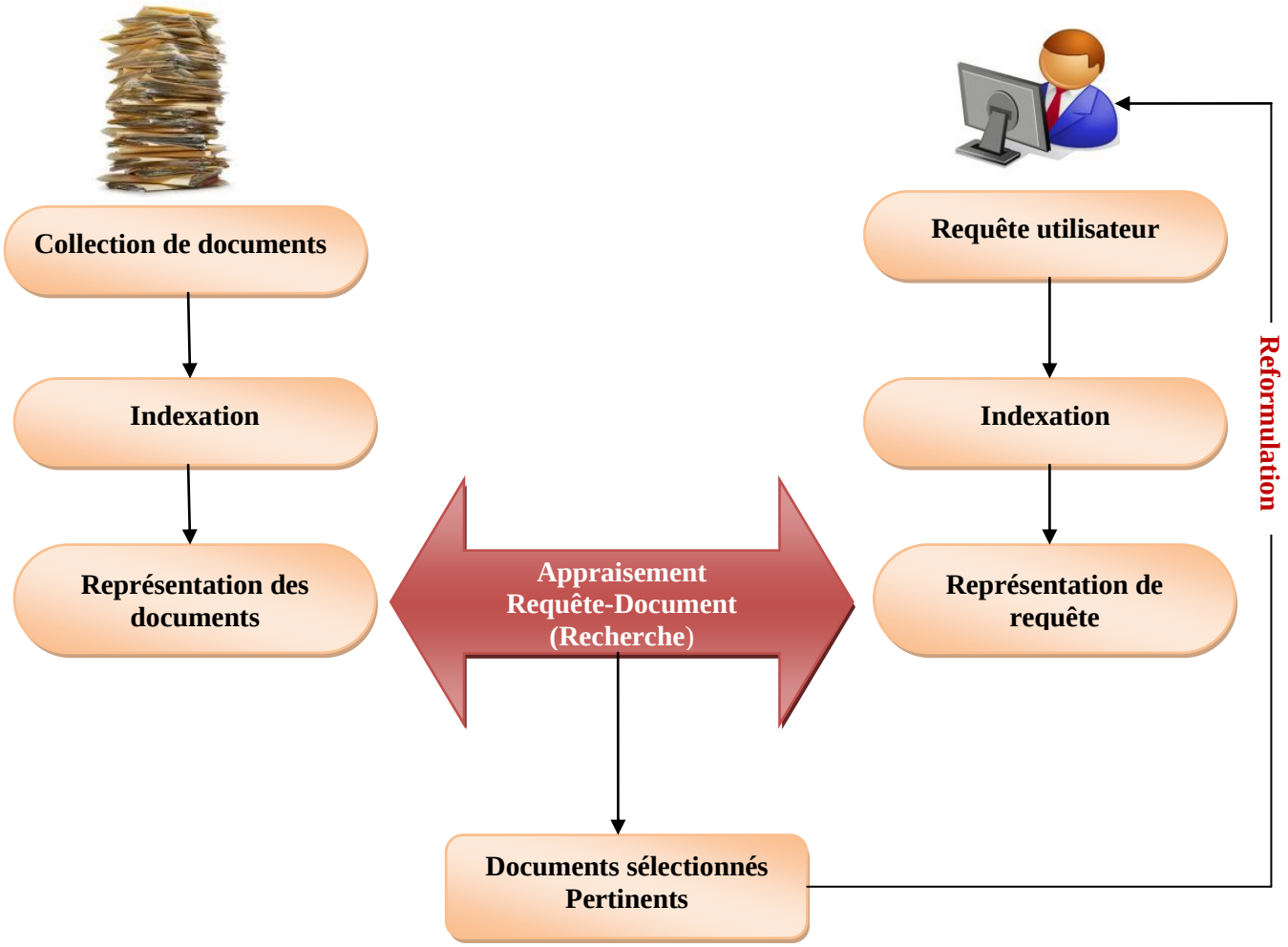


Figure 02: Processus de système de recherche d'information(SRI)

6.1. Le processus d'indexation

L'indexation consiste à analyser chaque document de la collection afin de créer un ensemble de mots-clés. Ces mots-clés seront plus facilement exploitables par le système lors du processus ultérieur de recherche. L'indexation permet ainsi de créer une représentation des documents dans le système. Son objectif est de trouver les concepts les plus importants du document (ou de la requête), qui formeront le descripteur du document **[13]**.

L'indexation peut se faire de trois (3) manières différentes : manuellement, semi-automatique, ou de manière automatique.

6.1.1. Indexation manuelle

Ce genre d'indexation est guidé par un spécialiste du domaine ou par un documentaliste. Après la lecture des documents, ce spécialiste détermine selon ses connaissances, les mots-clés qui lui semblent les plus adéquats pour représenter le contenu du document. Ce mode d'indexation est fondé sur le jugement

humain. Il se caractérise par sa profondeur, sa cohérence et sa qualité. Cependant, il dépend de l'indexeur ce qui induit la subjectivité de ses résultats. De plus l'augmentation du nombre de documents à indexer rend la tâche d'indexation manuelle difficile et coûteuse en temps. L'indexation automatique permet de pallier à ce problème **[11]**.

6.1.2. L'indexation semi-automatique

Elle consiste en un premier temps à indexer automatiquement les documents en s'appuyant sur un vocabulaire contrôlé comme un thésaurus[3] ou n'importe quelle base terminologique. Ce type d'indexation requiert le contrôle manuel du processus par un spécialiste du domaine afin de valider le résultat obtenu et pour établir des relations sémantiques entre les termes d'indexation **[12]**.

6.1.3. L'indexation Automatique

Le processus d'indexation est entièrement informatisé, elle regroupe un ensemble de traitements automatisés sur un document. On distingue : l'extraction automatique des mots des documents, l'élimination des mots vides, la lemmatisation (radicalisation ou normalisation), le repérage de groupes de mots, la pondération des mots et enfin la création de l'index **[13]**.

Nous détaillons ces différentes étapes ci-dessous.

6.1.3.1. Analyse lexicale

L'étape de l'analyse lexicale permet d'extraire l'ensemble des termes appartenant à un document. Cette extraction est effectuée en tenant compte des espaces de séparation entre mots, des chiffres et des ponctuations. Un terme peut être un mot simple (pomme) ou un mot composé (pomme de terre) mais en RI on utilise souvent les mots simples **[14]**.

6.1.3.2. Élimination des mots vides

Un des problèmes majeurs de l'indexation consiste à extraire les termes significatifs et à éviter les mots vides (pronoms personnels, prépositions,...etc.).

Les mots vides peuvent aussi être des mots athématiques (les mots qui peuvent se retrouver dans n'importe quel document parce qu'ils exposent le sujet mais ne le traitent pas, comme par exemple contenir, appartenir, …etc.) **[15].** On distingue deux techniques pour éliminer les mots vides :

- ✓ L'utilisation d'une liste de mots vides (stop words).
- ✓ L'élimination des mots dépassant un certain nombre d'occurrences dans la collection.

[3] Un thésaurus est un vocabulaire d'un langage d'indexation contrôlé, organisé formellement de façon à expliciter les relations à priori entre les notions

6.1.3.3. Lemmatisation

Un mot peut avoir plusieurs formes dans un texte dont le sens est presque similaire. La lemmatisation est une technique qui permet de ramener un mot à sa racine. Par exemple, programmes et programme, programmer et programmation, programmeurs et programmées font tous références à la racine 'programme'. Elle désigne l'analyse lexicale du contenu textuel regroupant les mots d'une même famille afin de réduire les mots à leurs racines grammaticales **[16]**.

6.1.3.4. Pondération

La pondération permet d'attribué un poids au terme d'indexation qui représente l'importance de cet index dans le document respectivement dans la requête et de réduire la taille de l'ensemble des descripteurs de document et des requêtes (nombre d'index). La plupart des techniques de pondération sont basées sur les facteurs TF et IDF **[17]** :

- **TF (Term Frequency) :** mesure l'importance d'un terme dans un document. Cette mesure est souvent en fonction de la fréquence d'un terme dans un document ou une requête. Elle peut être utilisée telle quelle ou selon plusieurs déclinaisons (log(TF), présence/absence,...).
- **IDF (Inverse of Document Frequency) :** ce facteur mesure l'importance d'un terme dans toute la collection. Un terme qui apparaitre souvent dans la base documentaire ne doit pas avoir le même impact qu'un terme moins fréquent. Il peut être calculé selon:

$$idf = \log\left(\frac{N}{n_i}\right) \quad (1)$$

$$idf = \log\left(\frac{N-n_i}{N}\right) \quad (2)$$

Où :

- n_i est le nombre de documents contenant le terme
- N le nombre total de documents de la collection.

La mesure TF * IDF est une bonne approximation de l'importance d'un terme dans un document, elle est particulièrement dans les corpus de documents de tailles homogènes, tels que les corpus contenant des résumés. Cette mesure a eu un succès limité dans les corpus de tailles très variables **[17]**.

6.2. L'appariement requête-document (recherche)

La comparaison entre le document et la requête revient à calculer un score, supposé représenter la pertinence du document vis-à-vis de la requête. Cette valeur est calculée à partir d'une fonction ou d'une probabilité de similarité notée RSV **(Q, d)** (Retrieval Status Value), où **Q** est une requête et **d** un document. Cette mesure tient compte du poids des termes dans les documents, déterminé en fonction d'analyses statistiques et probabilistes **[15].**

Plus précisément, l'appariement requête-document dépend du modèle de RI utilisé. Un modèle de RI définit la manière dont la requête et les documents sont représentés, ainsi que, le modèle formel qui permet

d'interpréter cette notion de pertinence. Plusieurs modèles ont été proposés dans le domaine comme les modèles vectoriels, probabilistes,...etc., qui seront détaillés dans le chapitre 2.

6.3. Reformulation de requêtes

De façon générale les utilisateurs qui font la recherche ne maitrisent pas le domaine et tous les termes accessibles par ce dernier. Le processus de reformulation de requêtes est utilisé lorsqu'un utilisateur est incapable de reformuler sa requête du début afin de donner une information pertinente. Le principe de reformulation est basé sur l'ajout des termes à la requête initiale ou l'ajustement des poids des index. Nous distinguons principalement deux approches pour la reformulation **[17]** :

6.3.1. Reformulation directe

Elle consiste à ajouter de nouveaux termes à la requête initiale. Cette modification est réalisée grâce aux liens de cooccurrence entre les termes On parle alors de reformulation de requêtes basée sur les concepts (Concept-based Query Reformulation).

6.3.2. Reformulation indirecte

Dans cette approche la requête est modifiée en tenant compte d'une liste de documents déjà jugés sélectionnés. Ce processus est appelé réinjection de la pertinence (relevance feed-back) si le processus est supervisé et de pseudo réinjection de pertinence si le processus est automatique. Cette méthode a un double avantage : une simplicité d'exécution pour l'utilisateur qui ne s'occupe pas des détails de la reformulation, et un meilleur contrôle du processus de recherche en augmentant le poids des termes importants et en diminuant celui des termes non importants.

7. Domaines d'application

La RI est un domaine vaste qui se situe dans les frontières de plusieurs disciplines tel que **[25]** :

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

8. Conclusion

Ce premier chapitre est porté essentiellement sur les principes fondamentaux de la RI de manière générale, nous avons décrit les concepts de base liés a ce domaine et nous avons présenté l'architecture typique de n'import quel systèmes de recherche d'information notamment l'appariement document/requête et la reformulation des requêtes puis nous avons présentés les étapes essentielles d'une bonne indexation.

Chapitre II

Les Modèles de Recherche d’information

1. Introduction

L'un des rôles d'un système de recherche d'information est de mesurer la pertinence d'un mot par rapport à une requête. Un modèle de RI fournit une formalisation au processus de recherche d'information. Il doit accomplir plusieurs rôles dont le plus important est de fournir un cadre théorique pour la modélisation de cette mesure de pertinence.

Ce chapitre a pour but de présenter les différentes modèles de recherche d'information, ainsi que le rôle de ces modèles dans le domaine de recherche d'information.

2. Le rôle du modèle de recherche d'information

Si l'indexation qui choisit les termes pour représenter le contenu d'un document ou d'une requête, c'est au modèle de leur donner une interprétation. Étant donné un ensemble de termes pondérés issus de l'indexation [15], le modèle remplit les deux rôles suivants :

- ✓ Créer une représentation interne pour un document ou pour une requête basée sur ces termes;
- ✓ Définir une méthode de comparaison entre une représentation de document et une représentation de requête afin de déterminer leur degré de correspondance (ou similarité)

3. Les modèles de Recherche d'information

Le modèle joue un rôle central dans la RI. C'est lui qui détermine le comportement clé d'un système de RI. De nombreux modèles ont été proposés pour accomplir cette tâche, ils sont généralement regroupés autour des trois familles de base, à savoir :

- ❖ Les modèles ensemblistes
- ❖ Les modèles algébriques
- ❖ Les modèles probabilistes

La figure 03 présente une taxonomie des principaux modèles de RI.

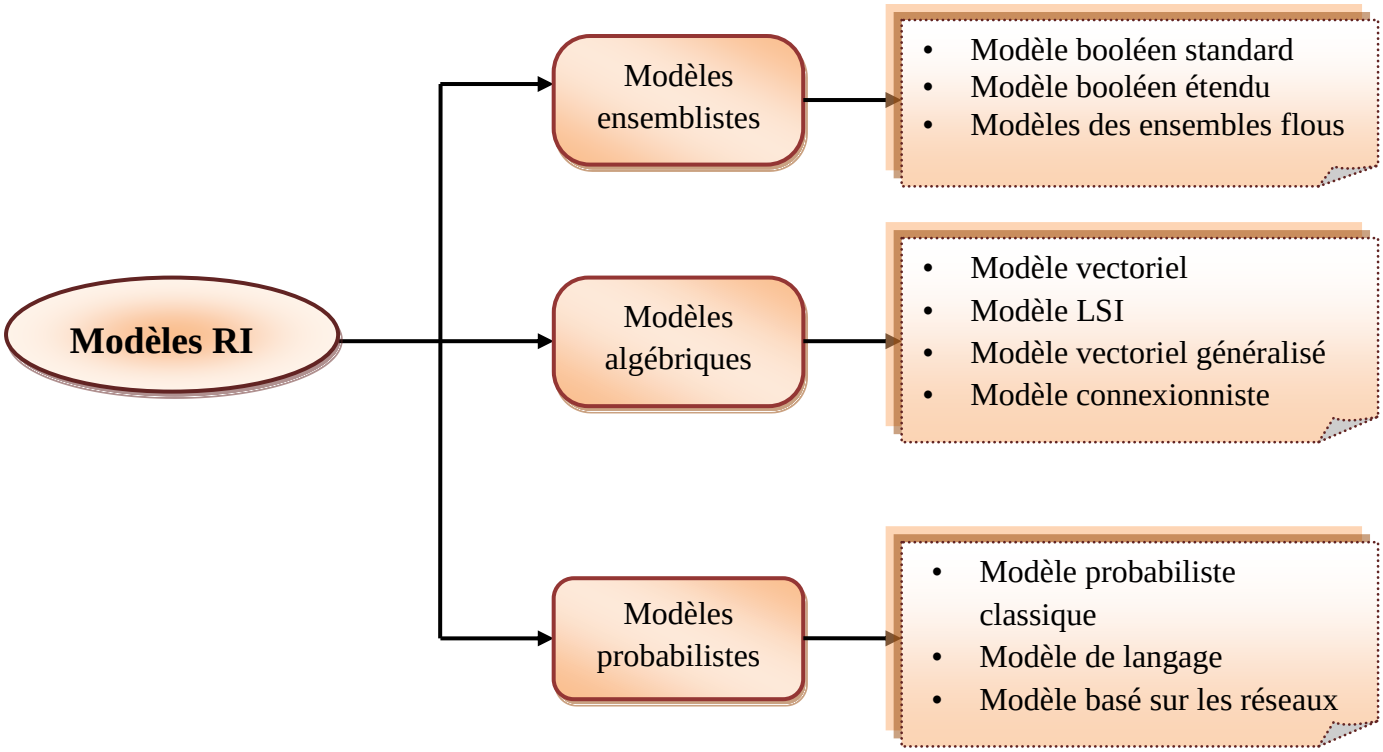


Figure 03 : Taxonomie des principaux modèles de RI

3.1. Les modèles ensemblistes

Les modèles ensemblistes reposent sur la théorie des ensembles. Dans ces modèles, les termes de la requête sont séparés par des opérateurs logiques : conjonction (ET), disjonction (OU) et négation (NON). Ces opérateurs permettent d'effectuer des opérations d'union « OU », d'intersection « ET » et de différence « NON » entre les ensembles de résultats associés à chaque terme **[14]**. Les différentes variantes de ce type de modèle sont :

3.1.1. Modèle booléen

Le modèle booléen est le premier modèle qui s'est imposé dans le monde de la recherche d'information.

