RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOHAMED KHIDER DE BISKRA
FACULTÉ DES SCIENCES EXACTES ET SCIENCES DE LA NATURE ET DE LA VIE

DÉPARTEMENT D'INFORMATIQUE

N°d'ordre :…….
N° de Série :…….

THÈSE

EN VUE DE L'OBTENTION DU DOCTORAT EN SCIENCES
SPÉCIALITÉ : INFORMATIQUE

Présentée par:

ABDELLI BELKACEM

Titre

UNE APPROCHE SÉMANTIQUE POUR LES DOCUMENTS NUMÉRIQUES

Année 2016

Jury **:**

M. Mahmoud BOUFAIDA	Professeur à l'université de Constantine	Président
M. Okba KAZAR	Professeur à l'université de Biskra	Rapporteur
M. Youssef AMGHAR	Professeur à INSA de Lyon	Examinateur
M. Noureddine ZERHOUNI	Professeur à ENSMM Besançon	Examinateur
M. Labib Sadek TERRISSA	Maître de conférences A, à l'université de Biskra.	Examinateur
M. Saber BENHARZALLAH	Maître de conférences A, à l'université de Biskra.	Examinateur
M. Jean-Marie PINOM	Professeur à INSA de Lyon	Co-encadreur

Remerciement

Tout d’abord, je tiens à exprimer ma gratitude à mon directeur de thèse M. Okba KAZAR pour avoir dirigé ce travail. Je remercie pour sa patience, et son encouragement qui m'ont permis de finir cette thèse. Je le remercie également, parce qu'il m'a permis de bénéficier de ces relations avec les enseignants et les chercheurs du Laboratoire LIRIS d'INSA de Lyon.

Mes plus vifs remerciements vont aussi, à Mon codirecteur de thèse, M. Jean-Marie PINON, qui m'a ouvert les portes du laboratoire LIRIS, et il a facilité mon installation, pendant les 18 mois de mon stage. Je lui exprime ma gratitude pour la confiance qu’il m’a accordée, pour son encadrement continu, et pour ces précieux conseils qu’il m’a données.

Je remercie les membres de laboratoires LIRIS que j’ai pu rencontrer durant la période de mon stage, et qui m'ont aidé dans mon travail.

Je suis très reconnaissant au professeur Mahmoud BOUFAIDA d'avoir accepté la tache de président du jury. Je le remercie aussi pour son déplacement de Constantine afin d'être présent dans le jury.

J'exprime mes sincères remerciements aux professeurs Youssef AMGHAR et Noureddine ZERHOUNI qui m'ont honoré par leur présence comme des examinateurs de ce travail, leurs remarques pertinentes m'ont permis d'améliorer la thèse.

Un grand merci va également à mes collègues de département d'informatique de Biskra les docteurs Labib Sadek TERRISSA et Saber BENHARZALLAH qui ont pris la peine de lire et évaluer ce mémoire.

Dédicace.

Je dédie ce modeste travail à toute ma famille, et mes amis, et à tous ceux, de près ou de loin, m'ont aidé.

Résumé.

L'immense volume de documents numériques disponibles, est devenu une problématique pour l'organisation automatique de ces documents afin de faciliter l'interrogation et l'accès à l'information pertinente.

La plupart de ces documents n'ont aucune structuration. Ou bien ils ont une structuration (physique et logique), mais difficile à l'identifier et l'exploiter, ce qui rend pénible la récupération des informations pertinentes à partir de ces documents.

Dans notre thèse on s'intéresse à la modélisation structuro-sémantique de la représentation des documents et la mettre dans un format interprétable et exploitable efficacement par les algorithmes de recherche d'information. Afin de retourner les fragments de documents les plus pertinents.

Dans notre travail, nous modélisons les collections de documents homogène avec un contenu textuel en langue naturelle, comme les publications scientifiques, les articles et les thèses en format PDF ou Word, et les documents web comme les pages de Wikipédia en format XML.

Mot clés : Documents numériques, web sémantique, ontologies, similarité sémantique, structure de documents, indexation, recherche d'information.

Abstract

The huge volume of digital documents available, has become a problem for the process of interrogation and access to relevant information.

Most of these documents have a structure (physical and logical), but it is difficult to be identified and used, making painful retrieving relevant parts of information.

In our case we are interested in the structural-semantic modeling of documents representation and put them in a readable format to be effectively usable by information search algorithms.

Our objective in this work is to return the fragment of the most relevant documents. We model the homogeneous collections of documents, such as scientific publications, articles and theses in PDF or Word format, and web documents such as Wikipedia pages in XML format.

Key words: Digital documents, semantic Web, ontologies, semantic similarity, structure of documents, indexing, information retrieval

ملخص :

أصبح الكم الهائل من الوثائق الرقمية المتاحة، مشكلة أثناء عملية البحث، مما صعب الوصول إلى المعلومات القيمة.

معظم هذه الوثائق لديها هيكل (مادي ومنطقي)، ولكن من الصعب تحديده واستغلاله، مما يجعل مهمة الباحث عن المعلومات المهمة عملا شاقا.

في عملنا هذا نحن مهتمون بتمثيل النموذج البنيوي والدلالي للوثائق ووضعه في شكل سهل وقابل للقراءة بشكل فعال من طرف الآلة لتسهيل عملية البحث عن المعلومات.

هدفنا في هذا العمل هو البحث بطريقة ذكية و فعالة عن المستندات التي تنطبق و طلب المستخدم وإرجاع أجزاء الوثائق الأكثر أهمية. نستغل في تنفيذ عملنا مجموعات متجانسة من الوثائق الالكترونية، مثل المنشورات والمقالات العلمية وأطروحات التخرج بتنسيق PDF أو Word وكذلك الوثائق الموجودة على شبكة الإنترنت مثل صفحات ويكيبيديا في شكل XML.

الكلمات الرئيسية :وثائق الكترونية، الويب الدلالي، تشابه الدلالي، بنية الوثائق، فهرسة الوثائق ، البحث عن المعلومات

Table des matières

Introduction générale

1. Contexte

La quantité d'informations disponibles sous forme numérique est en augmentation exponentielle et draconienne grâce à l'avancement des technologies d'acquisition numérique de documents. Ainsi que le développement de la publication sur internet et les moteurs de recherche ont rendu l'accès à de grandes quantités de documents très facile.

Cependant, cet immense volume de documents disponibles pour les personnes, est devenu une problématique pour l'organisation automatique de ces documents d'une manière efficace afin de faciliter l'interrogation et l'accès à l'information pertinente. Les systèmes documentaires retournent généralement un grand nombre de documents en réponse aux requêtes des utilisateurs, mais une grande partie de ces résultats n'est pas utile et ne correspond pas aux besoins des utilisateurs.

Il devient de plus en plus difficile pour les utilisateurs de localiser les documents pertinents à l'égard de leur besoin formulé par des requêtes. Ce qui a amené les chercheurs dans ce domaine a proposé plusieurs solutions sur l'organisation des documents afin d'améliorer la précision et l'efficacité de l'accès à l'information et de réduire le temps de recherche.

Un document numérique peut être tout type de médias qui contient des données. Un document peut être un morceau de texte, une page Web, une image, une séquence vidéo, etc. Le type le plus souvent utilisé est le texte.

Les documents textuels ont différents formats tels que HTML, PDF et Word. La plupart de ces documents ou bien ils n'ont aucune structuration (appelés documents non structurés), Ou bien ils ont une structuration (physique et logique), mais vu leur format, il est difficile de les identifier et les exploiter, ce qui rend pénible la récupération des éléments et des parties pertinentes à partir de ces documents.

Dans notre thèse on s'intéresse aux collections de documents homogène avec un contenu textuel en langue naturelle, comme les publications scientifiques, les articles et les thèses en format PDF ou Word, et les documents web comme les pages de Wikipédia en format XML.

Récemment, il y a eu une augmentation rapide du nombre de documents semi-structurés de type XML, et commencent à avoir une place importante dans le web. Ce type de format permet de modéliser la structure logique de documents en utilisant des balises.

Les systèmes de traitement de documents comprennent généralement deux parties principales; la première partie consiste à modéliser d'une manière automatique la représentation de documents et la mettre dans un format lisible et interprétable par les machines. Cette partie est appelée indexation dans les systèmes de recherche de documents. L'indexation permet d'analyser syntaxiquement le contenu textuel des documents puis elle met les termes sous forme d'un vecteur.

La deuxième partie représente l'algorithme qui permet d'exploiter efficacement la représentation du document. Parmi ces algorithmes : Les algorithmes de recherches, les algorithmes de classifications, les algorithmes de clustering (regroupement), les algorithmes pour le résumé automatique... etc.

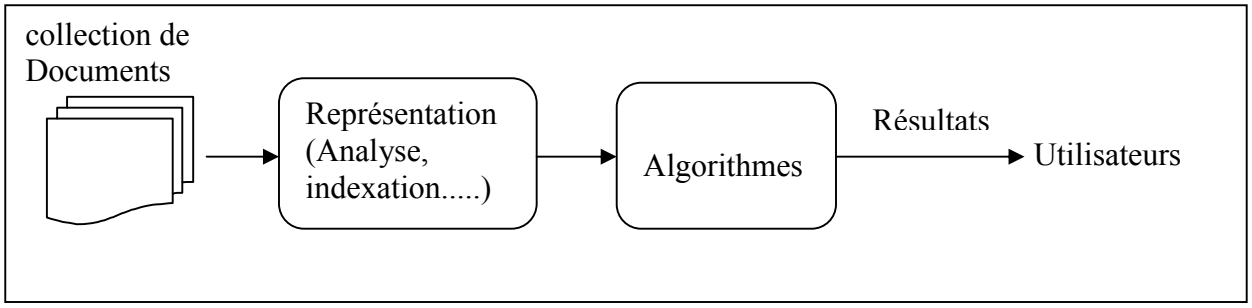


Figure 1 : Etapes de traitement de documents.

Pour évaluer nos contributions, Nous avons choisi, dans notre travail, le domaine de recherche d'information.