Il se base sur la manipulation des ensembles et l'algèbre de Boole. Dans ce modèle une requête est une expression logique composée de termes séparés par des opérateurs logiques (ET, OU et NON) **[14]**.

Les poids des termes dans l'index sont binaires, c'est-à-dire que les termes sont présents ou absents du document ($w_{ij} \in \{0,1\}$). Le modèle booléen utilise l'appariement exact, c'est-à-dire qu'il ne permet de restituer que les documents appartenant à l'ensemble décrit par la requête. La similarité RSV[1] entre un document et une requête est définie par **[14]** :

[1] Retrieval Status Value

$$\textbf{RSV}\ (q, d) = \begin{cases} \textbf{1} & \text{si } \textbf{d} \text{ appartient à l'ensemble décrit par } \textbf{q} \\ \textbf{0} & \text{sinon} \end{cases} \quad (3)$$

Ainsi, un document est considéré dans le modèle booléen comme étant pertinent, ou bien non pertinent.

Les résultats de la fonction de similarité ne permettent pas de renvoyer à l'utilisateur une liste ordonnée de documents, ce qui empêche le modèle d'avoir de bonnes performances **[14]**.

3.1.2. Modèle booléen étendu

Ce modèle est une extension du modèle booléen introduit par Salton en 1983 **[26]**, l'idée est de donner un poids chaque terme du document, Il tient compte de l'importance des termes dans la représentation des documents tout en proposant une pertinence graduée, Ce modèle peut être vu comme une combinaison des modèles booléen et vectoriel. Le poids d'un terme dans un document est une valeur comprise entre 0 et 1.

La requête reste toujours une expression logique **[18]**.

Considérons un ensemble de termes $\{t_1, \ldots, t_N\}$ et soit « wd_{ij} » le poids du terme «t_i» dans le document «d_j», où $D = (wd_{1j}, \ldots, wd_{Nj})$, avec $1 \leq i \leq N$ et $0 \leq wd_{ij} \leq 1$

La correspondance entre une requête et un document est définie de la façon suivante :

$$\textbf{Opérateur OU :} \quad \mathrm{RSV}(D_j, Q_k) = \left(\frac{\sum_{i=1}^{n} (wq_{ij}^{P} \times wd_{ij}^{p})^{\frac{1}{p}}}{\sum_{i=1}^{n} wq_{ik}^{p}} \right)^{\frac{1}{P}} \quad (4)$$

$$\textbf{Opérateur ET :} \quad \mathrm{RSV}(D_j, Q_k) = \left(\frac{\sum_{i=1}^{n} (wq_{ij}^{P} \times (1 - wd_{ij}^{p}))^{\frac{1}{p}}}{\sum_{i=1}^{n} wq_{ik}^{p}} \right)^{\frac{1}{P}} \quad (5)$$

Où :

- $0 \leq p \leq 1$ est une constante,
- wq_{iq}^{p} le poids du terme t_i dans la requête Q_k.

3.1.3. Modèles des ensembles flous

Une extension du modèle booléen est basée sur la théorie des ensembles flous proposée par Zadeh en 1965 **[19]**. Dans la théorie des ensembles flous, quand un élément a un degré d'appartenance à un ensemble, cet ensemble est dit ensemble flou. Cette théorie a influencé les chercheurs en RI pour modéliser les notions d'incertitudes et d'imprécisions qui existent à différents niveaux du processus de RI **[19]**. Dans ce modèle, un document est représenté comme un ensemble de termes pondérés comme suit :

$$\mathbf{D_j} = \{(t_1, a_1), \ldots, (t_i, a_i), \ldots\} \quad (6)$$

Où : «a_i» est le degré d'appartenance du terme « t_i » au document «D_j».

La correspondance RSV entre une requête « Q_k » et un document « D_j » est déterminée comme suit :

$$\textbf{RSV}\ (D_j, q_1 \wedge q_2) = \min\ (\text{RSV}\ (D_j, q_1)\ ,\ \text{RSV}\ (D_j, q_2)) \quad (7)$$

$$\textbf{RSV}\ (D_j, q_1 \vee q_2) = \max\ (\text{RSV}\ (D_j, q_1)\ ,\ \text{RSV}\ (D_j, q_2)) \quad (8)$$

$$\textbf{RSV}\ (D_j, \neg q_i) = 1 - (\text{RSV}\ (D_j, q_i)\ ) \quad (9)$$

Les objectifs pour lesquels les modèles de recherche d'information intègrent les ensembles flous sont **[15]** :

- ✓ de réduire l'imperfection et de traiter l'imprécision qui caractérise le processus d'indexation,
- ✓ de contrôler l'imprécision de l'utilisateur dans sa requête
- ✓ de traiter les réponses reflétant la pertinence partielle des documents par rapport aux requêtes.

L'inconvénient majeur de ces modèles est qu'ils ne sont pas adaptés au classement des documents pertinents, étant donné que les scores de pertinence qu'ils attribuent aux documents sont calculés par des fonctions min ou max qui ne prennent pas nécessairement en compte toutes les valeurs de pertinences des termes de la requête **[15]**.

3.2. Les modèles algébriques

Les modèles algébriques se basent sur la théorie algébrique. Dans ces modèles, la pertinence d'un document vis-à-vis d'une requête est définie par des mesures de distance (ou similarité) dans un espace vectoriel. Les différentes modèles de ce type sont :

3.2.1. Modèle vectoriel

Le modèle vectoriel (nommé aussi VSM[2]). Ce modèle propose de représenter les documents et les requêtes par des vecteurs d'indexation dans un espace engendré par les termes d'indexation. Le modèle vectoriel représente les requêtes et les documents sous forme de vecteurs dans un même espace vectoriel **[20]**. Soit l'espace vectoriel défini par l'ensemble des termes :

$$T = < t_1, t_2, t_3, \ldots, t_n > \quad (10)$$

La mesure de similarité entre le document et la requête représenté par les vecteurs suivants :

$$d = (d_1, d_2 \ldots d_n) \quad (11)$$

$$q = (q_1, q_2 \ldots q_n) \quad (12)$$

Où :

- « d_i » et « q_i » représente le poids de terme «t_i» dans le document et dans la requête.

Ils existent plusieurs mesures pour calculer la similarité entre le document et la requête, voir le tableau (01) suivant :

[2] Vector Space Model

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

Tableau 01: les mesures de similarité utilisé dans le modèle

Sauf la première formule, toutes les autres sont normalisées[3] .

Toutes ces mesures ont l'avantage :

- Ils permettent la pondération des termes, ce qui augmente les performances du système;
- Ils permettent de renvoyer des documents qui répondent approximativement à la requête et effectivement de trier les documents répondant à une requête.
- Les documents sont en effet restitués dans un ordre décroissant de leur degré de similarité avec la requête **[15]**.

Le principal inconvénient du modèle vectoriel est le fait qu'il suppose que les termes d'indexation forment une base. Or ils existent énormément de relations sémantiques qui font qu'un terme pourra s'exprimer en fonction des autres[4]. Par ailleurs il est très difficile voire impossible de traduire des relations par des combinaisons linéaires de termes, or ceci s'avère indispensable à la construction de vraie base de termes d'indexation **[15]**.

3.2.2. Modèle LSI (Latent Semantic Indexing)

Le modèle LSI est basé sur la décomposition en valeurs singuliers (SVD[5]) de la matrice termes-document,

[3] C'est-à-dire qu'elles donnent une valeur dans l'intervalle [0, 1].
[4] Les ontologies sont des représentations très complexes de telles relations.
[5] Singular Value Decomposition

qui représente l'espace d'indexation du modèle vectoriel. Cette décomposition permet de projeter la matrice termes-documents dans un espace de dimension réduit permettant de faire ressortir les relations sémantiques latentes entre mots de document. Ces relations sont basées sur la notion de cooccurrence où deux mots peuvent être considérés sémantiquement proches s'ils apparaissent dans des contextes (ou documents) similaires. Ainsi, dans ce modèle, les documents qui partagent des termes co-occurrents proches sont groupés (ou clustérisés) dans une seule représentation **[21]**. la SVD se décomposer en :

$$X_{n*d} = T_{n*m} \times S_{m*m} \times D'_{m*d} \quad (13)$$

Où :

- X_{n*d} : matrice termes-documents.
- n : le nombre de termes distincts de la collection
- d : le nombre de documents dans cette collection
- T_{n*m} : la matrice orthogonale des vecteurs singuliers de gauche.
- m : le rang de M, tel que (m≤ min (n, d)).
- S_{m*m} : la matrice diagonale triée des valeurs singulières.
- D'_{m*d} : la matrice contenant les colonnes orthogonales des vecteurs singuliers de droite.

Une fois que la SVD de la matrice «*X*» est calculée, il s'agit de réduire «$X_{n\,x\,d}$» par la matrice «Y_{n*d}» contenant uniquement les *«k» t*ermes ayants les plus grandes valeurs singulières de «S_{m*m}»

La matrice réduite «Y_{n*d}»dans l'espace de dimension« *k»* est calculée par la formule suivante :

$$Y_{n*d} = T_{n*k} \times S_{k*k} \times D'_{k*d} \quad (14)$$

D'autre part, la requête «Q» est aussi transformée dans ce nouvel espace en un pseudo- document D_Q comme suit :

$$D_Q = X'_Q \times T_{n*k} \times S^{-1}_{k*d} \quad (15)$$

Où : «X_Q» est le vecteur contenant les mots-clés de la requête «Q». « X'_Q» est son transposé.

La requête, ou pseudo-document, est ajoutée dans la matrice « D_{k*d}» comme un nouveau document. Lors de la recherche, le système calcule la similarité entre chaque paire de documents en vérifiant la formule (16), puis les documents qui sont proches sémantiquement sont comparés au pseudo-document «D_Q» suivant le modèle vectoriel afin de calculer le degré de pertinence entre la requête «Q» et ces documents **[21]**.

$$Y_{t*d} \times Y'_{t*d} = D_{k*d} \times S^2_{k*k} \times D'_{k*d} \quad (16)$$

Les avantages principaux du modèle LSI **[21]** :

- il pouvoir de retrouver les documents pertinents pour une requête utilisateur même s'ils ne partagent aucun mot avec elle.
- permet de résoudre partiellement les problèmes liés à la polysémie et la synonymie des mots dans la représentation des documents.

Néanmoins, ce modèle perd son efficacité comparativement au modèle vectoriel, lorsque le nombre de documents est faible. En effet, une collection de petite taille donne une approximation erronée de la matrice documents-termes dans l'espace réduit **[21]**.

3.2.3. Modèle connexionniste

Les SRI basés sur l'approche connexionniste utilisent le fondement des réseaux de neurones, tant pour la modélisation des unités textuelles que pour la mise en oeuvre du processus de RI. L'idée de base est que la RI est un processus associatif qui peut être représenté par les mécanismes de propagation d'activation des réseaux de neurones. De plus, les capacités d'apprentissage de ces modèles peuvent permettre d'obtenir des SRI adaptatifs **[22]**.

Deux modèles théoriques ont été utilisés : les modèles à auto-organisation et les modèles à couches.

- **Les modèles à auto-organisation :** permettent à partir de la description des documents, d'en réaliser une classification par l'apprentissage du réseau de neurones. Ces modèles sont basés sur les cartes auto-organisatrices de Kohonen **[22]**.
- **Les modèles à couches :** Les SRI basés sur un modèle connexionniste à couches sont représentés par un minimum de trois couches de neurones interconnectées : la couche requête (Q), la couche termes (T) et la couche documents (D). Le mécanisme de recherche est basé sur une activation initiale des neurones termes induite par une requête, et qui se propage vers les documents à travers les connexions du réseau **[22]**. Dans le modèle MERCURE, une requête Q est représentée par un vecteur de poids sous forme :

$$\mathbf{Q_u^{(t)}} = (q^{(t)}_{u1}, q^{(t)}_{u2,\ldots,} q^{(t)}_{uT}) \quad (17)$$

Les poids des termes dans la requête sont affectés aux liens requête-termes. L'activité initiale du réseau correspond à l'activation d'un noeud requête en envoyant un signal de valeur 1 à travers les liens requête-termes. Chaque neurone terme « t_j » affecté par la requête, reçoit une entrée « In (t_j) » et fournit une sortie « Out (t_j) » respectivement définies par :

$$\text{In } (t_j) = q_{uj}^{(t)} \quad (18)$$

$$\text{Out } (t_j) = g \text{ (In } (t_j)) \quad (19)$$

Un document «d_i» qui a des termes «t_j» en commun avec la requête recevra une entrée «In (d_i)» et Calculera sa sortie «Out (d_i)» telles que :

$$In(d_i) = \sum_{i=1}^{T} Out(t_j) * W_{ij} \quad (20)$$

$$Out\ (d_i) = g\ (In\ (d_i)) \quad (21)$$

Où «w_{ij}» est le poids du terme «t_j» dans le document «d_i».

Les valeurs de sortie des différents documents correspondent à leurs degrés de pertinence pour la requête donnée **[22]**.

3.3. Les modèles probabilistes

Les modèles probabilistes se basent sur la théorie des probabilités. Pour ces modèles, la pertinence d'un document vis-à-vis d'une requête est vue comme une probabilité de pertinence document/requête. Les différentes modèles de ce type sont :

3.3.1. Le modèle probabiliste classique

Le modèle probabiliste aborde le problème de la recherche d'information dans un cadre probabiliste.

Le principe de base consiste à présenter les résultats de recherche d'un SRI dans un ordre basé sur la probabilité de pertinence d'un document vis-à-vis d'une requête. Robertson résume ce critère d'ordre par le "principe de classement probabiliste", aussi désigné par PRP[6].

Etant donnés une requête utilisateur notée «Q» et un document «D», formellement, le modèle PRP peut être traduit de la manière suivante : pour chaque document «D» et chaque requête «Q», Quelle est la probabilité que ce document soit pertinent pour cette requête ? Deux événements sont alors possibles **[23]** :

- « R, D » est pertinent pour Q;
- « $\overline{R}$, D » est non pertinent pour Q.

Selon PRP, le score d'appariement entre le document «D» et la requête, noté RSV (Q, D), est donné par :

$$RSV(Q, D) = \frac{P(R\backslash D)}{P(\overline{R}\backslash D)} \quad (22)$$

Rappel du théorème de **Bayes** :

$$p\left(\frac{A}{B}\right) = \frac{p\left(\frac{B}{A}\right).p(A)}{p(B)} \quad (23)$$

En utilisant la règle de Bayes et en simplifiant, cela revient à ordonner les documents selon :

$$RSV(Q, D) = \frac{P(D\backslash R)}{P(D\backslash \overline{R}} \quad (24)$$

[6] Probability Ranking Principle

Plusieurs solutions ont été proposées pour représenter le document D et pour estimer les paramètres du modèle. Parmi elles citons BIR[7].

3.3.2. Les réseaux inférentiels bayésiens

Un réseau inférentiel bayésien est un graphe de dépendances, orienté et acyclique. Dans ce graphe les nœuds représentent des variables propositionnelles (ou également des constantes) et les arcs des liens de dépendances entre les nœuds. Ainsi, si la proposition représentée par le nœud «p» cause ou implique la proposition représentée par le nœud «q», on trace alors un arc de «p» vers «q» **[24]**.

Dans le contexte de la recherche d'information les nœuds et les arcs sont définis comme suit :

- **Les nœuds:** représentent des concepts, des groupes de termes ou des documents.
- **Les arcs:** représentent les dépendances entre termes et entre termes et documents.

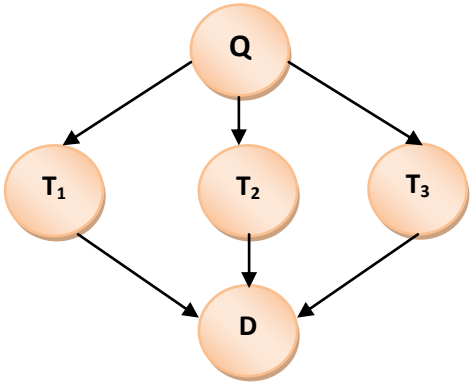


Figure 04 : Modèle de réseaux bayésien simple.

Le réseau inférentiel de la figure 04 illustre le réseau de Tortue de pertinence d'un document vis a vis d'une requête composée de trois termes. Les nœuds de la requête représentent des variables aléatoires ayant pour valeur 0 ou 1. L'événement "la requête est accomplie"(Q = 1) est réalisé si le sujet lie a un terme (T1 = 1, T2 = 1 ou T3 = 1), ou si une combinaison de ces événements sont vrais. Les trois sujets sont infères par l'événement "le document est pertinent" (D = 1). Par Recherche d'information : concepts de base et modèles **[24]**.

L'enchainement de règles de probabilités, la probabilité jointe des autres nœuds du graphe est :

$$P(D, T1, T2, T3, Q) = P(D)\, P(T_1|D)\, P(T_2|D, T1)\, P(T_3|D, T1, T2)\, P(Q|D, T1, T2, T3) \quad (25)$$

La direction des arcs indique les relations de dépendance entre les variables aléatoires. L'équation (25) devient :

$$P(D, T1, T2, T3, Q) = P(D)\, P(T_1|D)\, P(T_2|D)\, P(T_3|D)\, P(Q|T_1, T2, T3) \quad (26)$$

La probabilité de réalisation de la requête P (Q = 1|D = 1) peut être utilisée comme score de rangement des documents :

(27)

[7] Binary Independance Retrieval

$$P(Q = 1|D = 1) = \frac{P(Q = 1, D = 1)}{P(D = 1)} = \frac{\sum P(D = 1, T_1 = t_1, T_2 = t_2, T_3 = t_3, Q = 1)}{P(D = 1)}$$

Le modèle nécessite la connaissance de P (D = [0|1]), P (T_i = [0|1]|D = [0|1]), P(Q = [0| 1]|(T_1,T2 . . .T_n) $\in \{0,1\}^n$), cette dernière est la plus difficile à trouver car le nombre de probabilités à spécifier augmente exponentiellement avec le nombre de termes dans la requête **[24].**

3.3.3. Les modèles de langage

Par modèle de langue, on désigne une fonction de probabilité «P» qui assigne une probabilité «P(s)» à un mot ou à une séquence de mots s = $m_1m_2 \ldots m_n$ en une langue **[20]**.

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

$$P(S) = \prod_{i=1}^{n} P(mi|m1 \ldots mi - 1) \quad (28)$$

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

- Uni-gramme : $P(S) = \prod_{i=1}^{n} P(mi)$ (29)
- Bi-gramme : $P(S) = \prod_{i=1}^{n} P(mi|mi - 1) = \prod_{i=1}^{n} \frac{P(mi-1mi)}{P(mi-1)}$ (30)
- Tri-gramme : $P(S) = \prod_{i=1}^{n} P(mi|\, m_{i-2}m_{i-1}) = \prod_{i=1}^{n} \frac{P(m_{i-2}m_{i-1}m_i)}{P(m_{i-2}m_{i-1})}$ (31)

3.4. Autres modèles de recherche d'information

3.4.1. Modèles logiques

Le modèle logique introduit basé sur l'idée qu'un document est pertinent s'il implique logiquement la requête, représente un modèle intelligent. Ces modèles ne se limitent pas à faire une correspondance simple des représentations des requêtes avec celles des documents, mais ils permettent aussi l'évaluation de cette correspondance. Cette évaluation se base sur les comparaisons sémantiques entre le document et la requête. Elle se traduit par des déductions permettant de définir des fonctions de correspondance beaucoup plus perfectionnées, et ce à partir d'un ensemble de relations sémantiques entre les termes. Les techniques utilisées pour établir ces déductions sont issues de l'intelligence artificielle. Les relations sémantiques les plus couramment utilisées sont **[10]** :

- ✓ La généricité,
- ✓ La spécificité,
- ✓ Le voisinage sémantique,

- ✓ La synonymie et/ou la synonymie contextuelle.

Cette modélisation logique de la recherche d'information consiste à considérer qu'un document répond a une requête si l'on peut trouver une chaine de causalité qui part du document et arrive à la requête. Son utilisation doit permettre de mieux prévoir des résultats expérimentaux et de mieux concevoir les SRI. Pour ce faire, le modèle logique doit être basé sur un certain nombre d'hypothèses. Parmi ces hypothèses :

- La première hypothèse est de considérer formalisable le processus de recherche d'information, ceci suppose l'existence de formalismes dans lesquels le contenu des documents et les besoins de l'utilisateur peuvent être partiellement exprimés (dans un langage formel).
- La deuxième hypothèse est de considérer pertinent un document lorsque la correspondance entre ce document et la requête est directe et inverse. Dans ce contexte, nous pouvons utiliser la définition sémantique informelle de ces relations à savoir le critère d'exhaustivité et le critère de spécificité, …etc **[10]**.

3. Conclusion

Les modèles de recherche constituent le noyau des SRI qui permet de représenter et de comparer des représentations des documents et des requêtes. Ils utilisent des mesures de distances basées sur les attributs (les termes composant les documents) comme les méthodes de classification ou de catégorisation de documents textuels. Chacun de ces modèles ou stratégies contribue à la résolution des problèmes inhérents à la recherche d'information.

La finalité de chaque système de recherche d'information est de satisfaire les besoins des utilisateurs. Ces derniers sont préoccupés par un seul problème : celui de pouvoir récupérer tous les documents dont il a besoin d'une façon rapide et efficace.

Chapitre III

Conception et Architecture du Système

1. Introduction

L'objectif premier d'un système de recherche d'information est d'orienter une recherche ciblée dans un ensemble de documents à travers un besoin exprimé par une requête utilisateur. Dans un système de recherche d'information les documents sont manipulés sous une base informationnelle et sémantique. Pour cela, le système devrait assurer plusieurs fonctions dont la communication, le stockage, l'organisation et la recherche d'information.

Les modèles proposés et étudiés dans la littérature sont essentiellement basés sur des approches formelles (booléen, vectoriel, probabiliste). Ainsi que des techniques de mesure d'heuristique de recherche (modèle de Markov caché HMM, réseaux de neurones et algorithmes génétiques).

Pour notre projet, nous avons utilisé ses systèmes dans un but précis qui est la détection de plagiat. Cela consiste à chercher dans un ensemble de documents les cas de reprise des mots ou des phrases clés d'un texte cible. Notre système est basé sur une recherche d'information simplifiée décrite dans les chapitres précédents. Dans ce chapitre, nous allons présenter en détail les différentes techniques implémentées dans notre système à savoir : la méthode de représentation, la méthode de pondération et la méthode d'appariement. Nous avons baptisé ce système « **PlagZoom**[1] ».

2. Définition de plagiat

La définition de plagiat[2] d'après le dictionnaire de Larousse est un acte de quelqu'un qui, dans le domaine artistique ou littéraire, donne pour sien ce qu'il a pris à l'œuvre d'un autre.

3. Architecture générale du système « PlagZoom »

Nous avons envisagé de décomposer la réalisation de notre détecteur de plagiat « PlagZoom » en deux (02) phases complémentaires, qui sont :

- Phase d'indexation.
- Phase de recherche.

[1] **PlagZoom:** est la résultantes du chevauchement de deux mots qui sont: plagiat et zoom où on a scindé le premier en deux parties Plag et iat et remplacer le 2éme partie par Zoom pour que le résultat final soit PlagZoom.

[2] Définition de plagiat disponible sur http://www.larousse.fr/dictionnaires/francais/plagiat/61301 (Accédé le 14/05/2016)

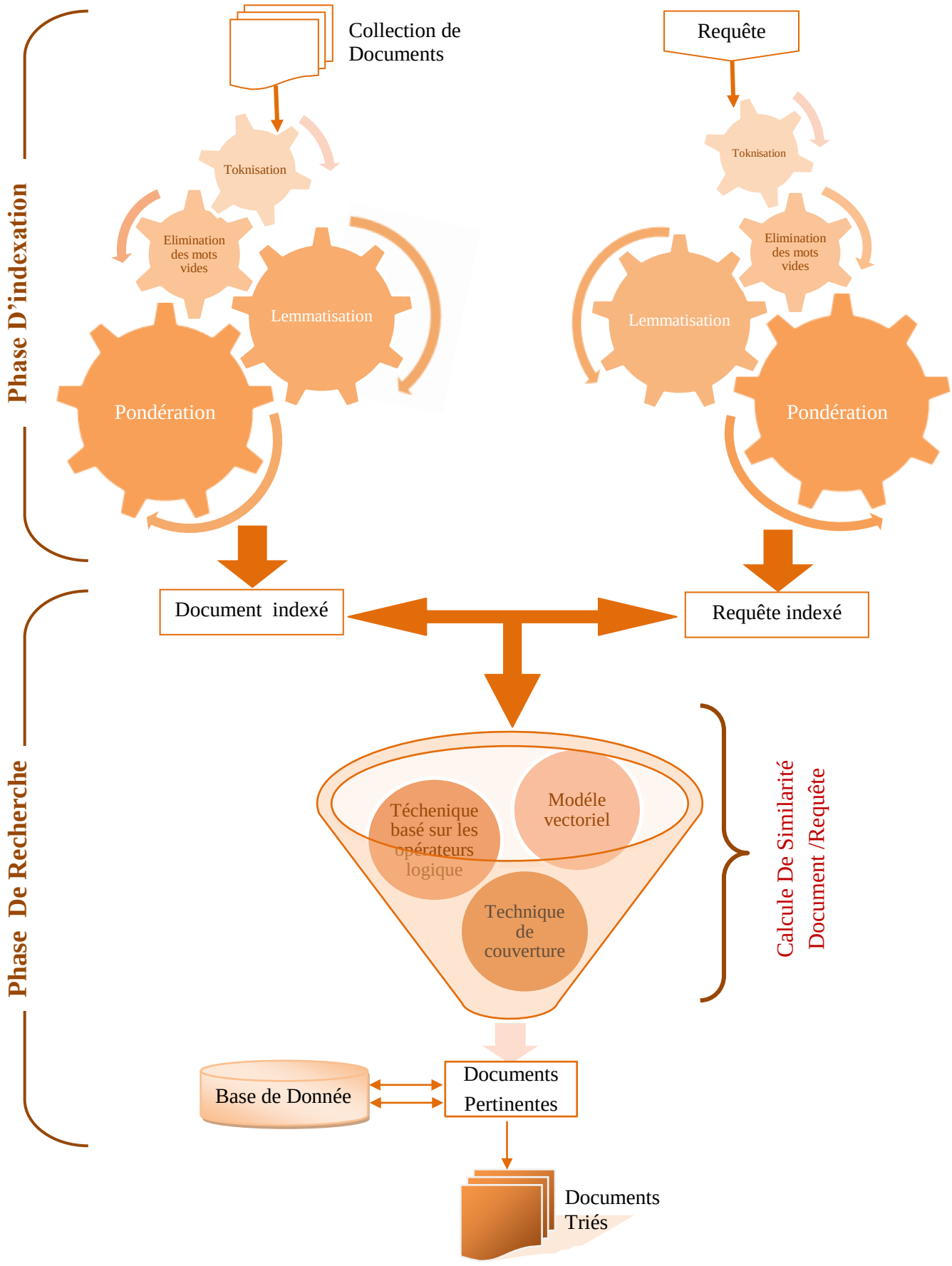
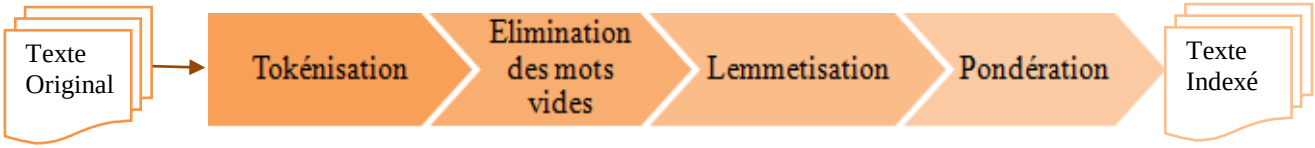


Figure 05 : Architecture générale du système « PlagZoom »

3.1. Phase d'indexation

L'indexation est l'opération qui vise à construire une structure d'indexe qui permet de retrouver très rapidement les documents incluant les mots demandés. Cette étape consiste à analyser chaque document de la collection afin de créer un ensemble de mots-clés. Son objectif est de trouver les concepts les plus importants du document (ou de la requête), qui formeront le descripteur du document.

La figure 06, Présente l'architecture générale de la phase d'indexation.



La figure 06 : L'architecture générale de la phase d'indexation

Cette phase nécessite les prétraitements suivants :

3.1.1. Tokénisation des documents

Cette étape consiste a transformé un texte en un ensemble de termes, on a choisi de considérer un mot comme une suite de caractères situés entre deux séparateurs et entre le vide.

La figure 07, présente la liste des séparateurs utilisée dans notre système.

'\n' , ' '

',' , ' ;' , ' :' , '.' , ' ?' , ' ! ' : La ponctuation

'=' , '+' , '-' ,'*' , '<' , '>' : les opérateurs arithmétiques et de comparaison

'(' , ')' , '[' , ']' , '{' , '}' : Les parenthèses

' « ' , '\' , '/' , ' , '%'.

'&' , '~' , '#' , '|' , '_' : les opérateurs logiques

'@' , '§' , '$' , '£'.

'0' , '1' , '2' , '3' , '4' , '5' , '6' , '7' , '8' , '9' : les chiffres.

Figure 07 : La liste des séparateurs

Dans cette opération chaque document sera représenté sans séparateurs (espaces de séparation des chiffres, des mots, des ponctuations, …etc.). La figure 08 présente un exemple de cette opération :

Entrée → Le cours, de Recherche d'Information (RI) du Master porte sur deux «02» axes.

Sortie →

Le	cours	de	Recherche	d	Information	RI	du	Master
porte	sur	deux	axes					

Figure 08 : Elimination des séparateurs

Afin de pouvoir réduire la taille du tableau il est nécessaire de formater les mots avec majuscules et les mots minuscules comme étant un seul mot. Par exemple les mots « master, Master, mASter, MASTER » considèrent comme un seul mot (traitement de la casse).

La figure 09 illustre l'exemple précédent en éliminant les majuscules.

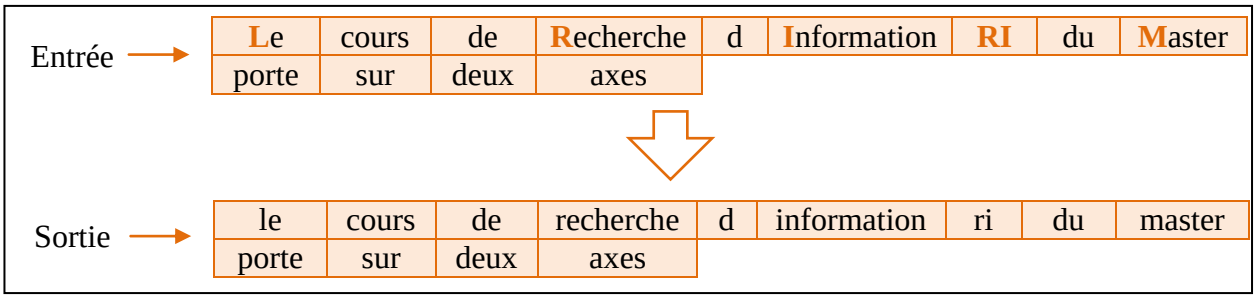


Figure 09 : Elimination des majuscules.