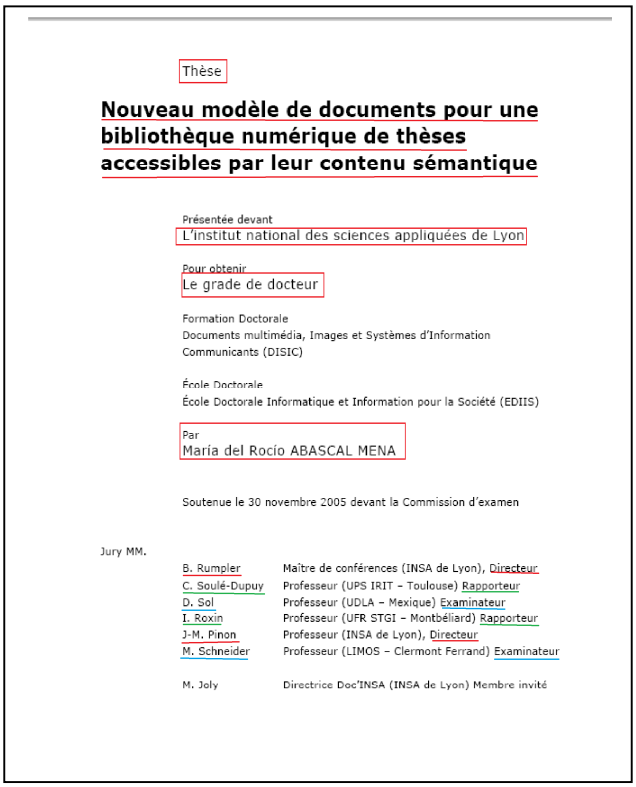
2. Problématique

Les systèmes de traitement de documents traditionnels examinent un document comme un texte brut. Cependant, la forme de document devient de plus en plus structurée, où la structure logique de documents a pris une place importante chez les auteurs. Les systèmes de rédaction de documents permettent aux auteurs de mentionner la structure logique et les parties importantes de documents tels que : Le titre, l'auteur, la date de création, la hiérarchie de sections (section, sous-section,…), Les titres de sections, paragraphes, résumé, ...etc.

Cette information structurelle peut faciliter la représentation et l'amélioration de l'efficacité des systèmes de traitement de documents (indexation, recherche...). Et si cette structure est ignoré une partie importante de l'information sur les documents, va être perdue.

L'analyse structurelle de documents numériques devient de plus en plus utile, elle permet de récupérer les structures physiques et logiques de ces documents, ce qui améliore l'indexation et la récupération.

Le langage XML est devenu le standard le plus utilisé pour la représentation de la structure logique des documents. Une partie de cette structure est appelée métadonnées (comme le titre, la date, auteur,...).



Thèse

Nouveau modèle de documents pour une bibliothèque numérique de thèses accessibles par leur contenu sémantique

Présentée devant
L'institut national des sciences appliquées de Lyon

Pour obtenir
Le grade de docteur

Formation Doctorale
Documents multimédia, Images et Systèmes d'Information Communicants (DISIC)

École Doctorale
École Doctorale Informatique et Information pour la Société (EDIIS)

Par
María del Rocío ABASCAL MENA

Soutenue le 30 novembre 2005 devant la Commission d'examen

Jury MM.

B. Rumpler	Maître de conférences (INSA de Lyon), Directeur
C. Soulé-Dupuy	Professeur (UPS IRIT – Toulouse) Rapporteur
D. Sol	Professeur (UDLA – Mexique) Examinateur
I. Roxin	Professeur (UFR STGI – Montbéliard) Rapporteur
J-M. Pinon	Professeur (INSA de Lyon), Directeur
M. Schneider	Professeur (LIMOS – Clermont Ferrand) Examinateur
M. Joly	Directrice Doc'INSA (INSA de Lyon) Membre invité

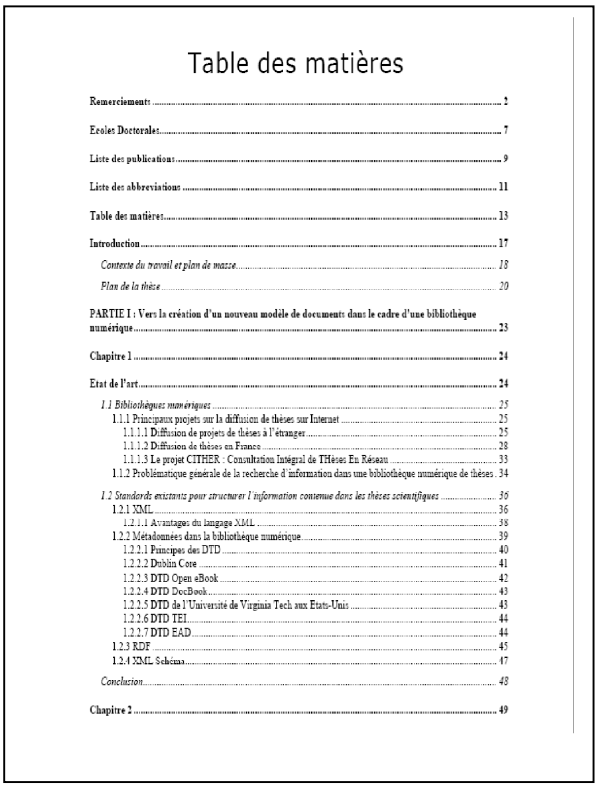


Table des matières

Figure 2 : Une partie de la structure Logique d'une thèse scientifique en format PDF.

La reconnaissance de la structure logique d'un document textuel (en format PDF ou Word) est une tâche complexe. Dans notre thèse, nous allons étudier cette problématique en essayant de répondre à des questions comme : Comment identifier la structure logique et physique d'un document?. Comment peut-on la récupérer?, et ensuite comment l'exploiter pour améliorer l'indexation et la recherche des documents?.

Un autre point intéressant dans notre problématique est la modélisation et l'exploitation sémantique des parties les plus importantes de la structure logique. En prenant comme exemple les termes des titres qui sont très importants, nous allons proposer une méthode pour détecter d'autres termes dans le contenu et qui ont des relations sémantiques avec les titres. Ces nouveaux termes seront considérés des termes importants.

3. Contributions

Pour répondre aux questions précédemment posées, notre thèse apporte les propositions suivantes :

- Proposition d'une architecture pour le système de recherche, en exploitant la structure logique et sémantique des documents numériques afin d'améliorer leur représentation (indexation), et enfin de retourner des documents pertinents comme résultats à une requête de l'utilisateur. Dans notre thèse nous avons utilisé

WordNet comme une ressource sémantique pour modéliser le contenu des documents.

- Proposition d'une approche pour détecter et identifier la structure logique des documents. A partir de cette structure nous allons récupérer les termes qui ont une grande importance et une signification éminente. Ces termes auront un impact sur la représentation des documents.
- Projection des termes importants des documents sur une ressource sémantique (WordNet), pour trouver et extraire d'autre termes, sémantiquement proches, à partir du contenu textuel des documents.
- Proposition d'une méthode efficace pour calculer la similarité sémantique entre les termes en exploitant le contexte de ces termes.
- Retournement des parties les plus pertinentes d'un document par rapport à une requête, en exploitant la hiérarchie de la structure logique de documents pendant la phase de représentation (indexation).

Nos contributions ont été évalué en utilisant deux collections de documents: La première collection est un ensemble de thèse scientifique en format PDF collecté à partir de la bibliothèque de thèse d'INSA [1] -- de Lyon. La deuxième collection est un ensemble de documents de la compagne d'évaluation INEX (INiative for the Evaluation of XML REtrieval) de l'année 2009. Cette dernière collection contient les pages web de Wikipedia en format XML.

4. Organisation de la thèse

La thèse est organisé en deux parties. La première partie décrit et représente l'état de l'art sur les documents numériques, et leur modélisation sémantique et structurelle (chapitre I, chapitre II, et chapitre III). La deuxième partie est destinée à la représentation et l'évaluation de nos contributions (chapitre IV, et chapitre V).

Le premier chapitre représente l'état de l'art des documents numériques, leur caractéristique, les déférents types de structure de documents, et le cycle de vie d'un document. Le chapitre II est un aperçu sur la sémantique de texte, l'ontologie, et le web sémantique. Dans ce chapitre nous présentons le processus de traitement automatique de la langue, et les méthodes de calcul de la similarité sémantique du contenu textuel.

Le chapitre III présente un domaine de traitement et d'exploitation des documents numériques, qui est la recherche d'information. Dans ce chapitre nous essayerons d'expliquer comment on modélise le contenu des documents afin de faciliter leur présentation (indexation) et leur recherche.

[1] Institut national des sciences appliquées

Dans le chapitre IV nous allons expliquer l'approche que nous avons utilisé pour la recherche sémantique des documents numériques. Nous allons détaillé toutes les étapes que nous avons suivie pour améliorer les résultats du système de recherche.

Un autre chapitre (5) est consacré à l'exposition de nos expérimentations et résultats des évaluations appliquées sur notre système. Nous terminons la thèse par une conclusion et des perspectives.

Chapitre 1
Documents Numériques

1.1. Introduction

Le concept «document numérique» est un concept très récent par rapport au document papier. Il est apparu avec l'apparition des nouvelles technologies de l'informatique. Mais il reste toujours difficile à le définir et à le référer. Nous pouvons reconnaître un e-mail et un rapport technique produit par un traitement de texte comme des documents numériques, mais au-delà de ces exemples simples le concept d'un "document" devient moins clair ; Est ce qu'un logiciel est un document ? Est ce qu'un système d'exploitation est un document ?. Il est nécessaire de préciser les normes en vue d'atteindre la bonne notion du document numérique.

Dans ce chapitre nous allons essayer de clarifier les caractéristiques et les déférentes définitions existantes du document numérique.

1.2. La notion de document

Il est difficile de donner une définition complète et précise de la notion du document. Ils existent de nombreux travaux autour des documents qui ont étudié et exploité leurs caractéristiques, mais peu d'entre ceux-ci ont tenté de donner les définitions générales du concept. Le mot document est d'origine latine « documentum » dérivé du verbe « docere », qui signifie enseigner. [ROI 99].

La notion de document se réfère à plusieurs objets ; tout dépend le domaine et le contexte utilisé. Nous pouvons dire qu'il est un document tous les concepts suivants : information, donnée, fichier, texte, image, papier, article, livre, journal, feuille, page... etc. [ALA 04]. La définition du dictionnaire Larousse[2] pour les mots; document, documentation et numériser est comme suit :

Document :

"Pièce écrite servante d'information, de preuve". "Objet quelconque servant de preuve, de témoignage".

Documentation :

Unité d'information correspondant à un contenu singulier.

Numériser :

- C'est le processus qui permet de convertir un objet matériel en une série de chiffres numériques (0 et 1) et de transformer les informations comme le texte, le son, les images et des symboles en symboles numériques pour automatiser le traitement des informations.

[2] http://www.larousse.fr/

- C'est convertir une information analogique à une forme numérique.

Le dictionnaire terminologique de l'office québécois de la langue française [OFF 15] donne plusieurs définitions du terme document. Parmi ces définitions : [NOU 06].

1. Données consignées sur support papier, électronique ou autre, pouvant être utilisées pour consultation, étude ou preuve.

2. Œuvre fixée à un support matériel au moyen du langage ou d'autres symboles.

[BAC 98] considère que le document est un support matériel plus un contenu exprimé : « Un document est un objet matériel exprimant un contenu ». Cette définition nous montre que le contenu en lui-même ne suffit pas pour décrire le document. Par conséquent, la présence du même contenu sur différents types de supports physiques changera la définition du document.

La numérisation n'est pas un but en soi, mais un moyen avec lequel des institutions veulent atteindre des objectifs spécifiques. Lorsque l'information est convertie en un format numérique, elle sera stockés dans un fichier sur l'ordinateur, et sera traité comme tout autre fichier (l'envoi d'un ordinateur à un autre, la copie, l'impression, l'affichage sur écran, …).

Cependant, le terme numériser prend plusieurs significations selon le contexte dans lequel il est utilisé, par exemple la numérisation signifie:

Dans l'informatique : Transformer les données dans un format numérique afin qu'ils puissent être traitées par ordinateur.

Dans les systèmes d'information : Convertir une donnée imprimée qui peut être lus par des humains (c.-à-d analogique), comme les livres, les photos, les cartes, et d'autres formats traditionnels, vers un format lu par l'ordinateur, en utilisant des dispositifs de balayage (scanner), ou bien les appareils photo numériques.

Dans le domaine de la communication : Se réfère à la conversion des signaux analogiques en signaux numériques.

1.3. Document numérique

La numérisation a permis l'annulation du support papier, ce qui a rendu la notion du document ambiguë [ALA 04], est-ce que c'est un fichier, le contenu d'un disque physique, ou un contenu affiché sur écran.

La définition la plus simpliste voit qu'un document est dit numérique quand le support du contenu devient numérique. Dans le domaine numérique, des objets très hétérogènes, tels qu'un fichier texte ou une séquence vidéo sont tous les deux considérer comme un document.

Avec la numérisation massive des données, de nouvelles caractéristiques sont apparues, comme le type du contenu qui regroupe différentes formes d'information (texte, son, vidéo,

graphique,…), le type du support physique, le matériel d'apparition (ordinateur, portable...), la date, la structure logique [ROI 99]. La structuration logique, fait référence à une organisation d'un texte en éléments logiques tels que chapitre, section, sous-section, titre, sous titre, paragraphe...

[ALA 04] définit un document traditionnel comme un (support + inscription) et définit un document numérique comme une (structure + données). Il définit la structure comme tout ce qui n'est pas données dans le document. Il résume la notion du document numérique dans la figure suivante :

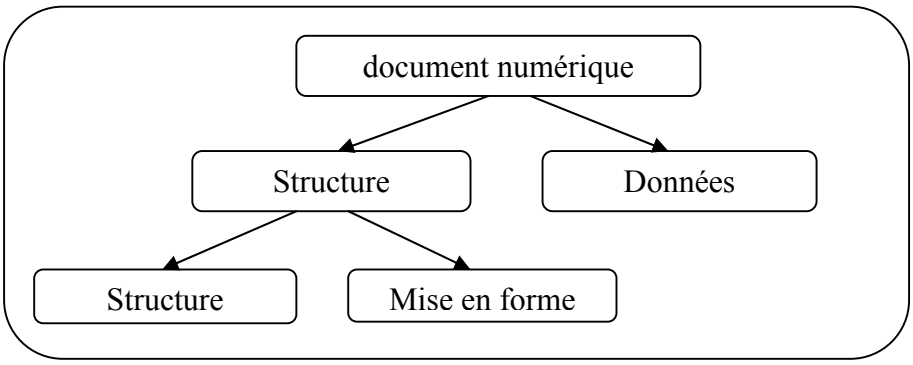


Figure 1.1 : Document numérique [ALA 04].

[ALA 04] analyse le document suivant trois axes : le document comme contenant, le document comme contenu et le document comme médium. L'axe *contenant* étudie les caractéristiques physiques du document. L'axe contenu a comme objectif l'analyse des informations du contenu considéré comme connaissance en essayant d'exploiter le sens. Le rôle de l'axe medium est de prendre en charge le document comme un moyen de communication et étudier l'aspect juridique et social de cet axe.

1.4. Représentation de documents

Les documents doivent être convertis en une forme de représentation qui est lisible par la machine et adéquate pour les logiciels. L'efficacité des applications de traitement de document est liée à la qualité de sa représentation et les données sur lesquelles elle travaille. La représentation d'un document est aussi importante que le choix d'un bon algorithme d'apprentissage.

1.4.1. Structures du document

Partant de la définition de [ALA 04] où un document numérique est composé d'une structure et de contenu (Figure 1.1), nous essayons de décrire les déférentes structures existantes pour les documents. Un document textuel qui se trouve sur un support physique est dit structuré, si le contenu est inscrit sur le support physique sous forme de lignes de textes, où

chaque partie de ce texte a un format (police, taille, couleur ;...). Nous pouvons dire que le document a au moins une structure physique.

Ils existent plusieurs types de structures pour les documents, mais nous pouvons les regrouper en trois catégories principales qui sont : La structure physique qui permet de regrouper les caractéristiques visuelles du contenu. La structure logique décrit l'organisation du contenu sous forme d'éléments logiques où ces éléments sont liés par des relations. La structure sémantique qui permet d'expliciter le sens d'un contenu.

1.4.1.1. Structure physique d'un document

La structure physique d'un document correspond à l'organisation du contenu en fonction d'une hiérarchie des régions délimitées par des images, des graphiques et des blocs de texte qui peut encore être subdivisé en lignes de texte, des mots et des caractères. La structure physique permet de décrire les relations existantes entre les divers objets physiques.

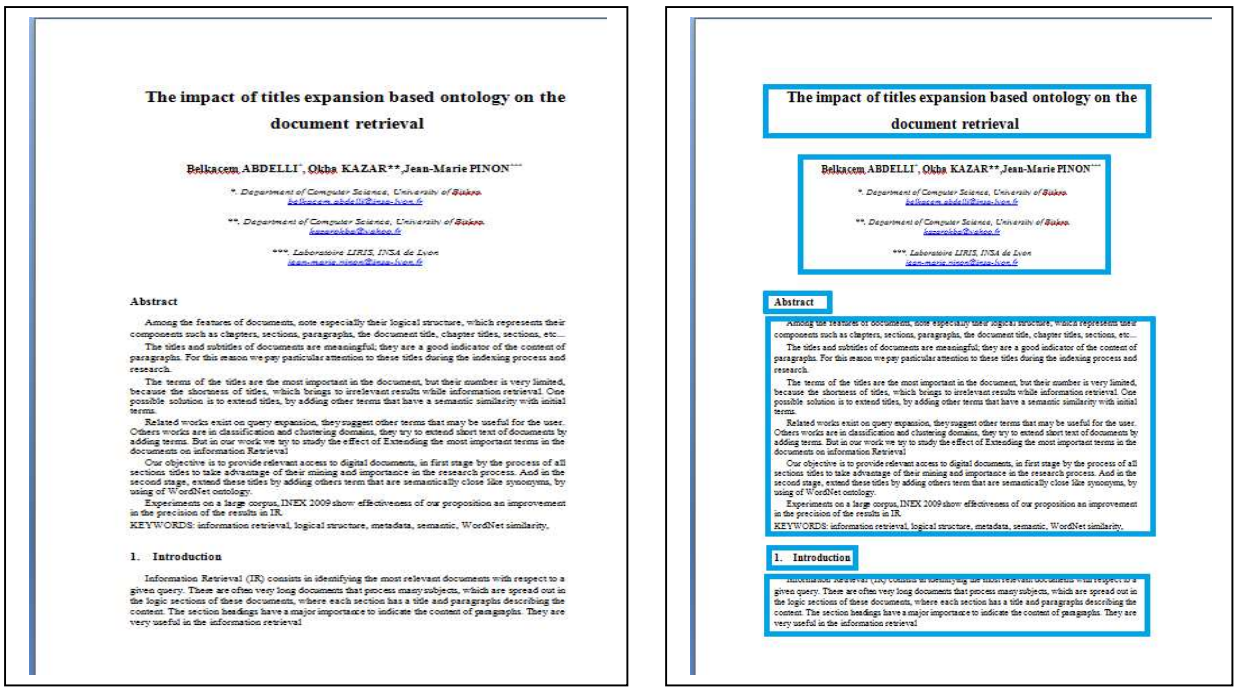


Figure 1.2 : les blocs d'un texte.

1.4.1.2. Structure logique d'un document

La structure logique d'un document explicite la signification de chaque composant de la structure physique. Elle reflète la façon dont l'information est organisée en termes d'objets logiques, à savoir, chapitres, sections, sous sections, titres, paragraphes, figures, en-têtes, etc. La structure logique spécifie la fonction et la signification des objets physiques formant le document et les relations entre eux.

Les relations entre les objets logiques dans la structure logique sont généralement sous une forme hiérarchique. Par exemple, une thèse scientifique comprend un titre, auteurs, résumé, et une séquence de chapitres, chacun d'entre eux, a un titre et une séquence de sections, et ainsi de suite.

La structure logique est généralement représenté sous la forme d'une structure arborescente afin de décrire les relations hiérarchiques existantes entre les différents objets logiques (Figure 1.3).

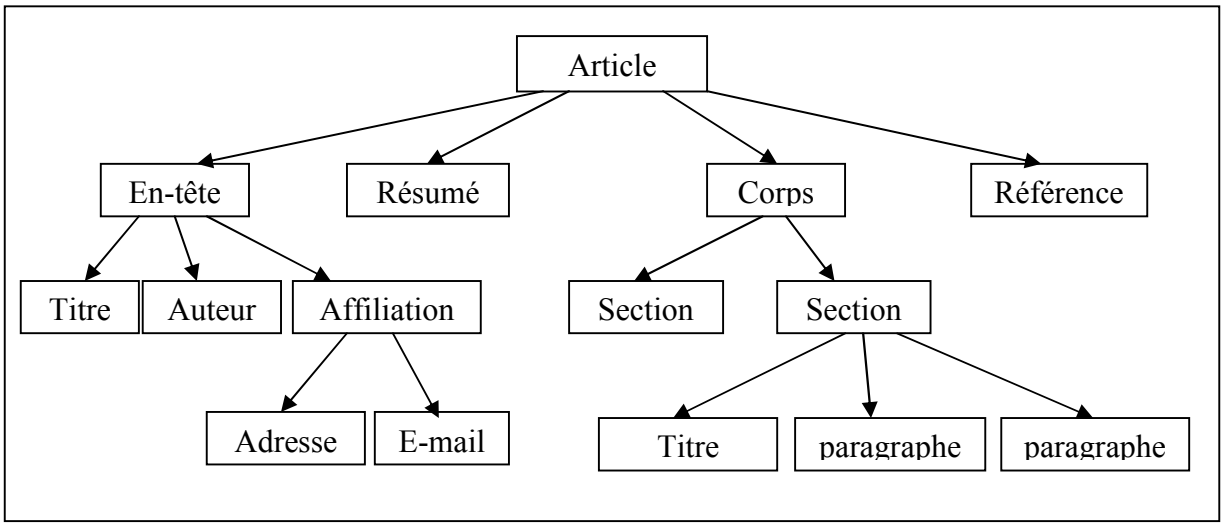


Figure 1.3 : Structure Logique d'un article

Pour modéliser la structure logique du document, plusieurs outils de structuration documentaires sont apparus, parmi ceux-ci il faut citer le langage XML (Extensible Markup Language défini par W3C[3]), qui est bien standardisé et bien formalisé.