3.1.2. Elimination des mots vides

Les mots vides ou mots outils sont les mots non significatifs trouvés dans les documents. En effet, ces mots ne traitent pas le sujet du document mais ils permettent de lier entre les mots d'une phrase pour la structurer comme les articles, les conjonctions de coordination, les verbes auxiliaires,…etc.

Chaque langue a sa propre liste des mots vides. Dans notre application nous avons utilisé un fichier qui comporte 165 mots vides de la langue française. Ces mots ne portent pas de sens. La figure 10 représente un extrait des mots vides de la langue française:

au, aux, avec, ce, ces , dans, de, des, du, elle, en, et, eux, il, je, la, le, leur, lui, ma, mais, me, même, mes, moi, mon, ne, nos, notre, nous, on, ou, par, pas, que, qui ...

Figure 10: un extrait des mots vides de langue française.

Cette étape permet d'analyser et de réduire la taille de l'index. La figure 11 montre l'exemple précédent après élimination les mots vides.

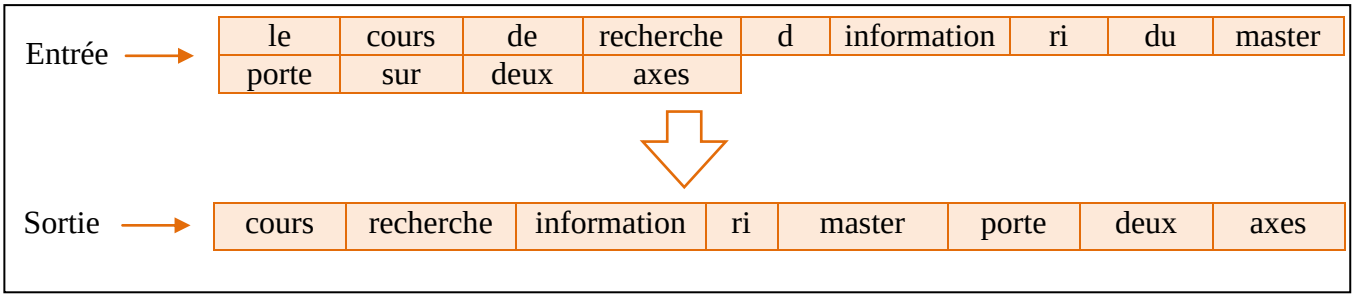


Figure 11 : Elimination des mots vides.

3.1.3. Lemmatisation

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

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

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

Il y a plusieurs algorithmes de lemmatisation telle que « l'algorithme de carry », « algorithme de paice/husk ».Dans notre système on a choisi d'utiliser « l'algorithme de porter », ce dernier est un algorithme de normalisation des mots. Il permet de supprimer les affixes des mots pour obtenir une forme canonique du mot. Cet algorithme a été proposé par Martin Porter en 1980. Il se présente comme un ensemble de règles dont l'application successive à un mot de l'anglais produit la racine de ce mot. Il reste toutefois un algorithme fondamental couramment utilisé en TALN [25].

Cette phase est consacré à réduire le nombre de terme dans le tableau et permet de représenter par un même descripteur des mots qui ont le même sens. Le cas de notre exemple comme indiqué dans la figure 12, on remarque que la taille du tableau a été réduite.

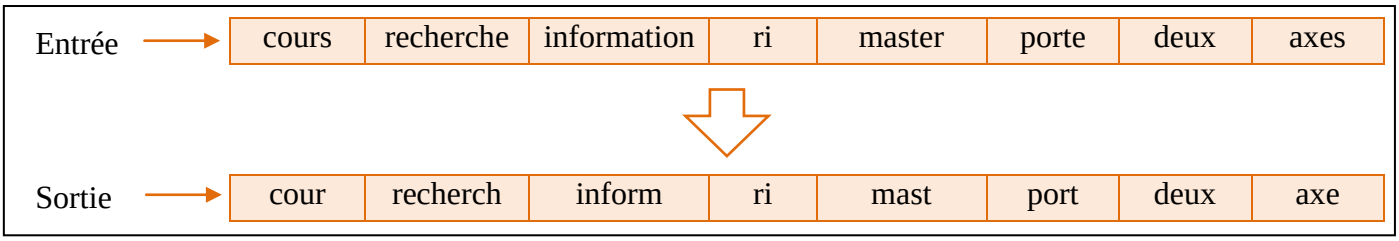


Figure 12 : lemmatisation des mots.

3.1.4. Pondération

Dans cette mesure statistique on affecte à chaque terme d'index extrait lors du traitement des documents un poids qui permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus.

Au niveau de notre système on a utilisé la formule « **Tf.Idf** » dans le calcul des poids. Cette formule est généralement utilisée pour le calcul des poids des termes dans les documents.

La formule « tf » est définie comme suit :

$$\mathrm{tf}_{(t,d)} = \frac{n_{t,d}}{N_d} \qquad (32)$$

Tel que :

- $n_{t,d}$: est le nombre d'occurrences de t dans d.
- N_d : la taille du document d (le nombre de mots).

La formule « idf » est définie comme suit :

$$\text{idf}_{(t,D)} = \log \frac{|D|}{|\{d \in D : t \in d\}|} \quad (33)$$

Tel que :

- $|D|$: nombre total de documents dans le corpus.
- $|\{d \in D : t \in d\}|$: le nombre de documents contenant le terme t.

Calcul de Tf.Idf : finalement, le poids s'obtient en multipliant les deux mesures

$$\text{tf.idf}_{(t,d,D)} = \text{tf}_{(t,d)} \times \text{idf}_{(t,D)} \quad (34)$$

Exemple

Document 1	**Document 2**	**Document 3**
nom celebr bocag frem ruisseau murmur vent emportent jusqu **arc** celest **arc** grac consol main tend nuag	pein distingu deux but extrem carri chen ombrag autour autr palmi dessin eclat soir	ah beau temp traval poetique beau jour pass pres premi inepuis joi paix libert derni empreint melancol bien charm **arc**

Tableau 02: les documents indexés.

Le tableau ci-dessus représente des documents indexés.
L'exemple porte sur le document 1 (soit $d1$) et le terme analysé est « arc » (soit $t1$ = arc).

- Calcul de tf

$$\text{tf}_{1,1} = \frac{n_{1,1}}{N_1} = \frac{2}{16}$$

Détails du calcul : la plupart des termes apparaissent une fois (14 termes), et ***arc*** apparaissent 2 fois (1 termes). Le dénominateur est donc 1*2 + 14 = 16. Cette somme correspond au nombre de mots dans le document.

- Calcul de idf

Le terme « arc » n'apparaît pas dans le deuxième document. Ainsi :

$$\text{idf}_1 = \log \frac{|D|}{|\{d \in D : t \in d\}|} = \log \frac{3}{2}$$

- Poids final

On obtient:

$$\text{tf.idf}_{1,1} = \frac{2}{16} . log \frac{3}{2} \approx 0.022$$

Pour les autres documents :

$$\text{tf.idf}_{1,2} = 0.log\frac{3}{2} \approx 0$$

$$\text{tf.idf}_{1,3} = \frac{1}{20}.log\frac{3}{2} \approx 0.008$$

Le premier document apparaît ainsi comme « le plus pertinent ».

3.2. Phase de recherche

Notre module de recherche est basé d'une part, sur un dictionnaire sous forme d'une base contenant des informations sur les documents sélectionnés. D'autre part, sur un programme de recherche à pour but, dans un premier temps, de mettre en correspondance une requête indexée et un document indexé.

3.2.1. Le dictionnaire de donnée

Notre détecteur «PlagZoom» adopte un dictionnaire pour préserver les informations utiles, ce dictionnaire est sous forme d'une base de données qui contient une seule table. Cette dernière contient toutes les informations concernent les documents pertinentes ou sélectionnés, ces informations correspond aux: Titres des documents, l'auteur du documents, l'année d'édition, l'université ou ils sont éditées et les types de ces documents (thèses ou revues), (voir la Figure 12).

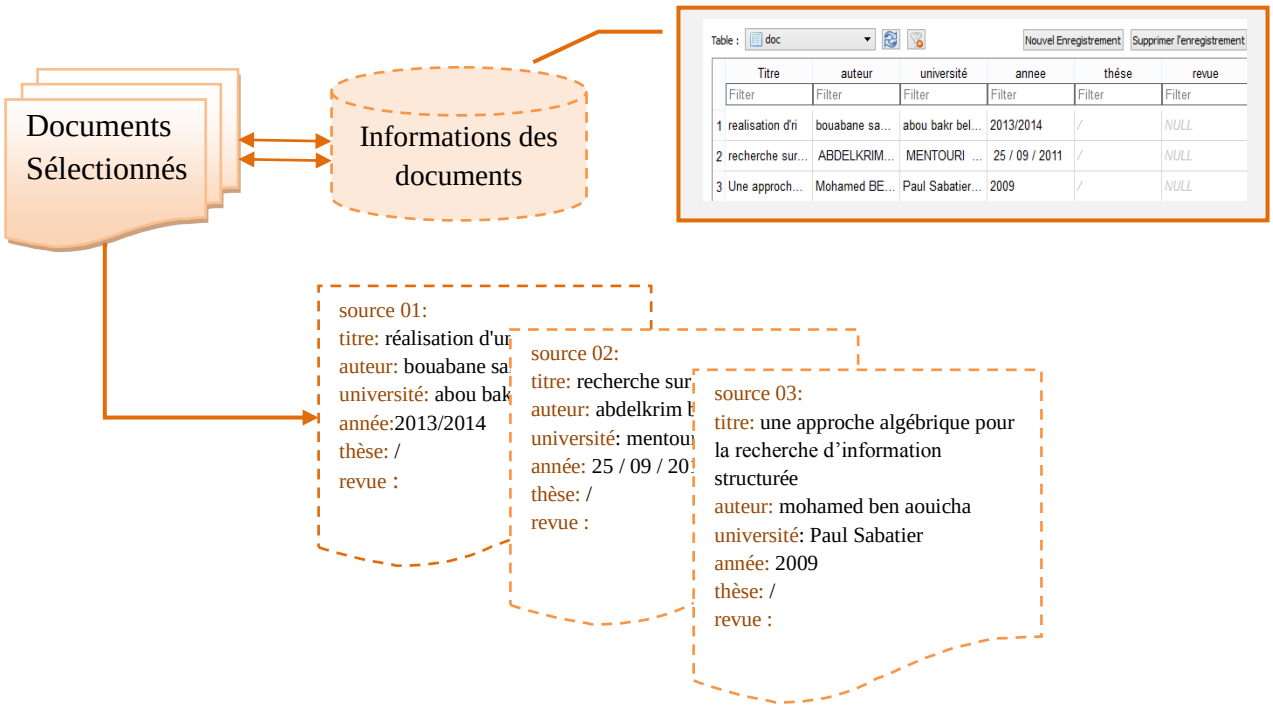


Figure 13 : Représentation de dictionnaire de données.

3.2.2. Programme de recherche

Notre programme de recherche comprend la fonction de décision fondamentale qui permet d'associer à une requête, l'ensemble des documents pertinents à restituer.

ce programme est basé sur des différentes techniques: la première est basée sur le modèle vectoriel, la deuxièmes est basée sur les opérateurs logiques et la troisièmes est basée sur la couverture requête/document , pour les bien comprendre on va les détaillées une par une .

3.2.2.1. la technique basée sur Modèle vectoriel (VSM[3])

Le modèle vectoriel vise à représenter mathématiquement un document- ainsi qu'une requête à son contenu sémantique. Selon ce modèle mathématique, chaque document et requête sont représentées par un vecteur dans un espace bien choisi, L'espace en question sera un espace dont chaque dimension correspondra à un mot « significatif » du dictionnaire de tous les mots utilisables. La taille du vecteur dépendra du nombre des mots significatifs dans la requête.

Par ailleurs, on va s'intéresser à la pondération de chaque mot significatif dans notre texte, chaque mot se verra donc associé à une valeur comprise entre 0 et 1, ce qui donnera la représentation vectorielle du texte.

pour mieux comprendre, on va citer cet exemple:

On utilisera un corpus contenant trois extraits et notre but est de déterminer quel document est le plus Pertinent pour la requête R.

- un extrait A du *Rouge et du Noir*
- un extrait B des *Misérables*
- un extrait C de *Candide*
- une requête R « Le crime de Omar était un crime, un crime affreux »

Dans un premier temps, on pourrait suggérer que le nombre d'occurrences de chaque mot dans le document constitue son « poids ».

Reprenons nos trois documents A, B et C et notre requête R. Voici les données dont nous avons besoin :

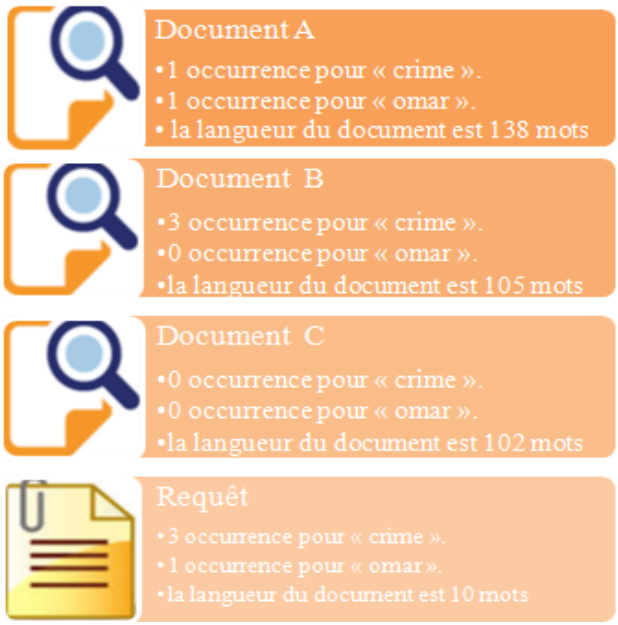


Figure 14 : représentation des données des documents et de requête.

[3] **VSM**: **V**ector **S**pace **M**odel

On va se limiter aux termes « omar » et « crime » et puisque le document C ne nous intéresse pas (à Première vue, il n'a aucune similarité avec notre requête) on ne le représentera pas. Cependant, on le Prendra en compte comme élément du corpus à part entière dans les calculs ci-dessous.

- ✓ Calcul des fréquences inverses

  « Omar » apparaît dans 1 document sur 3 : $\text{idf}(Omar) = \log\frac{3}{1} = 1{,}10$

  « crime » apparaît dans 2 documents sur 3 : $\text{idf}(Crime) = \log\frac{3}{2} = 0.405$

- ✓ Calcul des poids TF-IDF pour A

  $\text{tf-idf}(Omar)\text{A} = \frac{1}{138} \times \text{idf}(Omar) = \frac{1}{138} \times 1.10 = 0.0080$

  $\text{tf-idf}(Crime)\text{A} = \frac{1}{138} \times \text{idf}(Crime) = \frac{1}{138} \times 0.405 = 0.0029$

  On représentera le document A par le vecteur $\underset{OA}{\rightarrow} = (0.0080,\ 0.0029)$

- ✓ Calcul des poids TF-IDF pour B

  $\text{tf-idf}(Omar)\text{B} = \frac{0}{105} \times \text{idf}(Omar) = 0$

  $\text{tf-idf}(Crime)\text{B} = \frac{3}{105} \times \text{idf}(Crime) = \frac{1}{138} \times 0.405 = 0.012$

  On représentera le document B par le vecteur $\underset{OB}{\rightarrow} = (0,\ 0.012)$

- ✓ Calcul des poids TF-IDF pour R

  $\text{tf-idf}(Omar)\text{R} = \frac{1}{10} \times \text{idf}(Omar) = \frac{1}{10} \times 1.10 = 0.11$

  $\text{tf-idf}(Crime)\text{R} = \frac{3}{10} \times \text{idf}(Crime) = \frac{1}{10} \times 0.405 = 0.12$

  On représentera le document R par le vecteur $\underset{\text{OR}}{\rightarrow} = (0.11,\ 0.12)$

Une fois les documents et la requête modélisées selon le modèle vectoriel, la proximité sémantique entre chacun d'eux est exprimée par le cosinus de l'angle formé par leur vecteur respectif, c'est le fameux cosinus de Salton aussi appelé cosinus de similarité.

La similarité cosinus, entre autres, est pour cette raison parfois plus intuitive, elle vérifie que les documents « pointent » dans la même direction sans souci de leur norme, puisqu'elle est insensible à la multiplication par un scalaire.

$$similarité(A,R) = \cos\theta = \frac{A.R}{\|A\|.\|R\|} \tag{34}$$

revenant à notre exemple: Si on peut relier la similarité entre deux vecteurs (A et R) ou (B et R) à la mesure de l'angle θ qu'ils forment, alors on peut l'évaluer en calculant le cosinus de cet angle , c'est ainsi qu'est définie la similarité cosinus.

Le calcul du cosinus se base sur l'expression du produit scalaire $A.R = \|A\|.\|R\| \cos\theta$ et implique qu'aucun des deux vecteurs ne soit nul.

$$similarité(A,R) = \cos\theta = \frac{A.R}{\|A\|.\|R\|} = 0.97$$

$$similarité(B,R) = \cos\theta = \frac{B.R}{\|B\|.\|R\|} = 0.99$$

Le document B serait donc le plus pertinent pour R.

3.2.2.2. Technique basé sur les opérateurs logiques (TBOL[4])

Dans cette technique, on a jugé nécessaire qu'une deuxièmes indexation est indispensable pour rendre facile la recherche et surtout le résultat soit crédible. Cette deuxième indexation est conçue spécialement pour le corpus. son rôle ; après avoir bien sur lui assembler tous les documents ; de nous permettre de savoir avec exactitude non seulement les mots communs et non dans les documents mais ainsi leurs emplacement dans ces derniers, pour qu'en final on aura un seul fichier où toutes ces informations seront préservées .

La figure ci-dessous schématise un exemple de processus de l'indexation

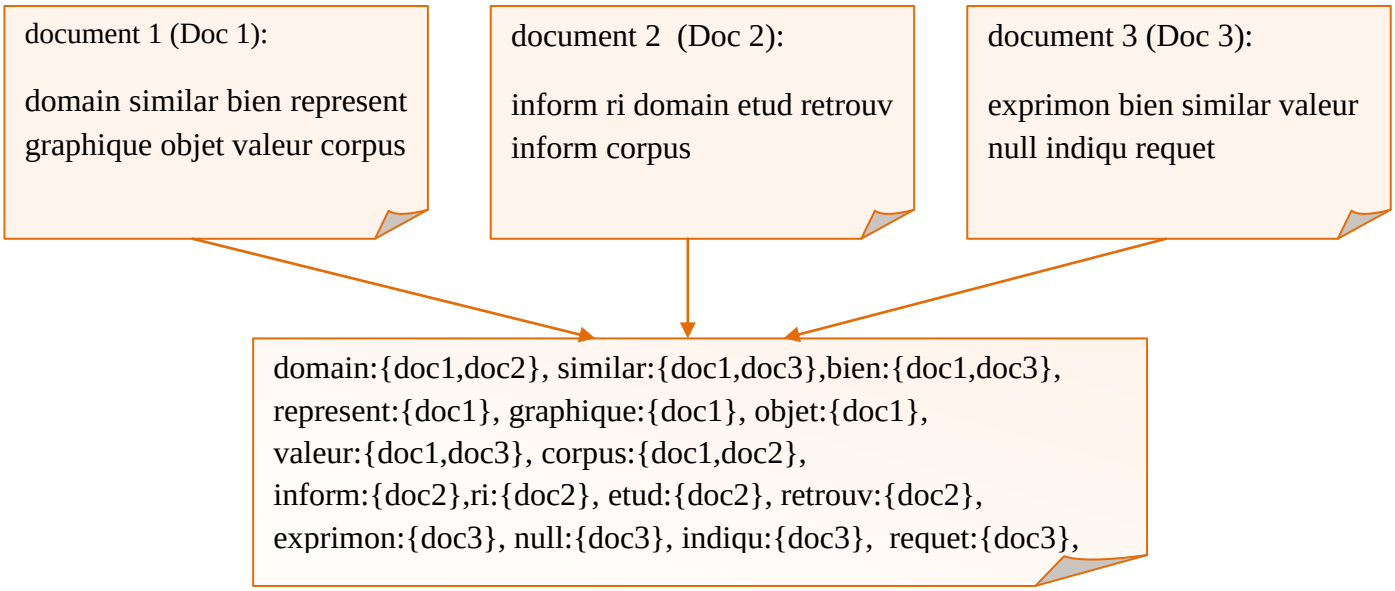


Figure 15 : un exemple sur la représentation de l'indexation de corpus.

Notre technique est une représentation mathématique du contenu d'une requête . les requêtes sont traitées comme des expressions logiques.

Exemple : q = (t1 ∧ t2) ∨ t3. tel que « q » est la requête et, « t » sont les termes.

Considérant un vocabulaire $T = t_1, \ldots, t_m$, un document est caractérisé par la présence ou l'absence de chaque t_i dans son contenu. La requête s'exprime alors avec des opérateurs logiques (et, ou). Un document du corpus est ainsi considéré comme pertinent uniquement quand son contenu est vrai pour l'expression de la requête.

[4] TBOL: **T**echnique **B**asé sur les **O**pérateurs **L**ogiques

La figure 16, schématise la technique basée sur les opérateurs logiques.

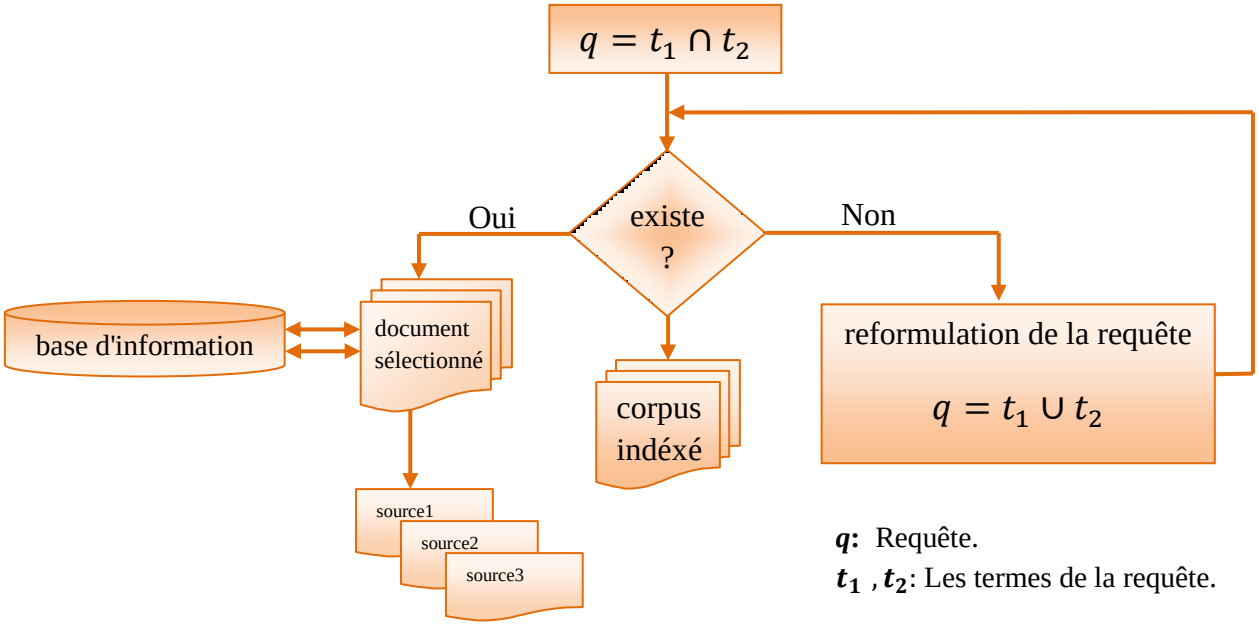
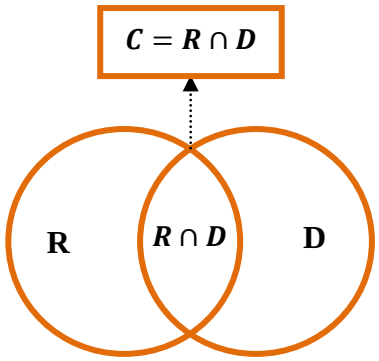


Figure 16 : Une représentation sur la technique basée sur les opérateurs logiques

3.2.2.3. Technique de couverture (CRD[5])

La technique de couverture qui couvre la requête par rapport au document et vis-versa, représente l'essentielle et l'ultime procédés dans notre programme **« PlagZoom ».** Cette technique nous permets, après avoir sur les mots communs entre documents et requête, de calculer le rapport entre le nombre des mots partagés et le nombre des mots de la requête donnant ainsi une idée précise sur la couverture de la requête par rapport au document et inversement, le rapport du nombre des mots communs au nombre des mots du document donneront la couverture de document par rapport à la requête.

Donc cette méthode vient en $3^{éme}$ rang pour consolider les deux techniques suscitées en calculant le nombre de mots partager entre le document et la requête. La figure ci-dessous schématise la technique de couverture.



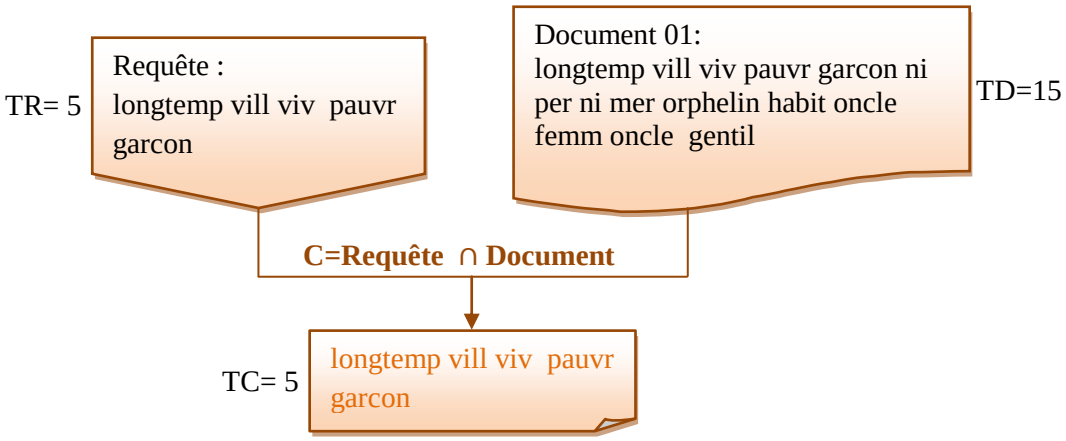
$\frac{C}{R}$: La couverture de la requête par rapport au document

$\frac{C}{D}$: La couverture de document par rapport à la requête

Figure 17 : une représentation sur la technique de couverture

[5] CRD: **C**ouverture **R**equête/**D**ocument

Exemple



La couverture de la requête par rapport au document : **TC/ TR**=1=100%

La couverture de document par rapport à la requête: **TC/TD**=0.33=33%

- **TR:** La taille de la requête.
- **TD:** La taille du document.
- **TC:** La taille des mots communs entre la requête et le

Figure 18 : Exemple sur la technique de couverture

4. Conclusion

Dans ce chapitre, nous avons détaillé les différentes techniques implémentées pour la détection des mots et des phrases reprises dans un ensemble de documents. Les techniques développées sont basées sur une recherche d'information simplifiée. Une comparaison des résultats obtenus par les différentes techniques sera menée et analysée dans le chapitre suivant.

Chapitre IV

Implémentation et résultats

1. introduction

Le rôle crucial de l'évaluation d'une application informatique dans un contexte scientifique, est de mettre en position nos résultats et de montrer la performance et l'efficacité de notre système proposé.

Le but de ce chapitre de mettre l'accent sur la partie implémentation de notre système « PlagZoom », à travers l'environnement de développement, l'interface graphique, la collection de documents d'évaluation et les résultats des tests.

2. Langage de développement

Figure 19 : Icones de langage de développement

Le langage de programmation que nous avons adopté pour implémenter notre application est le python, qui est un langage de programmation objet et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.

le python[1] permet (sans l'imposer) une approche modulaire et orientée objet de la programmation. ce langage est développé depuis 1989 par Guido van Rossum et de nombreux contributeurs bénévoles.

2.1. Pourquoi choisir python

Il faut tout de même savoir, que ce langage s'adapte bien au domaine de l'application à savoir la recherche d'information.

Python est un langage qui peut s'utiliser dans de nombreux contextes et s'accommoder à tout type d'utilisation grâce à des bibliothèques spécialisées **[27]**.

2.2. Caractéristiques du python

les caractéristiques du python sont **[27]:**

- ✓ Etre facile d'utilisation pour les débutants.
- ✓ Etre un langage généraliste.
- ✓ Etre portable, non seulement sur les différentes variantes d'UNiX, mais aussi sur les OS propriétaires: MacOS, BeOS, NeXTStep.
- ✓ Etre gratuit, mais on peut l'utiliser sans restriction dans des projets commerciaux.

[1] Présentation du langage Python disponible sur http://www.linux-center.org/articles/9812/python.html (Accédé le 30/04/2016)

- ✓ Python est (optionnellement) multi-threadé
- ✓ Python intègre, comme Java ou les versions récentes de C++, un système d'exceptions, qui permettent de simplifier considérablement la gestion des erreurs.
- ✓ Fournir des messages d'erreur clairs et conviviaux.
- ✓ Python est un langage qui continue à évoluer, soutenu par une communauté d'utilisateurs et responsables, dont la plupart sont des supporters du logiciel libre. Parallèlement à l'interpréteur principal, écrit en C et maintenu par le créateur du langage, un deuxième interpréteur, écrit en Java, est en cours de développement.

3. L'environnement de développement

Nous avons développé notre système dans des conditions bien spécifiques, la liste suivante indique clairement les exigences de développement :

- ❖ Version du langage de programmation : Python 2.6
- ❖ Caractéristiques de la machine (ordinateur) :
  - ✓ Disque dur : 500Gb ;
  - ✓ Résolution de l'écran : 1366*768.

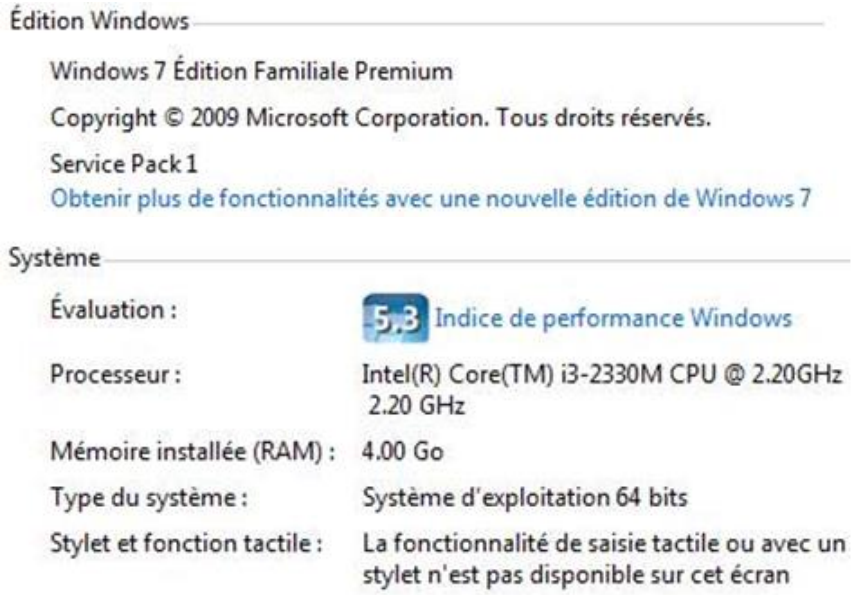


Figure 20 : Capture d'écran décrivant les caractéristiques de la machine

3.1. Base de données « SQLite 3 »

SQLite3[2] est un système de base de données qui a la particularité de fonctionner sans serveur, on dit aussi "standalone" ou "base de données embarquée". On peut l'utiliser avec beaucoup de langages : PHP, Python, C# (.NET), Java, C/C++, Delphi, Ruby...etc.