XML est un langage défini pour faciliter la manipulation et l'échange de documents, grâce à ce langage de nouvelles tendances sont développés tels que ; la recherche d'information par le contenu, le web sémantique, le web service... etc.

XML est connu par l'utilisation des marques spéciales appelée balises qui rendent la structure du contenu bien explicite, et bien délimité. Ce langage qui sera détaillé dans le deuxième chapitre) est la première étape dans le domaine du web sémantique.

[3] ww.w3c.org

```
<Article>
      <En-tête>
          <Titre>.....</Titre>
          <Auteur>.....</Auteur>
          <Affiliation> .....</ Affiliation>
          <Corps>
              <Section>
                    <Titre>.....</Titre>
                    <Paragraphe>.....</Paragraphe>
                    <Paragraphe>.....</Paragraphe>
              </Section>
              <Section>
                    .....
                    .....
              </Section>
.....
</ Article>
```

Figure 1.4 : La structure logique d'un article (de la figure 1.3) représenté par XML

1.4.1.3. Structure sémantique d'un document [ABA 05]

La structure sémantique permet de décrire le sens du contenu du document, et définir les relation sémantique entre les termes du contenu, la structure sémantique et décrite par un ensemble de concepts au lieu de mots. Pour extraire les concepts on doit utiliser les outils de traitement automatique de la langue (TAL).

TAL permet la compréhension du contenu textuel du document par l'identification de la sémantique à partir des multiples sémantiques possibles qui peuvent être dérivées d'une expression du langage naturel.

L'identification de la structure sémantique nécessite l'utilisation d'une ressource sémantique externe pour trouver le concept associé à chaque mot ou par une analyse statistique afin de regrouper les mots similaires à un concept unique, et de définir les relations sémantique entre les termes. En effet, l'objectif est de comprendre la signification d'un document et de détecter le même concept expliqué par des mots différents.

Beaucoup de termes ont plusieurs sens; la ressource sémantique permet de choisir le meilleur sens dans le contexte.

1.4.2. Les modèles de représentation des documents

Généralement les documents texte dans leur forme originale ne peuvent pas être interprétés par les machines. Pour cette raison, ils doivent être convertis en une forme de représentation qui est lisible par machine. L'efficacité du traitement et l'exploitation des documents sont liées à la qualité de la représentation. La représentation du document est aussi importante que le choix d'un bon algorithme de traitement. Ils existent différents modèles de représentation, nous présentons dans ce qui suit les modèles de base.

1.4.2.1. Modèle vectoriel

Le modèle vectoriel est une technique mathématique pour la représentation des documents. Il est Proposé par Salton en 1975 [SAL 75]. Dans ce modèle chaque document est représenté par un vecteur de termes de N dimension. N représente le nombre de termes dans la collection de documents.

La similarité entre deux vecteurs de documents $\overrightarrow{d1}$ et $\overrightarrow{d2}$ (ou bien document et requête) est calculée en prenant le cosinus de l'angle entre eux. Plus l'angle entre les deux vecteurs est petit , plus le score de cosinus sera plus élevé, plus les document seront considérés comme semblable.

$$\text{Cos}(\theta) = \frac{\overrightarrow{d1} \cdot \overrightarrow{d2}}{|\overrightarrow{d1}|\,|\overrightarrow{d2}|}$$

$(\overrightarrow{d1} \cdot \overrightarrow{D2})$ est le produit scalaire des deux vecteurs, tandis que $|\overrightarrow{d1}|\,|\overrightarrow{d2}|$ sont leurs longueurs euclidiennes utilisées pour normaliser les vecteurs correspondant à leurs vecteurs. En utilisant cette représentation, une requête est traitée comme un document et la similitude Sim(d, q) entre un document d et une requête q peut être calculée en prenant le cosinus de l'angle entre leurs vecteurs correspondants, $\vec{d}$ et $\vec{q}$.

Documents et requêtes sont représentés comme des vecteurs

$$D_j=\{w_{1,j}, w_{2,j},\ldots\ldots\ldots\ldots, w_{N,j}\},\quad q=\{w_{1,q}, w_{2,q},\ldots\ldots\ldots\ldots, w_{N,q}\}$$

N indique la dimension de l'espace vectoriel, $w_{i,j}$ est le poids du terme t_i dans le document d_j

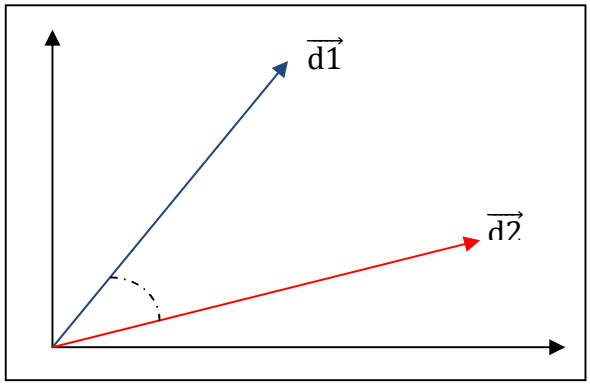


Figure 1.5: Un modèle vectoriel pour deux documents

Une valeur est associée avec chaque terme de document. Si le terme apparait dans le document sa valeur est différente de zéro. Cette valeur, connue comme un poids, peut-être simplement un nombre d'occurrences du terme dans le document, ou bien elle est calculée.

Le modèle vectoriel, ne précise pas comment les poids des termes devraient être calculé. Une approche pour obtenir des poids des termes et basé sur les statistiques du terme, peut êtres utilisée. cette approché est appelée TF-IDF [SAL 75]. TF se réfère à la fréquence de

terme dans un document. La fréquence inverse de document (IDF) est utilisé pour contrer les insuffisances de TF. Elle diminue le poids des termes qui apparaissent très fréquemment dans la collection des documents et augmente le poids des termes qui se produisent rarement.

En combinant TF et de la IDF, termes qui apparaissent fréquemment dans un document mais rarement dans la collection reçoit le poids le plus élevé

1.4.2.2. Modèle probabiliste

Est un modèle de représentation du contenu d'un document, il est proposée en 1976 par Robertson et Jones en 1976 [ROB 76]. L'approche du modèle probabiliste estime explicitement la probabilité qu'un document est pertinent à une classe. Les documents retrouvés sont présentés au chercheur et classés en fonction de cette probabilité de pertinence.

La représentation du document est basée sur un vecteur de caractéristique dans laquelle les poids des termes sont remplacés par leur probabilité d'être pertinentes à une classe obtenue à partir d'un ensemble de jeu de test. Une façon d'estimer la probabilité de pertinence est d'utiliser des jugements de pertinence précédents.

Alors pour vérifier si un document dj est associé à la classe C sa probabilité est calculée par $P(R \mid \overrightarrow{dj})$. Aussi la probabilité que le document n'est pas pertinent pour une classe sera représenté par $P(\overline{R} \mid \overrightarrow{dj})$:

$$Sim(d_j, C) = P\left(R \mid \overrightarrow{dj}\right) / P(\overline{R} \mid \overrightarrow{dj})$$

1.4.3. Représentation du contenu

Le contenu texte d'un document est composée d'une séquence de caractères. Avant l'exploitation des documents, il est nécessaire de transformer le contenu à des entités significatives. En effet, l'objectif est de passer d'un espace de caractère à un espace de terme, puis d'éliminer le bruit (comme les mots vides) et de sélectionner les mots les plus représentatifs. Ces opérations sont basées sur des règles qui diffèrent selon la langue du document.

Les mots sont de plusieurs types (nom, verbe, adjectif,...); comme ils se trouvent sous forme simple ou composés. Nous devrions appliquer différents processus pour les extraire. La sélection des meilleurs termes implique également les processus de filtrage qui permettent d'estimer l'importance des termes pour supprimer ceux non pertinents.

La Figure (1.6) montre les phases effectuées sur un document; pour extraire et sélectionner les composants les plus important dans le contenu. Ces phases seront expliquées dans les sections suivantes .

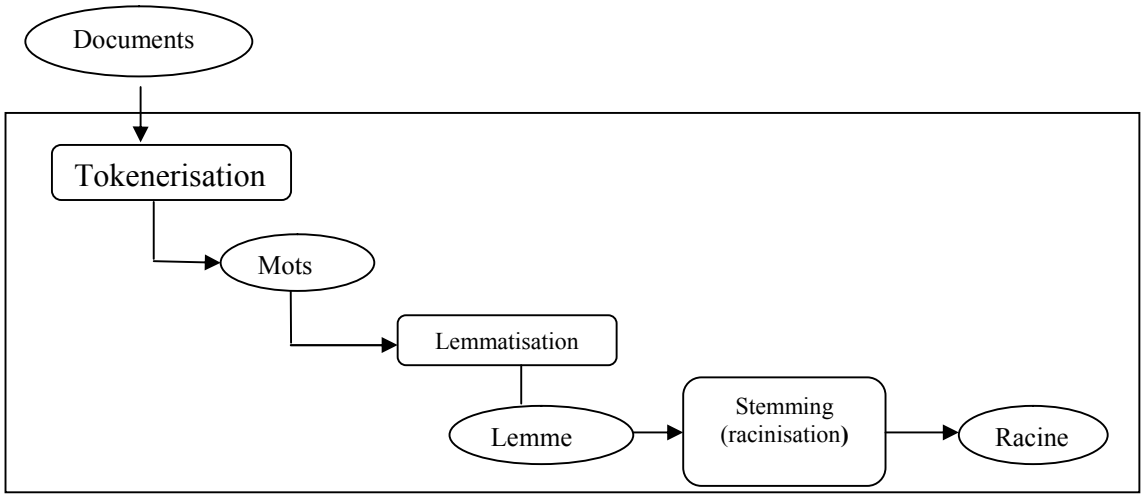


Figure 1.6 : Prétraitement du contenu d'un document

1.4.3.1. Types du contenu textuel

Les termes d'un document texte peuvent être un mot simple, mots composé, une phrase ou un concept extrait par une ressource sémantique; comme il peut être aussi être un élément de structure.

1.4.3.1.1. Unités lexicales

Est un ensemble de caractère (lettres ou chiffres ou symboles) terminés par un séparateur. Le séparateur est un caractère tel que l'espace ou de ponctuation. Une unité lexicale peut être un mot, un nombre, un symbole et une ponctuation.

1.4.3.2. Mot

C'est la plus courte et la plus simple unité textuelle significative dans un document texte. le texte est divisé en «lettres» et en «séparateurs», et un mot est une séquence de lettres entourées par des séparateurs [NAV 99].

1.4.3.3. Lemme

Les lemmes sont des formes canoniques d'un ensemble de mots avec le même sens qui sont utilisé comme une citation pour désigner ce groupe dans le dictionnaire. Par exemple, *informatique ,informer ,informés ,informez ,informé et information*, ce sont des différentes formes ayant le même lemme qui est *informe.* Le lemme dénote une étiquette qui se présente comme la forme infinitive de verbes ou la forme singulière des noms et des adjectifs [LAL 05].