[2] Définition de SQLite3 disponible sur http://www.finalclap.com/faq/180-sqlite-definition (Accédé le 14/05/2016)

L'intérêt c'est que c'est très léger et rapide à mettre en place, Une base de données SQLite est bien plus performante et facile à utiliser que de stocker les données dans des fichiers XML ou binaires, d'ailleurs ces performances sont même comparables aux autres SGBD fonctionnant avec un serveur comme MySQL, Microsoft SQL Server ou PostgreSQL.

4. Description de l'interface graphique de PlagZoom

Nous présentons dans cette section des capteurs d'écran, qui représentent notre system « PlagZoom ».

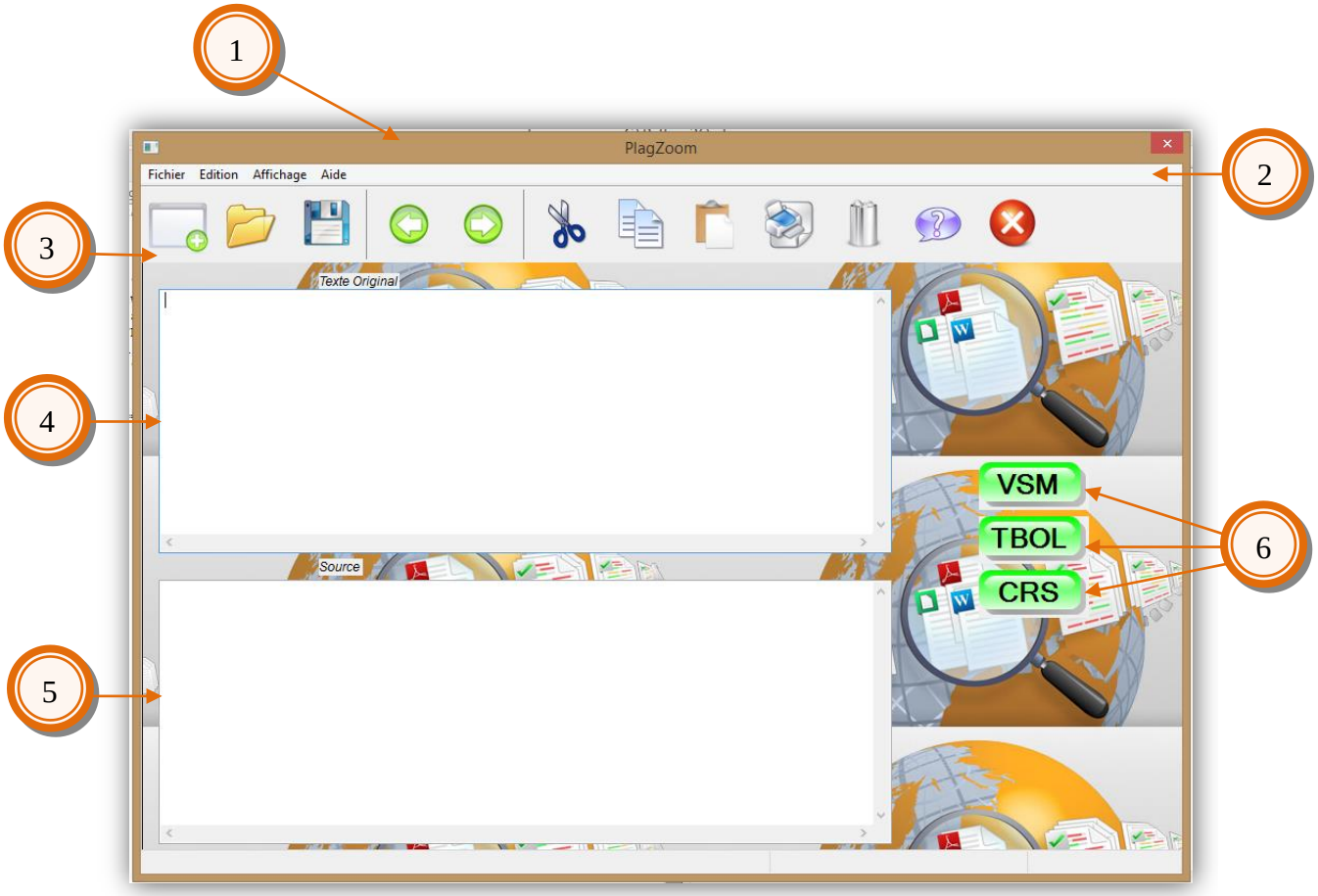


Figure 21 : Capture d'écran de l'interface graphique **« PlagZoom ».**

Numéro	Description
1	Barre de titre de logiciel
2	Barre de menu
3	Barre des outils
4	Zone du texte entré
5	Zone du texte sortie
6	Les boutons de logiciel **« PlagZoom »**

Tableau 03: Description des composants de l'interface graphique **« PlagZoom ».**

4.1. Barre de menu

Figure 22 : Le menu de l'interface graphique **« PlagZoom ».**

4.2. Menu (Fichier)

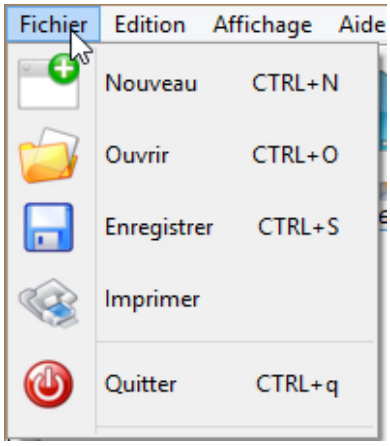


Figure 23 : Les sous-menus de Fichier.

- Nouveau : ouvrir une nouvelle fenêtre.
- Ouvrir : ouvrir un fichier texte.
- Enregistrer : enregistrer comme un fichier texte.
- Imprimer : Imprimer les résultats.
- Quitter : Quitter la fenêtre de l'application.

4.3. Menu (Edition)

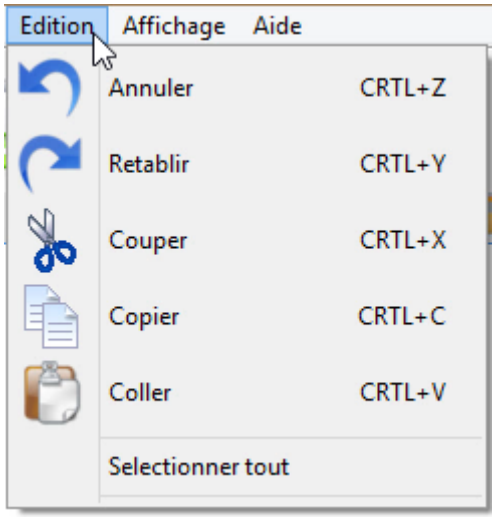
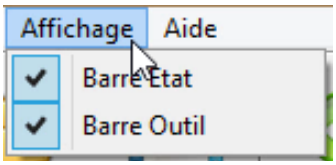


Figure 24 : Les sous-menus d'Edition.

- Annuler: retour au précédent.
- Rétablir : retour à la suivant.
- Couper : couper les résultats.
- Copier : copier les résultats.
- Coller : coller les résultats.
- Sélectionner tout: sélectionner tout le résultat.

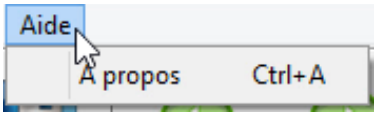
4.4. Menu (Affichage)



- Barre Etat : afficher la barre d'état.
- Barre outil : afficher la barre outil.

Figure 25 : Les sous-menus d'Affichage.

4.5. Menu (Aide)



- A propos : Des informations sur l'équipe de développement de l'application.

Figure 26 : Les sous-menus d'Aide.

4.6. Barre d'outils

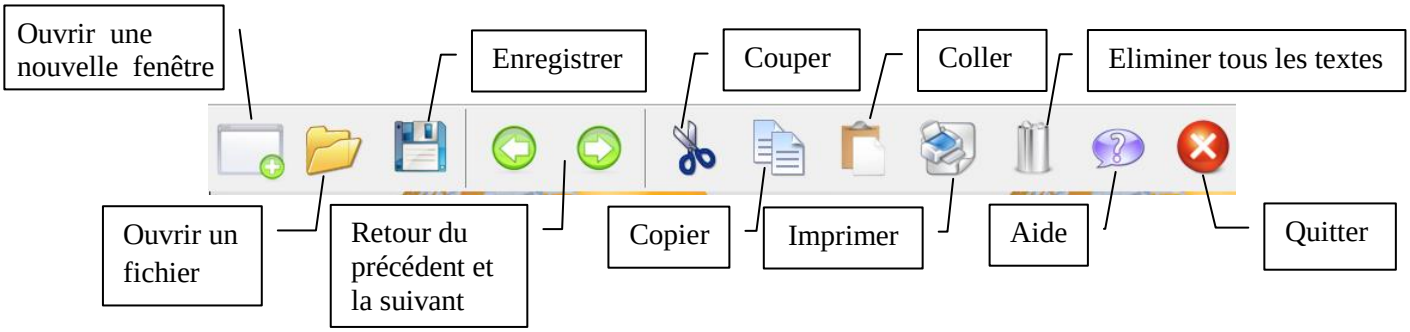


Figure 27: Les boutons de raccourses.

4.6. Les boutons du système «PlagZoom»

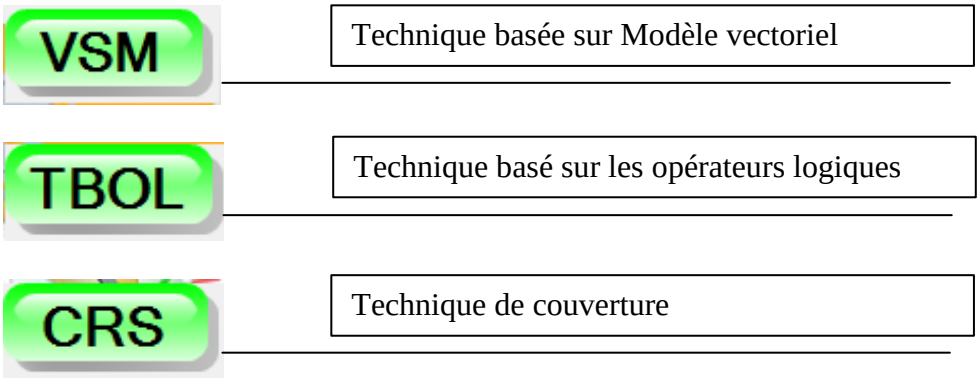


Figure 28 : Les boutons du système «PlagZoom»

4.7. Exemples sur la fonction du bouton « VSM »

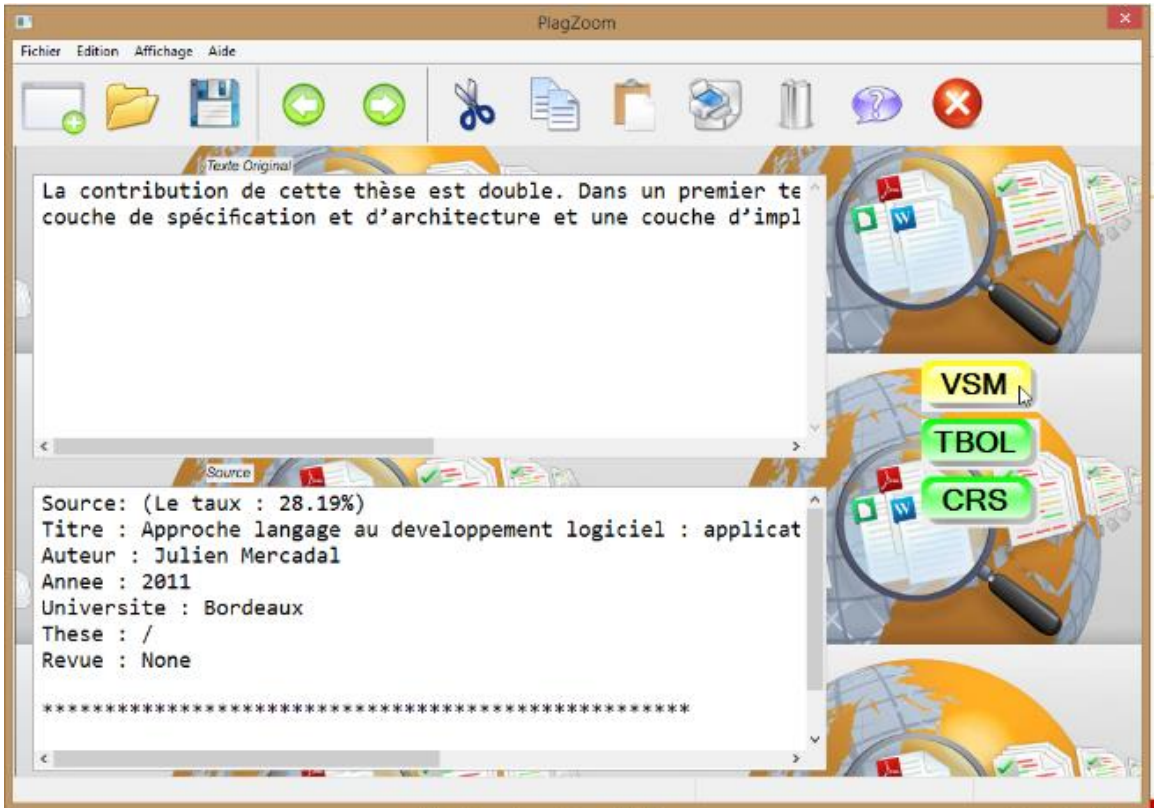


Figure 29: Le résultat du bouton « VSM » .

4.8. Exemples sur la fonction du bouton « TBOL »

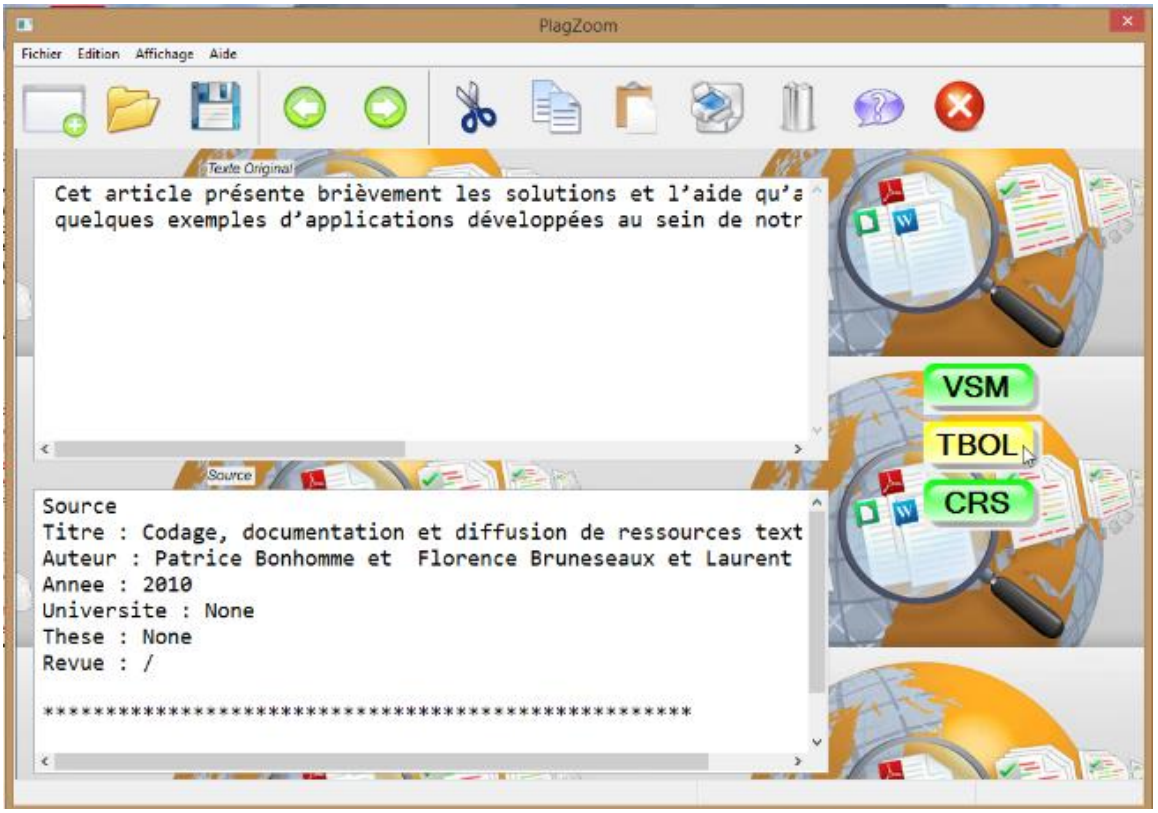


Figure 30: Le résultat du bouton « TBOL » .