1.4.3.4. Racine

Racine (ou Stem) est la forme de base du mot auquel les préfixes ou les suffixes sont ajoutés pour former toutes les formes du mot [Ben 08]. Il est en effet la plus petite partie d'un

mot qui est commun à toutes ses variantes. Par exemple les mots : ***étude, étudiant*** et ***étudier***, sont construits à partir de la racine "***étud".***

1.4.3.5. Mot composé

Un mot composé est constitué de deux ou plusieurs mots attachés pour former un nouveau mot. Cette composition permet une nouvelle signification différente par la signification de chaque mot. Par exemple "base de donnée", "*pommes de terre",...*

1.4.3.6. Phrase

Dans l'analyse grammaticale, une phrase se réfère à un seul élément dans la structure d'une expression contenant typiquement plus qu'un mot. Les phrases peuvent être classés en plusieurs types comme «syntagme nominal» et «syntagme verbal» . Un syntagme nominal peut être un sujet ou objet dans la phrase comme «chien noir » ou « petit enfant ». Un syntagme verbal joue le rôle d'un verbe unique et se compose d'un verbe principal et verbes auxiliaires, comme :"Ilyès aime l'informatique".

1.4.3.7. Concept

Concept se concentre plus sur la sémantique du texte. Il est une unité abstraite de connaissances qui explique le sens des mots. Il existe deux types de sémantiques ; sémantique lexicale considère la sémantique des mots séparés, sémantique grammaticale tient compte de la relation entre les mots pour atteindre leur concept associé. La représentation conceptuelle de document ne peut pas être atteint facilement. Il faut utiliser les ressources sémantiques externes pour extraire le concept à partir du texte**.**

1.5. Gestion électronique des documents

La Gestion Electronique des Documents GED; est le processus de gestion du cycle de vie d'un document électronique, Le processus de GED contient 3 étapes principales: l'acquisition, le stockage, la diffusion [STE 11].

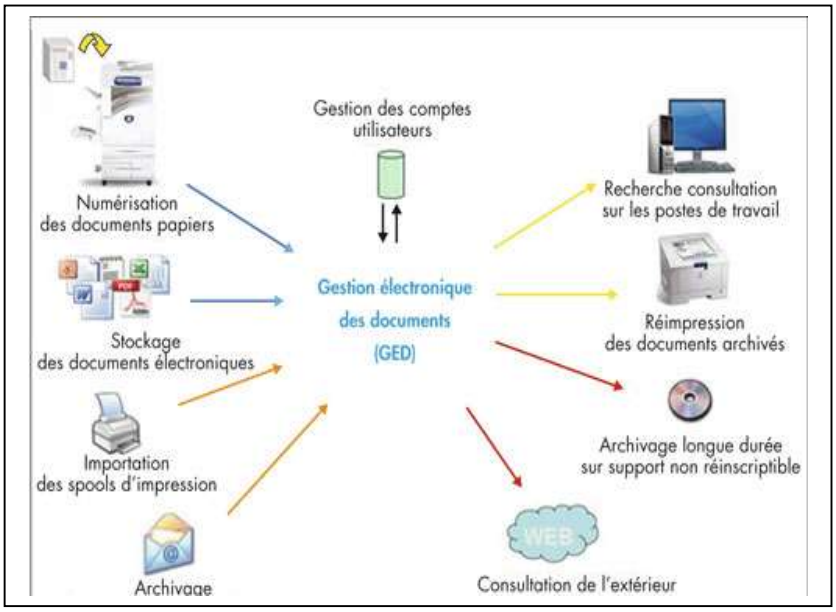


Figure 1.7: Processus du GED[4]

1.5.1. Acquisition des documents :

L'acquisition c'est le processus qui permet la numérisation et la création de document. Les différentes phases de l'acquisition sont : La création, Le classement, et l'indexation.

1.5.1.1. Création

La création est issue de différents procédés **:**

1.5.1.1.1. L'intégration de documents électroniques existants :

Acquisition directe de l'information sous forme numérique: par le processus d'entrée en utilisant le clavier ou par l'utilisation les appareils photo pour les images numériques et des clips vidéo.

La plupart des documents actuels sont conçus et crées à partir d'un ordinateur. Ces documents se présentent sous forme de fichiers avec des extensions différentes.

Le résultat de l'acquisition numérique est un ensemble d'objets numérique de type texte (TXT, DOC, PDF, HTML, XML.....), image (TIFF, GIF, JPEG, PNG), audio(WAV, MP3,WMA,....), ou vidéo (MPEG, MP4, FLV,.....). [BEL 00] :

1.5.1.1.2. La numérisation de documents papiers existants :

L'acquisition indirectement par l'intermédiaire de scanners, permettant la numérisation des documents en format analogique, ces dispositifs se généralisent et deviennent de véritables plate-forme d'acquisition.

[4] www.techniques-ingenieur.fr

1. Méthodes de numérisation

Les techniques et les outils utilisés pour réaliser des projets de la numérisation de contenu représentent une importance majeure, elles permettent de traiter les sources d'information pour les archiver et les stocker, pour faciliter ensuite l'accès aux utilisateurs. On peut généralement identifier trois (03) méthodes de numérisation de contenu: numérisation sous la forme d'une image, numérisation sous la forme d'un texte, et numérisation sous forme Vectorielle

A. Numérisation en mode image

Dans ce type de numérisation les images sont sous la forme de points et se composent d'une grille de points appelés Pixels, Le résultat de la numérisation est un ensemble d'images qui ont un codage spécifique du noir et blanc et niveaux de gris jusqu'à l'arrivée au reste des couleurs. [CAT 02].

Dans cette méthode, la numérisation de document textuels (Livres, page de texte, ...) est réalisée sous la forme d'une photo de chaque page du texte, elle nous donne une copie numérisée correspondant exactement au texte original.

Cependant, cette méthode est imparfaite, elle n'autorise pas d'effectuer des recherches dans le texte ce qui nous conduit à faire une description bibliographique complète du contenu textuel du document, ainsi un processus de catalogage, est nécessaire pour faciliter l'accès aux documents.

Ainsi parmi les inconvénients de cette méthode l'espace considérable de stockage des documents numérisé. Le résultat de la numérisation est un ensemble d'images bitmaps de grande taille qui nécessite des supports importants.

Bien que la méthode de numérisation sous la forme d'une image a l'avantage d'être la manière la plus facile et simple qui peut être appliquée, et elle est la méthode de numérisation la moins chers.

B. Numérisation en mode Vectoriel

C'est une technique de représentation par des équations mathématiques, le principe essentiel dans le processus de reconstitutions Les données d'image par des équations géométriques qui permettent de les lire mathématiquement.

L'image vectorielle est un ensemble d'éléments qui sont décrits en fonction de leurs formes géométriques.

La conversion des documents en forme vectorielle n'est pas simple parce que les symboles, les dessins, et les éléments géométriques, doivent être reconnus et convertis vers un modèle approprié à l'image vectorielle.

Parmi les avantages de cette catégorie d'images est la possibilité d'agrandir infiniment sans perdre sa qualité.

Les documents textuels en format PDF de l'entreprise Acrobat sont représenté en forme vectorielle. Ce qui les rend de petite taille. [CAT 02].

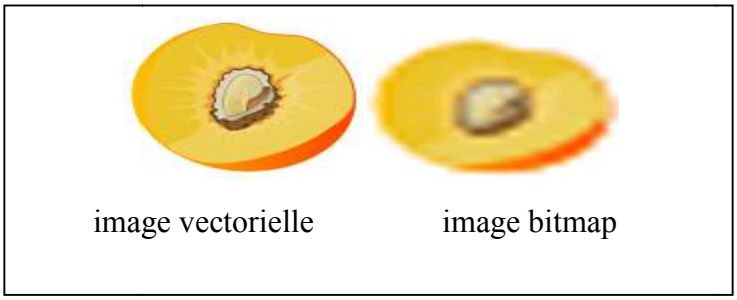


Figure 1.8 : différence entre image vectorielle et bitmap

Cependant, la transformation du document papier en un document numérique par cette technique est très coûteuse et il nécessite souvent le recours à un spécialiste pour confirmer le bon déroulement du processus de conversion.

Cette technique ne donne pas de bons résultats avec des documents textuels, Il est très difficile de représenter les mots et les lettres comme des formes géométriques .

C. Numérisation en mode texte

Cette technique de numérisation permet de coder les documents textuels en gardant le contenu textuel en tant que tel. Ce mode de numérisation s'obtient par deux méthodes.[ALB 02] :

1. A partir d'un ordinateur ou un logiciel de saisie et de création de documents (traitement de textes) : dans ce cas le document est présenté directement sous forme numérique et ne nécessite donc pas un processus de numérisation. Ce type de document garde alors sa forme de présentation en plus de son contenu.

Cette option se limite aux nouveaux documents qui seront écrits dès le début par un environnement traitement de texte. Mais pour reproduire les documents textuels déjà crée (textes manuscrits dans des polices anciennes) en utilisant cette option, ça nous nécessite beaucoup de temps et de main d'œuvre.

2. Numérisation du document par l'application d'une reconnaissance optique de caractères (OCR) aux images obtenues. Cette technique nous permet de récupérer le contenu textuel, mais pas la présentation qui sera perdue. Pour récupérer une partie de la structure logique du document on doit utiliser des techniques pour la reconnaissance des titres de section et les paragraphes. La numérisation en mode texte doit tenir compte de trois éléments

principaux : Le codage des caractères ; la structure physique du document ; la structure logique du document ;

2. Reconnaissance optique de caractère

Reconnaissance optique de caractères (OCR) est une conversion de document textuel scanné ou sous format imprimé [ARC 12] en un texte. Cette technologie permet aux machines de détecter et manipuler automatiquement le texte. OCR permet de convertir un document sous format image, en traduisant son contenu lettre par lettre et mot par mot en un fichier textuel, afin de faciliter la recherche et la navigation dans le texte.

Les programmes de reconnaissance optique de caractères utilisent des dictionnaires intégrés pour détecter les termes et corriger les erreurs.

1.5.1.2. Le classement des documents

Cette opération consiste à classer les documents pour qu'ils soient accessible facilement par les utilisateurs. Le classement est réalisé en ajoutant des mots clé (appelés Métas donnés) aux documents pour faciliter l'accès, cette opération nécessite une intervention des experts afin de définir les mots clé les plus adéquats au contenu des documents.

1.5.1.3. L'indexation des documents

L'indexation consiste à recueillir, extraire, et stocker les termes ou les expressions du contenu textuel des documents pour faciliter la recherche d'information rapide et précise, on distingue à ce titre deux méthodes :

La recherche par mot clés : elle utilise les mots clé (Métadonnées) ajoutés au document, pour faciliter la recherche. C'est la recherche la plus pertinente, mais elle nécessite l'intervention d'un expert du domaine pendant la phase de classement, afin de saisir les Métadonnées.

La recherche plein texte: exploite le contenu textuel du document. elle extrait les mots et les expressions contenus dans le document et les stocke dans un indexe qui est utilisé pendant la recherche

1.5.2. Conservation des documents numériques

Stocker les documents électroniques comprend souvent la gestion de ces mêmes documents; où ils sont stockés, pour combien de temps, la migration des documents d'un support de stockage à un autre et la suppression des documents.

1.5.3. La diffusion du document

Elle consiste à distribuer les documents dans un environnement fermé d'une entreprise (intranet), ou dans un environnement ouvert comme internet. Le document destiné à la diffusion doit être dans un format qui ne peut pas être facilement modifiée.

1.6. Les métadonnées

Le terme métadonnées est un terme large qui porte de nombreux sens, selon les domaines dans lesquels il est utilisé et selon les communautés professionnelles qui conçoivent et créent les documents électroniques.

Certains utilisent ce terme pour se référer aux informations comprises par les machines, tandis que d'autres l'utilisent seulement comme données qui décrivent les ressources électroniques.

Le terme métadonnée est composé de deux parties: Meta qui signifie «pour» ou «à propos de» et donnée. Dans le contexte de la terminologie Informatique, en le comparant au mot "métalangage" qui est un langage de tous les langages ou le langage qui décrie un autre, et donc Le terme métadonnées signifie données sur les données, ou des données qui décrivent d'autres données. [YOU 01].

Avec la diffusion de l'Internet, et la croissance du réseau mondial le terme «métadonnées» commence à être utilisé dans le contexte de la description des ressources numériques disponibles dans le web. Ainsi que les documents textuels qui sont compréhensibles par les êtres humains ont un besoin des métadonnées pour que les machines puissent les manipuler, les contrôler, les rechercher et les récupérer.

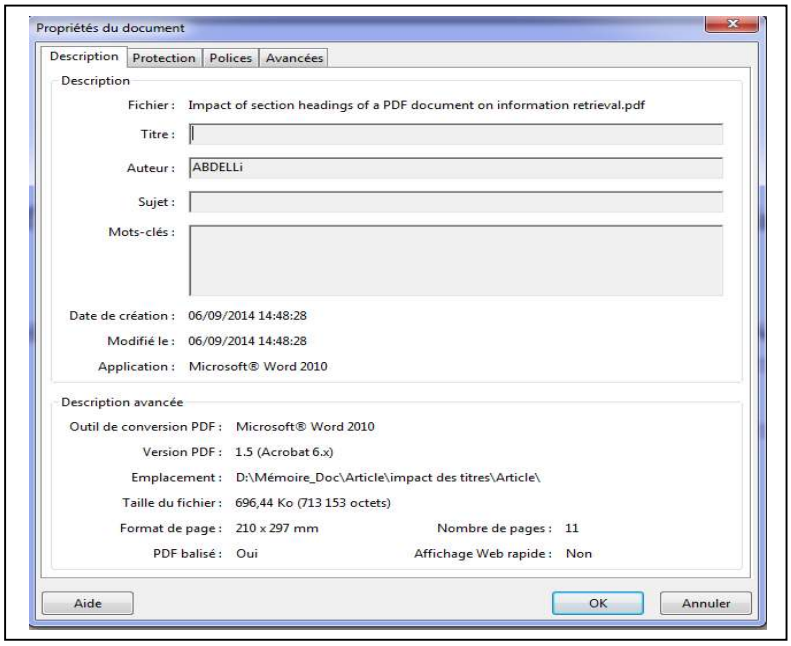


Figure 1.9: Métadonnées d'un document PDF

1.6.1. Définition de métadonnées

Les métadonnées sont des informations structurées qui décrivent, expliquent, localisent une ressource d'information, ces ressources deviennent plus facile à utiliser, à récupérer, et à gérer. Le terme “ métadonnées ” est utilisé différemment selon les communautés. Certains l'utilisent pour se référer à des informations “ lisible et compréhensible par la machine ” [YOU 01]. Tandis que d'autres l'utilisent uniquement comme des informations qui décrivent les ressources électroniques.

Dans l'environnement de la bibliothèque, les métadonnées sont utilisé dans un système formel de description des ressources, numérique ou non numérique.

Dans ce qui suit quelque définition les plus connue du terme sont citées:

Le W3C[5] adopte une définition très restreinte pour les métadonnées : " Métadonnée c'est l'information compréhensible par la machine pour le web".

Alors que nous constatons que **Tim Barners Lee** (le président de l'Union mondiale du W3C et l'inventeur du réseau World Wide Web ou WWW), définit métadonnée comme " Information lisible et compréhensible par la machine qui décrit des sources sur le Web ou d'autres sources".

Tandis que l'Organisation internationale de normalisation ISO [6] le définit comme données contenues dans une entité ou liées à une entité quelconque, et décrit cette entité et aide à la récupérer.

1.6.2. Importance des Métadonnées

Le but principal des métadonnées est de faciliter l'exploration des informations pertinente, en plus de la découverte de nombreuses autres tâches qu'on décrit dans les points suivants [NIS 04] :

1.6.2.1. Découverte des ressources

Les métadonnées permettent la découverte de ressources électroniques par le biais de :

- Diagnostiquer et identifier les ressources.
- Combiner ensemble les ressources similaires.
- Distinguer les ressources qui ne ressemblent pas.
- Donner des informations de localisation.

[5] World Wide Web Consortium : www.w3.org

[6] www.**iso**.org

1.6.2.2. Interopérabilité

L'interopérabilité est la capacité de différents systèmes, avec différentes plates-formes matérielles et logicielles, d'échanger des données avec une perte minimale du contenu et de la fonctionnalité. En utilisant des schémas définis de métadonnées, les ressources électroniques à travers le réseau peuvent être recherchées de façon plus transparente.

1.6.2.3. Organiser les ressources électroniques

Regrouper les sites et les portails est de plus en plus utile dans l'organisation des liens vers des ressources. Ces listes peuvent être construite, avec les noms et les emplacements des ressources stockés dans le code HTML. Cependant, il est plus efficace de construire des pages à partir des métadonnées.

1.6.2.4. Identification numérique

C'est un numéro standard pour identifier de manière unique la ressource ou l'objet pour que les métadonnées se réfèrent.

1.6.3. Dublin Core

Beaucoup de formats de métadonnées sont élaborés dans une variété d'environnements et de disciplines utilisateur. Le plus connu est Dublin Core[7].

L'objectif initial de Dublin Core était la définition d'un ensemble d'éléments qui pourraient être utilisés par des auteurs afin de décrire leur propre page Web. Face à une multiplication des ressources électroniques et l'incapacité des experts de la bibliothèque de cataloguer tous ces ressources, l'objectif était de définir quelques éléments et quelques règles simples qui pourraient être appliquées par les non-catalogueurs. Dublin Core contient 15 éléments : Titre, Créateur, sujet, description, Editeur, Contributeur, Date, Type, Format, Identificateur, Source, Langue, Relation, couverture, et Droits.

[7] www.dublincore.org

Title=”Metadonée”
Creator=”Belkacem, Abdelli”
Subject=”metadata”
Description=” un aperçu sur les métadonnées.”
Publisher=” edition universities”
Date=”2015-02"
Type=”Text”
Format=”application/pdf”
Identifier=”http://univ-biskra.dz/resources/
Metadonée.pdf”
Language=”FR”

Figure 1.10: Exemple de Dublin Core

1.7. Conclusion

Dans ce chapitre, nous avons présenté les notions et les concepts préliminaire liée au document numérique. Nous avons décrit les principes fondamentaux des systèmes de numérisation des documents. Nous avons montré que les documents ont plusieurs modèles de représentation, ainsi que plusieurs structures qui les forment.

Dans le chapitre suivant nous allons décrire la notion sémantique qui se trouve dans le contenu des documents.

Chapitre 2
Web Sémantique

2.1. Introduction

Avec la quantité énorme de documents sous format numérique dans le web, il est devenu très difficile de trouver le document le plus pertinent qui répond à la requête de l'utilisateur. Ce qui a mené les experts à exploiter les techniques et des méthodes de la linguistique et le traitement automatique de la langue.

Dans ce chapitre nous allons détaillé les notions et les concepts de la modélisation sémantique et linguistique des contenus textuels des documents. parmi ces notions; l'ontologie, la désambiguïsation et la similarité sémantique.

2.2. Web sémantique

Le web sémantique est un domaine qui s'intéresse à la sémantique du contenu des pages web en particulier et le contenu des documents en générale.

Le Web sémantique est un système qui permet aux machines de «comprendre» le contenu des documents et de répondre aux demandes complexes de l'humain en exploitant le sens. Pour se faire, les documents doivent êtres sémantiquement structurés. [BAK 05].

Selon le W3C[8], "Le Web sémantique fournit un cadre commun qui permet aux données d'être partagées et réutilisées par les applications, et les entreprises». Le terme a été inventé par **Tim Berners-Lee** pour un web de données qui peut être traité par les machines [BER 01].

2.1.1. Définition globale du web sémantique

Le web sémantique est une extension du web actuel, et son contenu est compréhensible par des machines, c'est pourquoi l'utilisation du terme « sémantique », il traite le « sens » qui se trouve dans le contenu des pages.[PET 15].

Définition du W3C : « Le web sémantique est une vision : l'idée que les données sur le web soient définies et liées de manière à être utilisées par des machines non seulement pour le but d'affichage, mais pour l'automatisation, l'intégration et la réutilisation sur des plates-formes variées ».

Pour parvenir à rendre le contenu du web compréhensible et exploitable par des machines, le W3C travaille pour élaborer des standards et des langages pour modéliser le contenu des documents, parmi ces standards nous citons les suivants :

1. L'utilisation de XML afin de permettre la structuration des documents.
2. L'utilisation de RDF pour permettre la signification de ces structures grâce aux « triplets ». À chaque tag XML correspondra un « triplet » RDF.