4.9. Exemples sur la fonction du bouton « CRS »

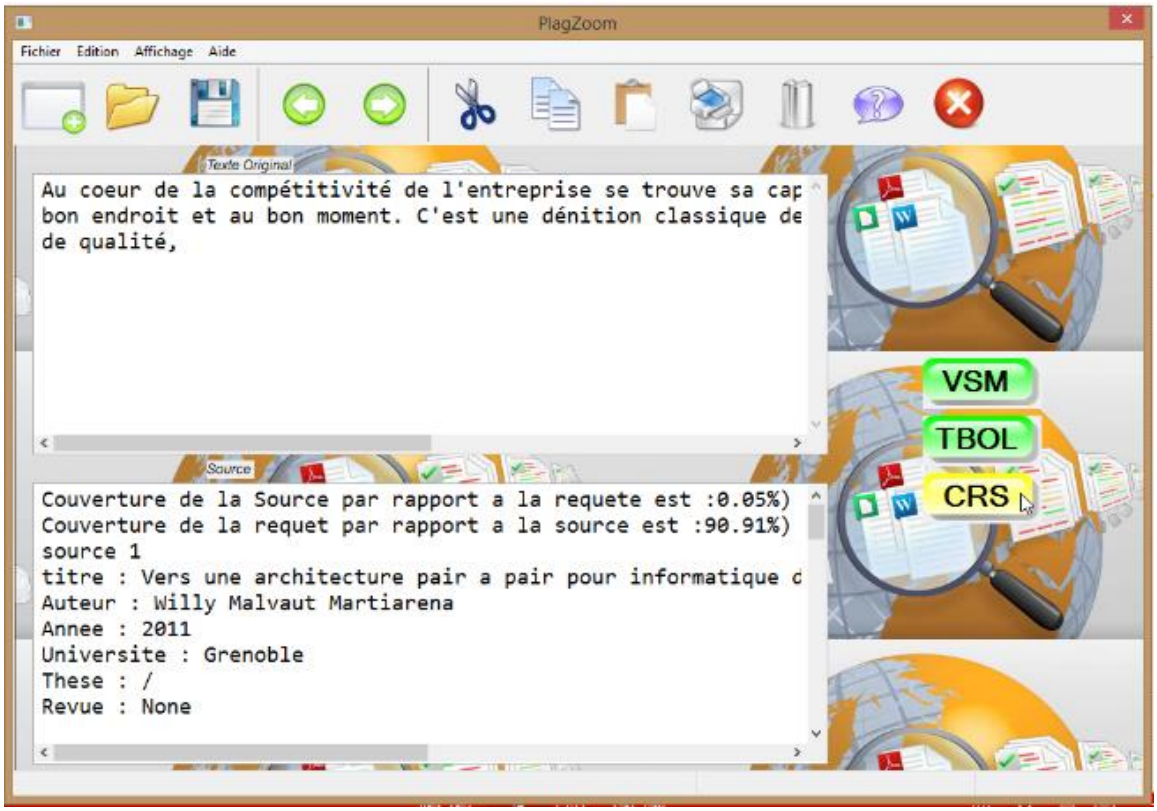


Figure 31: Le résultat du bouton « CRS » .

5. Evaluation

L'´evaluation constitue une étape importante lors de la mise en œuvre d'un modèle de recherche d'information puisqu'elle permet de paramétrer le modèle, d'estimer l'impact de chacune de ses caractéristiques et enfin de fournir des éléments de comparaison entre modèles.

5.1. Mesures d'évaluation

L'évaluation nécessite la définition d'un ensemble de mesures et de méthodes, ainsi que de collections de test assurant l'objectivité de l'évaluation. Nous présentons dans ce qui suit les deux principales mesures d'´evaluation : le rappel et la précision.

- ✓ **Rappel**

Le rappel mesure la proportion des documents pertinents restitués par le système relativement à l'ensemble des documents pertinents contenus dans la base documentaire. Elle mesure la capacité du système à retrouver tous les documents pertinents répondant à une requête **[28]**.

$$\textbf{Rappel} = \frac{\text{nombre total de documents pertinents retrouvé s par le systéme}}{\text{nombre total de documents pertinents dans la collection}} \quad (35)$$

- ✓ **Précision**

La précision mesure la proportion de documents pertinents relativement à l'ensemble des documents restitués par le système. Elle mesure la capacité du système à rejeter tous les documents non pertinents à une requête donnée par le rapport entre l'ensemble des documents sélectionnés pertinents et l'ensemble des documents sélectionnés **[28]**.

$$\textbf{Précision} = \frac{\text{nombre total de documents pertinents trouvés par le systéme}}{\text{nombre total de documents retrouvés par le systéme}} \quad (36)$$

Nous avons sélectionné une collection de 450 documents, et nous avons effectué 450 requêtes pour établir des tests dans le but de comparer les résultats entre les techniques implémentaient : VSM, TBOL, et CRS. Nous avons évalué les résultats interprétées par la métrique Rappel (R) et Précision (P) en prenant en trois (03) cas de requête pour chaque technique.

Tel que les trois requêtes sont:

Soit la requête *« Q $_1$»*

{Le couplage fort permet d'assurer automatiquement et systématiquement que l'implémentation du système est conforme aux descriptions, On est donc encore loin de toutes les possibilités ofertes par l'utilisation d'interfaces classiques comme la souris. D'où la nécessité d'étendre et d'enrichir les fonctionnalités accessibles, nous vollons montrer}.

Soit la requête *« Q $_2$»*

{La recherche se tourne désormais vers la formalisation, le but étant de représenter les connaissances, de modéliser l'analyse de la langue. Ainsi seront développés, dans les années 1970 et 1980, Dans ce qui suit, nous nous intéresserons donc uniquement aux méthodes combinatoires, c'est-à- dire basées sur l'étude d'ensemble fnis d'objets mathématiques. & D et al., 2008), ou des réalisations , 1992 ; « block de composion Cette approche peut se justifer par » dans des optimales (c'est-à-dire de la recherche d'un graphe plusieurs arguments. L'apprentissage automatique (ou apprentissage artificiel) est, suivant la définition de Tom Mitchell dans (Mitchell, 1997), l'étude des algorithmes qui permettent aux programmes de s'améliorer automatiquement par expérience.}.

Soit la requête *« Q $_3$»*

{Depuis l'avènement de la loi "Informatique et Libertés" en 1978, l'image de la protection des données personnelles auprès du public et de l'industrie a beaucoup évolué..}.

5.2. Résultats Pour La Technique VSM

Pour la requête ***« Q $_1$»***

Soit **N** = $\{D_1, D_{14}\}$ l'ensemble des documents pertinentes.

Les documents pertinents sont marqués par la lettre "P" comme indiqué dans tableau suivant:

Document renvoyer par le système	Pertinentes	Rappel	précision
D_1	P	0.5	1
D_2		0.5	0.5
D_3	P	1	0.6
D_4		1	0.5
D_5		1	0.4

Tableau 04: les résultats de la requête *« Q $_1$»* dans VSM.

Pour la requête *« Q $_2$»*

Soit **N** =$\{D_{10}, D_6, D_3\}$ l'ensemble des documents pertinentes.

Document renvoyer par le système	Pertinentes	Rappel	précision
D_1	P	0.3	1
D_2	p	0.6	1
D_3		0.6	0.66
D_4	p	1	0.75

Tableau 05: les résultats de la requête *« Q $_2$»* dans VSM.

Pour la requête *« Q $_3$»*

Soit **N** = $\{D_{17}\}$ l'ensemble des documents pertinentes.

Document renvoyer par le système	Pertinentes	Rappel	précision
D_1	p	1	1
D_2		1	0.5
D_3		1	0.3

Tableau 06: les résultats de la requête *« Q $_3$»* dans VSM.

La figure (32) présente, la moyenne de rappel et précision pour les trois requêtes.

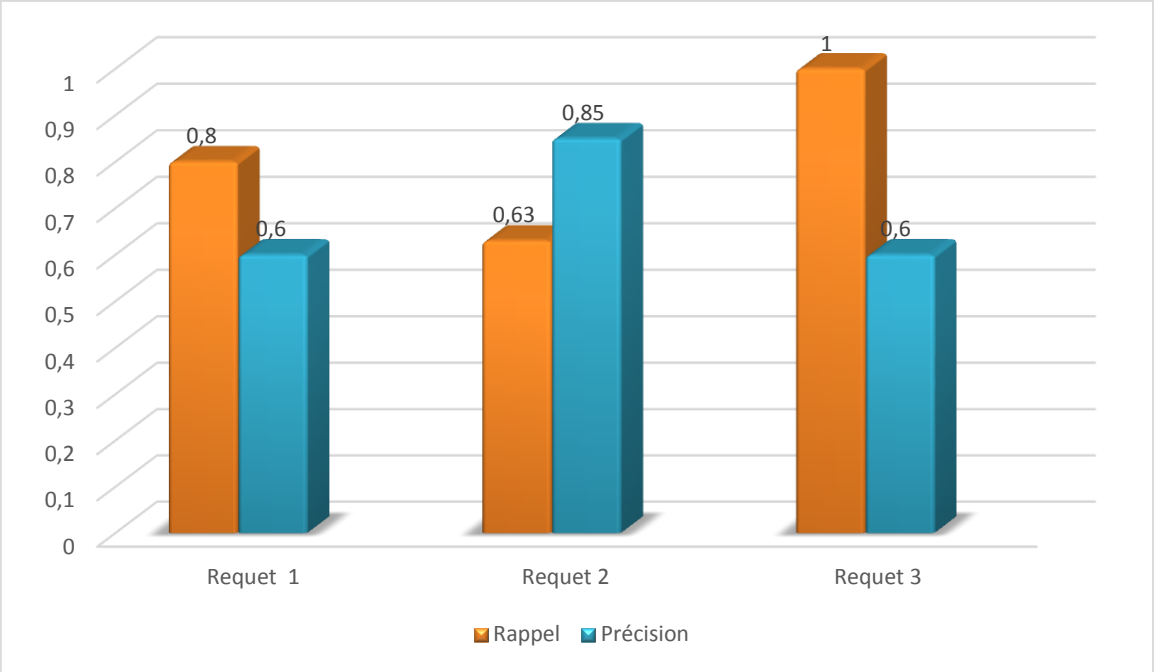


Figure 32: la moyenne de rappel et précision pour les trois requêtes dans la technique VSM

D'après les résultats de ces évaluations, nous constatons que les valeurs du rappel sont comprises entre 0,63 et 1 et que celles de la précision sont comprises entre 0,6 et 0,85. Ce qui donne une valeur moyenne pour le rappel de 0,81 et 0,68 pour la précision.

5.3. Résultats Pour La Technique TBOL

Pour la requête *« Q $_1$»*

Soit **N** ={D_1, D_{14}} l'ensemble des documents pertinentes.

Document renvoyer par le système	Pertinentes	Rappel	Précision
D_1	p	0.5	1
D_2		0.5	0.5
D_3		0.5	0.3
D_4		0.5	0.25
D_5		0.5	0.2
D_6		0.5	0.16
D_7		0.5	0.14
D_8		0.5	0.12
D_9		0.5	0.11

Tableau 07 A: les résultats de la requête *« Q $_1$»* dans TBOL.

Document renvoyer par le système	Pertinentes	Rappel	Précision
D_{10}		0.5	0.1
D_{11}		0.5	0.09
D_{12}		0.5	0.08
D_{13}	p	1	0.15

Tableau 07 B: les résultats de la requête *« Q $_1$»* dans TBOL.

Pour la requête *« Q $_2$»*

Soit **N** = $\{D_{10}, D_6, D_3\}$ l'ensemble des documents pertinentes.

Document renvoyer par le système	Pertinentes	Rappel	Précision
D_1		0	0
D_2		0	0
D_3	p	0.3	0.3
D_4		0.3	0.25
D_5		0.3	0.2
D_6		0.3	0.16
D_7		0.3	0.14
D_8		0.3	0.12
D_9		0.3	0.11
D_{10}		0.3	0.1
D_{11}	p	0.6	0.18
D_{12}		0.6	0.16
D_{13}		0.6	0.15
D_{14}		0.6	0.14
D_{15}	p	1	0.2

Tableau 08: les résultats de la requête *« Q $_2$»* dans TBOL.

Pour la requête *« Q $_3$»*

Soit **N** = $\{D_{17}\}$ l'ensemble des documents pertinentes.

Document renvoyer par le système	Pertinentes	Rappel	Précision
D_1		0	0
D_2		0	0
D_3	p	1	0.3

Tableau 09: les résultats de la requête « **Q** $_3$» dans TBOL.

La figure (33) présente, la moyenne de rappel et précision pour les trois requêtes.

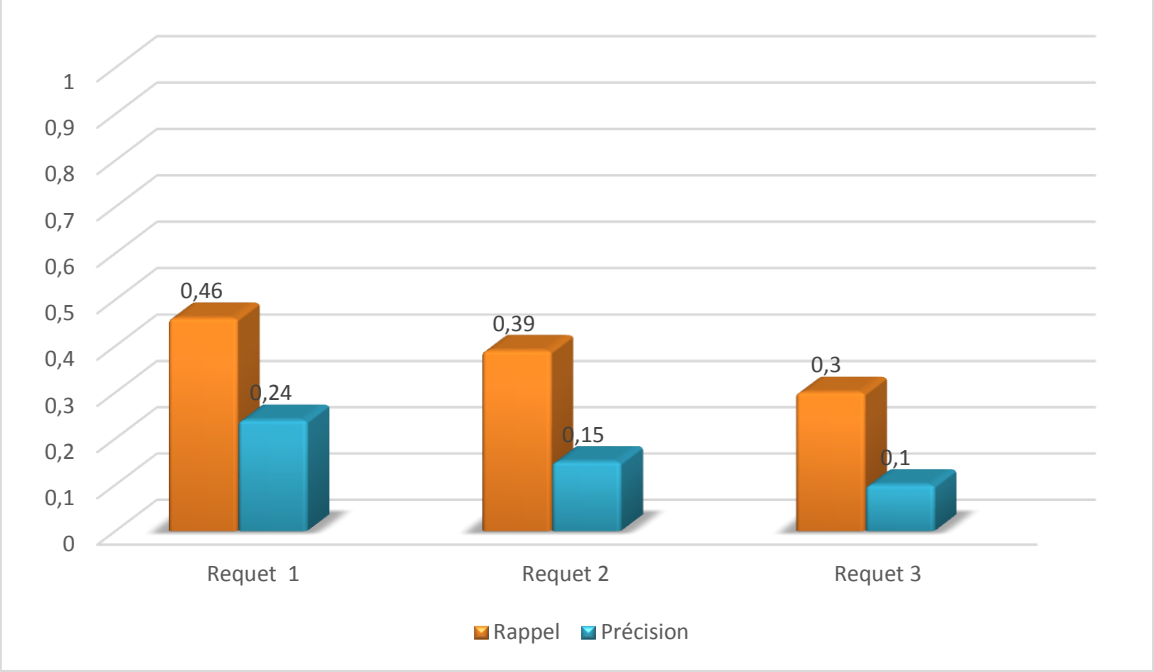


Figure 33: la moyenne de rappel et précision pour les trois requêtes dans la technique TBOL

D'après les résultats de ces évaluations, nous constatons que les valeurs du rappel sont comprises entre 0, 3 et 0.46 et que celles de la précision sont comprises entre 0,1 et 0,24. Ce qui donne une valeur moyenne pour le rappel de 0,4 et 0,2 pour la précision.