[8] www.w3c.org

3. L'utilisation des ontologies, pour résoudre les problèmes de terminologie. Chaque ontologie comprendra un ensemble de règles d'inférences.
4. L'utilisation d'agents logiciels capables de collecter les contenus de web, de les traiter, de se coordonner entre eux, et même d'échanger des « preuves »

Tim Berners Lee résume le fonctionnement du web sémantique dans l'architecture suivante :

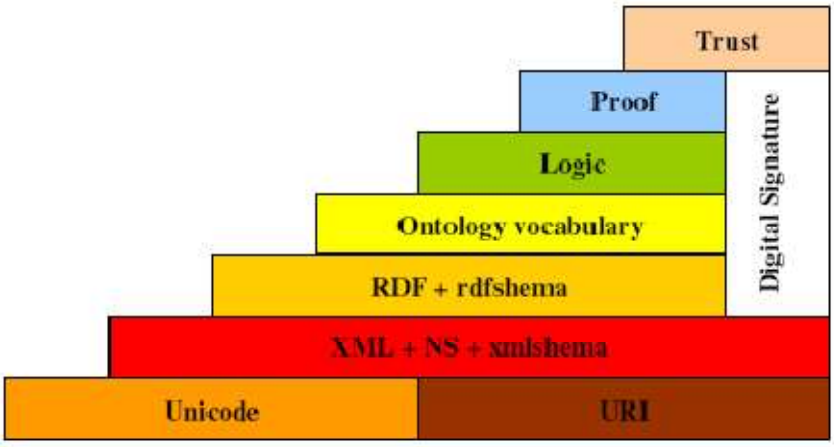


Figure 2.1 : Les couches du Web sémantique selon le W3C

2.1.2. Langages du Web Sémantique

2.1.3.1. XML

eXtensible Markup Language ou XML est une version simplifiée de SGML publié par W3C en 1996. Il conserve les avantages clés SGML tels que, la structure et la validation, mais il est conçu pour être simple et facile à apprendre et à utiliser que le langage SGML (LAL 09). XML permet aux utilisateurs de créer leurs propres étiquettes sans aucune limite. Cependant, dans HTML les balises sont limitées à une liste définie. XML est devenu le format standard le plus populaire. Il permet la représentation du contenu et la structure des documents de façon indépendante.

Dans la Figure 2 Un XML simple document est affiché. Le document représente un article avec quelques éléments comme le titre, auteur, année, résumé et section. L'élément «section» lui-même contient deux éléments ; titre et paragraphe.

En HTML, le rôle d'une balise est de définir dans un navigateur web comment une partie de texte doit être affiché. En XML, les balises définissent généralement la structure logique du contenu, alors que le format de l'affichage du texte est spécifié par les feuilles de style. XML en effet sépare la structure logique du contenu de document.

```
<?xml version="1.0" encoding="utf-8"?>
<article>
<titre> web sémantique </titre>
<auteur> Belkacem Abdelli </auteur>
<année> 2015 </année>
<résumé> Le Web sémantique est un système qui permet aux machines de «comprendre» le
        contenu .....
</résumé>
<section>
        <titre>  Définition globale du web sémantique  </titre>
        <paragraphe> Définition du W3C : « Le web sémantique est une vision : l'idée que
              les données sur le web soient définies et liées de manière à .....
        </paragraphe>
</section>

<section>
        <titre>  Historique  </titre>
        <paragraphe> Le concept du Réseau sémantique a été formé dans les années 1960
        </paragraphe>
</section>
</article>
```

Figure 2.2 : Exemple d'un document XML

Chaque élément XML est nommé par son étiquette, par exemple, l'élément <section>. Les éléments peuvent être imbriquées, mais ne doivent pas se chevaucher. Comme montré dans la Figure 2 la première ligne est la déclaration XML qui identifie la version XML et l'encodage utilisé. La deuxième ligne indique l'élément racine qui définit ce document qui est un "article". Les éléments suivants à l'intérieur de la racine sont ses enfants ; aussi chaque enfant peut avoir autres éléments.

Le Document Type Definition (DTD) associé à un document XML décrit la structure générique du document. Il contient toutes les balises qui peuvent être incluses dans les documents et aussi les relations entre ces balises. DTD peut être déclarée à l'intérieur du Document XML ou séparément stockées dans un fichier de données et être référencée dans la partie supérieure du document XML. La Figure 3 représente un DTD simple, associée au Document XML de la Figure 2.

DTD comme on le voit dans l'exemple n'est pas écrit en syntaxe XML :

"DOCTYPE": indique que l'élément racine dans le document est «article»

"ELEMENT article (titre, auteur, ...)": indique que l'élément «article» contient 5 éléments à l'intérieur; titre, auteur, année, résumé et de l'article

· " ELEMENT titre (#PCDATA)": définit ce que l'élément 'titre' peut impliquer un texte.

```
<!DOCTYPE Article [
<!ELEMENT article  (titre, auteur, année, résumé, section )>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT auteur (#PCDATA)>
<!ELEMENT année (#PCDATA)>
<!ELEMENT résumé (#PCDATA)>
<!ELEMENT section (titre, paragraphe)>
<!ELEMENT titre (#PCDATA)>
<!ELEMENT paragraphe (#PCDATA)>
])
```

Figure 2.3: Exemple d'un DTD

Document structuré et XML

Au contraire d'un document non structuré qui est un texte brut, sans aucune organisation, un document structuré est écrit dans un ordre logique pour permettre la représentation de son contenu d'une meilleure façon pour qu'il soit clair et bien compris avec des titres en gras, différentes tailles de polices, couleurs etc. [RAM 10]. En effet, un document structuré permet aux auteurs d'organiser leurs documents en sections, paragraphes et autres éléments dans un ordre souhaité. Dans le document structuré des balises sont utilisées pour donner une forme structurels à différentes parties du document.

Représentation de la structure logique de document:

La structure décrit le rôle de chaque unité logique dans des documents (titre, chapitre, paragraphe...) Toutes ces unités sont organisées en un arbre hiérarchique pour représenter la relation entre eux. Le langage de balisage XML, est devenu l'un des moyens le plus pratique pour représenter des documents structurés dans l'internet. Transformer le document vers XML a causé un énorme volume de documents XML stockés sur le web. Par conséquent, l'augmentation du nombre de ce type de documents a considérablement augmenté l'intérêt de leur exploitation en fonction de leur structure en amant avec leur contenu textuel. [RAM 10].

Représentation de la structure physique :

Pour décrire la structure logique d'un document le langage XSL [9] (eXtensible StyleSheet Language) est utilisé. Ce langage qui appartient à la famille des langages dérivés de XML, permet de définir les feuilles de styles des documents XML afin de générer d'autres documents à partir de XML (HTML, PDF, RTF...).

XSL permet de deviser les données d'un document en une liste de blocs qui, à leur tour, contiennent chacun une liste de données texte, et pour chaque type de texte (titre, chapitre, paragraphe, numéros de pages) il est appliqué la mise en forme adéquate. Le langage XSL est subdivisé en deux variantes :

1. Le langage de transformation des données XSLT (eXtensible Stylesheet Transformation) qui permet la transformation de documents XML en d'autres formats tels que HTML ou PDF.
2. Le langage de formatage des données (XSL/FO) pour la mise en forme de données XML

Le langage XML permet de séparer la structure logique de la structure physique. Cette séparation permet de rendre XML un document portable et réutilisable par plusieurs applications [RAM 10]. La structure logique est prise en compte par plusieurs domaine de recherche pour exploiter sa richesse sémantique et traiter efficacement le contenu de ces documents, alors que la structure physique sert juste pour décrire la présentation d'un document.

2.1.3.2. RDF

RDF (Resource Description Framework), est un standard de la famille du W3C, il est conçu pour décrire les ressources Web. RDF peut être utilisé pour décrire le titre, l'auteur, le contenu et les informations des pages Web. Décrire les ressources c'est un axe majeur dans l'activité du Web sémantique, il permet aux applications de stocker, échanger et utiliser des informations lisibles par machine répartie sur tout le Web, ce qui permet aux utilisateurs de récupérer et traiter l'information avec plus d'efficacité et de sécurité.

Un document structuré en RDF est un ensemble de triplets : (Sujet, prédicat, objet) ; Le sujet représente la ressource à décrire, Le prédicat représente un type de propriété applicable à cette ressource, L'objet représente une donnée ou une autre ressource.

L'exemple suivant illustre l'utilisation de certaines des propriétés Dublin Core dans un document RDF

[9] www.w3c.org

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc= "http://purl.org/dc/elements/1.1/">