5.4. Résultats Pour La Technique CRS

Pour la requête « **Q** $_1$»

Soit **N** = $\{D_1, D_{14}\}$ l'ensemble des documents pertinentes.

Document renvoyer par le système	Pertinentes	Rappel	Précision
D_1	p	0.5	1
D_2		0.5	0.5
D_3		0.5	0.3

Tableau 10 A: les résultats de la requête « **Q** $_1$» dans CRS.

Document renvoyer par le système	Pertinentes	Rappel	Précision
D_4		0.5	0.25
D_5		0.5	0.2
D_6		0.5	0.16
D_7		0.5	0.14

Tableau 10 B: les résultats de la requête *« Q $_1$»* dans CRS.

Pour la requête *« Q $_2$»*

Soit **N** = $\{D_{10}, D_6, D_3\}$ l'ensemble des documents pertinentes.

Document renvoyer par le système	Pertinentes	Rappel	Précision
D_1		0	0
D_2		0	0
D_3		0	0
D_4		0	0
D_5		0	0
D_6		0	0.
D_7	p	0.3	0.14
D_8	p	0.7	0.25
D_9		0.	0.22

Tableau 11: les résultats de la requête *« Q $_2$»* dans CRS.

Pour la requête *« Q $_3$»*

Soit **N** =$\{D_{17}\}$ l'ensemble des documents pertinentes.

Document renvoyer par le système	Pertinentes	Rappel	Précision
D_1		0	0
D_2	p	1	0.5
D_3		1	0.3
D_4		1	0.25

Tableau 12: les résultats de la requête *« Q $_3$»* dans CRS.

La figure (34) présente, la moyenne de rappel et précision pour les trois requêtes.

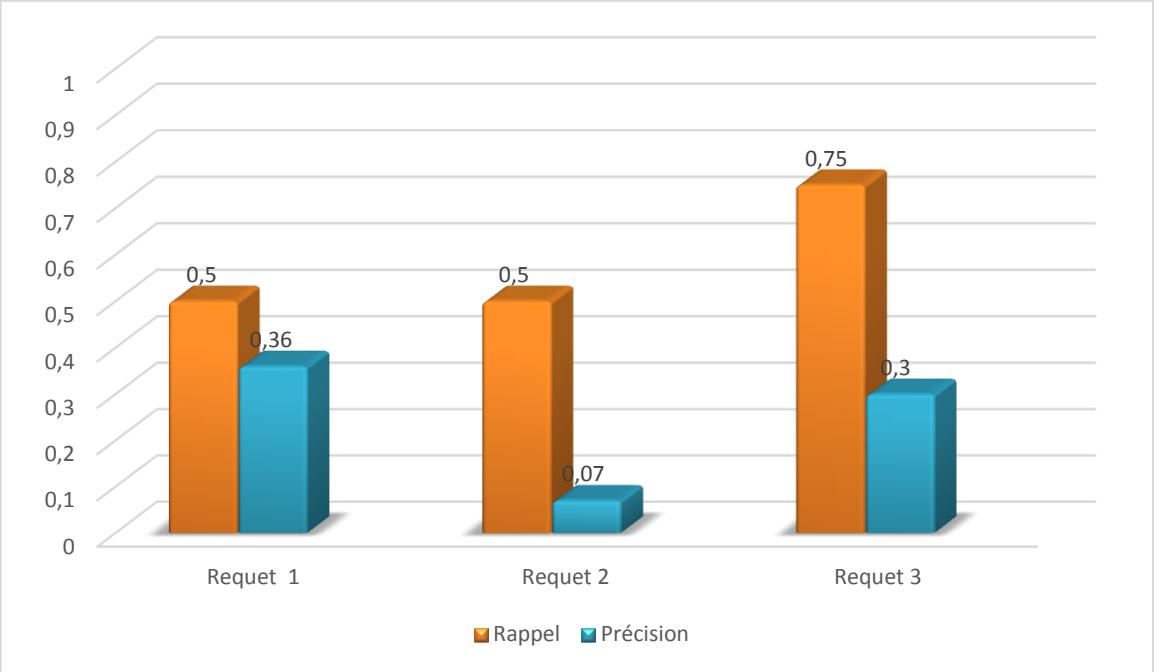


Figure 34: la moyenne de rappel et précision pour les trois requêtes dans la technique CRS

D'après les résultats de ces évaluations, nous constatons que les valeurs du rappel sont comprises entre 0, 5 et 0.75 et que celles de la précision sont comprises entre 0,07 et 0,36. Ce qui donne une valeur moyenne pour le rappel de 0,6 et 0,25 pour la précision.

6. Analyse critique

La figure (35) présente, la comparaison des trois techniques implémentaient en utilisant les mesures rappel et précision.

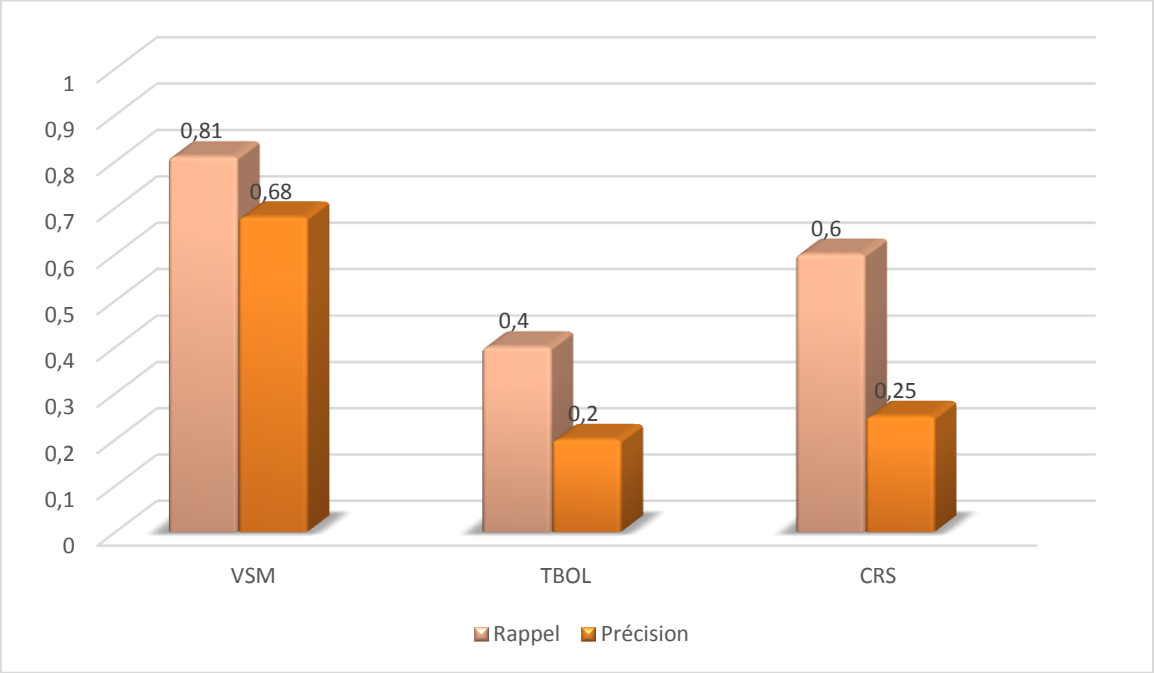


Figure 35: la comparaison des trois techniques implémentaient en utilisant les mesures rappel et précision.

La figure (35) présente la comparaison des trois techniques implémentaient en utilisant les mesures Rappel-Précision. D'après les résultats de ces évaluations, nous constatons que les valeurs Rappel-Précision liées à la technique VSM sont très élevé par rapport aux deux autre techniques (TBOL et CRS). Cela est dû en grande partie à la simplicité conceptuelle et mise en œuvre de la technique VSM. Il permet également la pondération des termes, ce qui augmente les performances du système.

La technique CRS retourne des résultats satisfaisantes car elle permet de donner un double apport. D'une part, le pourcentage de l'existence de la requête dans le document et d'autre part le pourcentage de la présence du document dans la requête.

Concernant la technique TBOL nous remarquant qu'elle est transparente et simple à comprendre pour l'utilisateur.

Du point de vue de notre méthode d'évaluation, ces techniques donnent des performances très acceptables pour notre système «PlagZoom» et montrent qu'elle sélectionne des réponses très proches de celles sélectionnées par un sujet humain avec un gain de temps considérable.

7. Conclusion

Dans ce chapitre, nous avons présenté l'implémentation de notre système « PlagZoom». Ensuite, nous avons évalué les trois techniques proposés (VSM, TBOL et CRS) en utilisant les mesures rappel et précision pour tester la performance de notre outil.

Les expériences montrent que les résultats sont bon pour les différents techniques. Cependant, ils manquent un peu d'optimisation pour obtenir des résultats plus performants à l'avenir.

Conclusion Générale

Bilan et Perspectives

1. Bilan et Perspectives

Depuis l'apparition de l'informatique jusqu'à aujourd'hui le nombre des utilisateurs de cette technologie est en augmentation continu ce qui implique l'accroissement du besoin des applications pour la recherche d'information.

Le but de ce mémoire est de contribuer à ce développement en ajoutant une pierre à l'édifice, par la réalisation d'un outil fondamental et nécessaire à n'importe quelle application en RI, à savoir la détection de plagiat.

Plagzoom est un détecteur de plagiat dans un document. L'originalité de ce travail réside dans le fait d'adopter trois techniques de recherche basées sur des approches formelles (booléen, vectoriel, probabiliste), pour détecter la similarité entre une requête et un document (à l'échelle du corpus). En effet, il est tout à fait possible que seule une partie d'un document soit plagiée, le reste étant un travail original du fraudeur.

Au final, nous avons obtenu de bons résultats au niveau de VSM, TBOL et CRS comme la montré l'évaluation de rappel et précision effectuée sur notre système. C'est pourquoi, on peut conclure que nous avons atteint notre objectif initial à savoir offrir aux chercheurs un système performant.

Comme perspectives, on peut recenser quelques points qui peuvent améliorés la qualité de notre outil PlagZoom, tel que:

- Enrichir la base de données en ajoutant d'autres documents.
- L'utilisation de d'autres formules de pondération en complément à la formule utilisée (tf-idf).
- Améliorer le temps d'exécution d'une requête, c'est-à-dire rendre le système plus rapide pendant l'opération de la recherche.
- développer notre outil pour des recherches online (connecté à Internet).
- Étendre notre système pour un fonctionnement sur d'autres types de document (.pdf, .xml, .docx, ...etc).

Références

Références

[1] Jian-Yun Nie, « Le domaine de recherche d'information – Un survol d'une longue histoire», Département d'informatique et recherche opérationnelle Université de Montréal ;

[2] Herzallah Abdelkarim, « recherche d'information», support de cours, Université Bouira (2014).

[3] Gerard Salton, « Search and retrieval experiments in real-time information retrieval ». In IFIP Congress (2), pages 1082–1093, (1968).

[4] C. J. Van Rijsbergen, « Information Retrieval». Butterworth-Heinemann, Newton, MA, USA, 2nd edition, (1979).

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

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

[7] bouramoul abdelkrim. « Recherche d'information contextuelle et sémantique sur le web», thèse de doctorat en informatique, université mentouri de constantine. (2011).

[8] Ihab Mallak. «De nouveaux facteurs pour l'exploitation de la sémantique d'un texte en Recherche d'Information», thèse de doctorat en informatique, Université Toulouse -Paul Sabatier. (2011).

[9] M.Hammache Arezki, « Recherche d'information: un modèle de langue combinant mots simples et mots composé», Thèse de Doctorat en informatique, Université Mouloud Mammeri de Tizi-Ouzou.

[10] Karbasi Soheila , «Pondération des termes en Recherche d'Information: Modèle de pondération basé sur le rang des termes dans les documents», thèse de doctorat en informatique, Université Toulouse -Paul Sabatier. (2007).

[11] ABBAS Nacira, «Vers une Extension Sémantique de l'Analyse Formelle de Concepts : Application à la Recherche d'Informations», mémoire de magister en informatique, université mouloud Mammeri, Tizi-Ouzou. (2014).

[12] Boughareb Djalila, « Recherche d'information multicritères», thèse de doctorat en informatique, Université Badji Mokhtar –Annaba. (2014).

[13] Saidi Ilham Et Hamsi Essma, «Réalisation d'un moteur de recherche**»,** mémoire de Licence en Informatique, université abou bakr belkaid– tlemcen. (2013).

[14] Mataoui M'hamed, «Reformulation de requêtes dans les systèmes de recherche d'information

dans des documents XML**»**, mémoire de magister en informatique, Université M'hamed bougara de boumerdes (2007).

[15] Mammeri Karima, « Recherche d'information par croisement de média texte et image », mémoire de magister en informatique, Université M'hamed bougara de boumerdes (2009).

[16] Moussaoui Kamal Et Feredj Dhia Elhak, « Conception et développement d'un Outil de recherche sur le web à base d'agent**»,** mémoire de master académique en informatique, universite kasdi merbah Ouargla. (2013).

[17] Lamraoui Younes, «Recherche intelligente des informations dans le coran **»,** Mémoire d'ingénieur en Informatique, Université Abou Bakr Belkaid– Tlemcen. (2011).

[18] Mehidi tawfiq et Rabah zakarya, « conception et implémentation d'un système de recherche à base d'annotations sociales**»**, mémoire de Master en informatique ,Université Abou Bakr Belkaid– Tlemcen. (2014)

[19] Bordogna et al., « Flexible Querying of Structured Documents**»**. Proceedings of the Fourth International Conference on Flexible Query Answering Systems (FQAS), (2000).

[20] Ben Aouicha Mohamed, « Une approche algébrique pour la recherche d'information structurée **»,** thèse de doctorat en informatique, Université Toulouse -Paul Sabatier. (2009).

[21] Azzoug Wassila, «Contribution à la définition d'une approche d'indexation sémantique de documents textuels **»,** mémoire de magister en informatique, Université M'hamed bougara de boumerdes (2012).

[22] Boubekeur Fatiha, « Contribution à la définition de modèles de recherche d'information flexibles basés sur les CP-Nets **»**, thèse de doctorat en informatique, Université Toulouse - Paul Sabatier. (2008).

[23] Hlaoua Lobna, « Reformulation de Requêtes par Réinjection de Pertinence dans les Documents Semi-Structurés », Thèse de doctorat à l'Université Paul Sabatier, (2007).

[24] Nawel Nassr, « Croisement de langues en recherche d'information : traduction et désambiguïsation de requêtes », Thèse de Doctorat à l'Université Paul Sabatier, (2002).

[25] Bouabane Samia Et Benghelima Mohamed Amine, « Réalisation d'un système de recherche d'information», Mémoire de Licence en Informatique, Université Abou Bakr Belkaid– Tlemcen. (2014)

[26] Salton, G., E.A. Fox, H. Wu. « Extended Boolean information retrieval system». *CACM* 26(11), pp. 1022-1036, (1983).

[27] Gérard Swinnen, «Apprendre à programmer avec python 3 », livre, pages 7-8, (2012)