<rdf:Description rdf:about="http://www.w3schools.com">
  <dc:description>W3Schools - Free tutorials</dc:description>
  <dc:publisher>Refsnes Data as</dc:publisher>
  <dc:date>2008-09-01</dc:date>
  <dc:type>Web Development</dc:type>
  <dc:format>text/html</dc:format>
  <dc:language>en</dc:language>
</rdf:Description>
</rdf:RDF>
```

Figure 2.4 : Exemple d'un document RDF (W3C)

2.1.3.3. SPARQL

SPARQL (**P**rotocol **a**nd **R**DF **Q**uery **L**anguage) (prononcé "sparkle) est un langage de requête sémantique pour les documents en format RDF, capable d'extraire et manipuler des données stockées dans le format RDF. Il est devenu un Standard dans le groupe W3C depuis 2008, et il est reconnu comme l'une des technologies clés du web sémantique [HEB 09]

2.1.3.4. OWL

Le langage d'ontologie Web (OWL) est un langage de représentation de connaissances pour la création des ontologies. Les ontologies sont une façon formelle pour décrire la structure des connaissances de différents domaines: les noms représentant les classes d'objets et les verbes représentant les relations entre les objets. [KNU 09]

2.3. Ressources sémantiques (les ontologies)

Une ressource sémantique est un vocabulaire contrôlé. C'est une liste de termes d'un domaine ou de plusieurs domaines qui ont été énumérés explicitement. Cette liste est contrôlée par un expert. Chaque terme du vocabulaire doit avoir (théoriquement) une définition non ambiguë et non redondante (dans la pratique n'est pas toujours vrai). Il existent différents types de ressources sémantiques tel que les taxonomies, les thesaurus, les ontologies, glossaires et dictionnaires. Les gens utilisent dans la plupart du temps le mot ontologie pour indiquer ces différents concepts.

2.3.1. Taxonomie

C'est la technique utilisée pour faire un classement dans un système hiérarchique des organismes. Une taxonomie est une collection de termes organisés en une structure hiérarchique. Les termes dans une taxonomie sont reliés entre eux par des relations parent-enfant. Il peut y avoir différents types de relations parent-enfant dans une taxonomie (par exemple, tout-partie, genre-espèces, Type-Instance), mais une bonne pratique limite les relations parent-enfant [VIC 13]

Computational and artificial intelligence
....Artificial intelligence
.....Context awareness
.....Cooperative systems
.....Decision support systems
.....Intelligent systems
.....Intelligent robots
.....Knowledge based systems
.....Expert systems
.....Mobile agents

Figure 2.5 : Une partie de la taxonomie du IEEE [10](2014)

2.3.2. Thesaurus

Un thésaurus est un réseau de termes d'un vocabulaire contrôlé. Il utilise d'autres relations entre les termes, en plus de la relation parent-enfant.

Dans le domaine de recherche d'information un thésaurus cherche à dicter les relations sémantiques entre les termes pendant la phase d'indexation. Un thésaurus sert à minimiser l'ambiguïté sémantique en garantissant l'homogénéité et la cohérence dans le stockage et la récupération des termes du contenu. [ANS 05]

Un thésaurus sert à guider à la fois la phase d'indexation et la phase de recherche en sélectionnant le meilleur terme ou la meilleure combinaison de termes qui représente un sujet donné.

[10] www.ieee.org

PREFERRED TERM	stone fruits
DEFINITION	Fruits of the botanical family Rosaceae that contain a single hard seed, called a stone, pit, or pip. The term includes plums, cherries, greengages, peaches, apricots, almonds, and sloes.
BROADER CONCPT •	fruits
NARROWER CONCEPTS	apricots
•	cherries
•	dates
•	nectarines
•	olives
•	peaches
•	plums
IN OTHER LANGUAGES	فواكه منواة Arabic
	核果类 Chinese
	peckoviny Czech

Figure 2.6 : Exemple du thesaurus AGROVOC[11]

L'organisation internationale de normalisation ISO[12], définit un thésaurus pour la recherche d'information, comme un vocabulaire "contrôlé et structuré dans lequel les concepts sont représentés par des termes, organisées afin que les relations entre les concepts sont rendues explicites ".

Un thésaurus est composé d'au moins trois éléments : une liste de mots, une relation entre les mots, et un ensemble de règles sur la façon d'utiliser le thésaurus.

Il existe des thesaurus spécialisés dans un domaine précis tels que MeSH (domaine biomédical), et des thesaurus généralistes comme WordNet.

2.3.2.1. Composants d'un thesaurus

Les composants principaux d'un thesaurus sont [ISO 13] :

1. Les Concepts
2. Les Termes

[11] http://aims.fao.org/

[12] www.**iso**.org

3. Les Relations entre concepts et entre concepts et termes
4. Les Regroupements de concepts thématiques ou par facettes

……lait
……(lait par matière grasse)
……lait entier
……lait demi-écrémé
……lait écrémé
……(lait par source animale)
……lait de bufflonne
……lait de vache
……lait de chèvre
……lait de brebis
……(lait par type de traitement)
……lait condensé
……lait évaporé
……lait homogénéisé
……lait pasteurisé
……lait stérilisé

Figure 2.7 :Exemple d'un regroupement par facettes [ISO 13]

2.3.2.2. WordNet

Le WordNet [Mil 95] est une grande, base de données lexicale pour la langue anglaise lisible par les machines. En raison de sa conception et sa large couverture, cette ressource a trouvé une large acceptation dans le domaine de la linguistique [FEL 05].

Un réseau sémantique comme WordNet est une tentative pour générer un modèle dans lequel les concepts et les mots pourraient être organisées, avec leurs significations et les relations sémantiques entre ces concepts [FEL 05].

L'élément le plus fondamental qui compose WordNet est le synset, dont le nom dérive de "set of synonyms" qui signifie ensemble de synonymes. Il se compose d'un groupe de mots synonymes, qui ont un sens commun dans le même contexte. chaque synset a des relations sémantiques qui sont utilisées pour le relier à d'autres synsets, ce qui nous fournit un réseau dense qui favorise l'expression de la connaissance sémantique d'un certain mot dans un contexte donné.

A titre d'exemple, le mot "dog" peut avoir 8 significations différentes (7 significations comme non et une signification comme verbe).

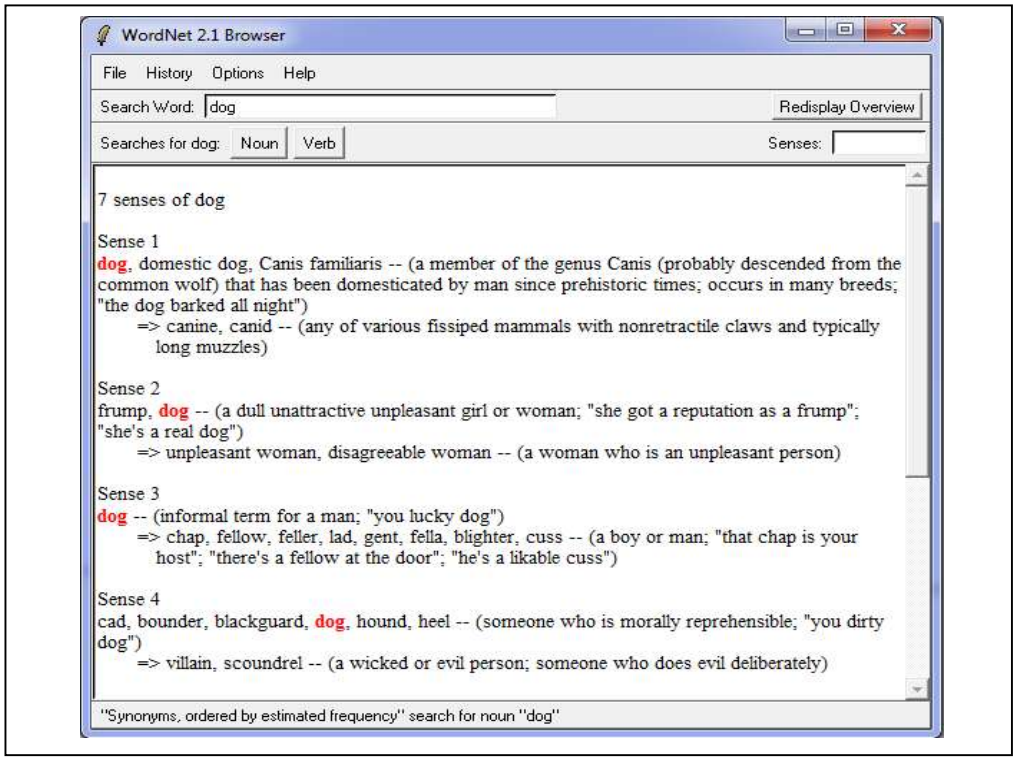


Figure 2.8 : un exemple de WordNet (version bureau).

2.3.2.2.1. Les relations dans WordNet

WordNet tient compte du fait que la définition d'un mot peut être perçue selon d'autres termes avec les quels il est liée [Mil 95]. Ces relations sont la caractéristique la plus importante fournie par WordNet et qui la distingue des autres bases de données lexicales disponibles. Les relations sémantiques disponibles dans WordNet sont : : Super-subordonné et partie-tout, antonymie, similitude.

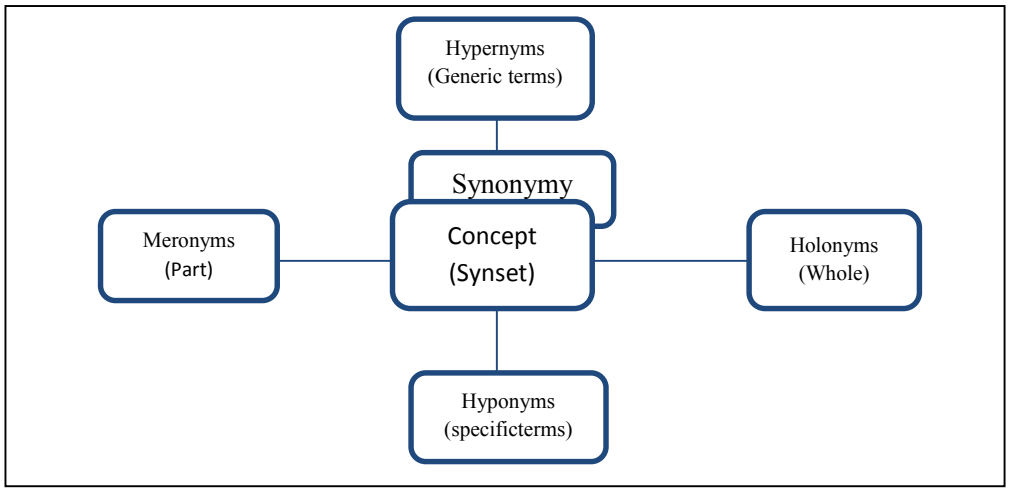


Figure 2.9**.** Relations sémantiques dans WordNet [Baz 2005].

La version de WordNet (3.1) contient un total de 117,659 synsets, représentant des relations sémantiques entre les mots et les sens de mots.

La relation la plus fréquemment utilisée entre les synsets est la relation super-subordonnés, aussi appelé **hyperonymie** (dénotant qu'un synset est un type plus générale d'un autre synset).

Hyponymie ou la relation EST-UN (dénotant qu'un synset est un sous-type d'un autre synset). Il est utilisé pour relier synsets à synsets plus générales. Par exemple il est possible de dire que «un chien est un type de mammifère» et que «un mammifère est un type d'animal». Dans cet exemple, l'animal est hyperonyme de mammifère, qui à son tour est le hyperonyme de chien. A l'inverse, un chien est hyponyme de mammifère, qui est une hyponyme d'animal.

Une autre relation très courante appelée partie-tout, aussi connu comme **Holonymie** (dénotant qu'un synset est une partie d'un autre synset) ou **Méronymie** (indiquant qu'une synset est composé d'autres synsets). Il décrit la relation de composition d'un synset à l'égard des autres synsets

Par exemple le Clavier est un **Holonymie (partie de)** d'un Ordinateur et la clés est un **Méronymie** du clavier.

2.3.2.2.2. Une ressource pour la désambiguïsation

WordNet a été largement utilisé comme ressource pour les techniques du traitement automatique de la langue. La langue naturelle a plusieurs mots qui portent un ensemble de significations (sens). Lors de la rédaction d'un texte, le sens voulu pour chacun de ces mots peut être déterminée par le lecteur selon le contexte où ils sont utilisés. Au contraire des êtres humains, les machines doivent traiter l'information textuelle et l'analyser afin de déterminer le sens correspondant [LEA 13].

Désambiguïsation (qui sera détaillé dans la section 4) consiste à déterminer le meilleur sens approprié pour un mot donné dans un contexte, en utilisant des méthodes de calcul. La désambiguïsation est un sujet de recherche en linguistique informatique et traitement automatique du langage naturel [NAV 12]. Elle est considéré comme un axe dans l'Intelligence Artificielle (IA).

2.3.3. Ontologie

Le terme **Ontologie** a une longue histoire dans le domaine de la philosophie, c'est une notion qui s'intéresse à l'étude de l'existence de l'être humain ou à supposer son existance pour convaincre ou arriver à la vérité.

Récemment, le terme a été utilisé pour décrire les objets qui peuvent exister dans un domaine particulier et indiquer les connaissances partagées par les personnes qui travaillent dans un domaine particulier. [AHC 05].

En informatique, cette notion est apparue dans les années 90, avec l'émergence de l'Ingénierie des Connaissances (IC), les ontologies sont apparues comme des réponses aux problématiques de représentation et de manipulation des connaissances au sein des systèmes informatiques.

2.3.3.1. définition

Plusieurs définitions ont été proposées, dont la plus couramment citée est celle de Gruber (En 1993) : « Une spécification explicite et formelle d'une conceptualisation faisant l'objet d'un consensus » [GRU 93].

En 1997, Borst modifie légèrement la définition proposée par Gruber en énonçant : « Une ontologie est définit comme étant une spécification formelle d'une conceptualisation partagée » [BRO 97].

Le terme **conceptualisation** signifie un modèle abstrait, qui est une représentation simplifiée d'un domaine. Le terme **explicite** signifie que l'ensemble des concepts utilisés sont définis d'une façon explicite. L'adjectif **formel** précise que l'ontologie construite doit être lisible par un ordinateur et, le terme **commun** montre qu'une ontologie fournit un vocabulaire conceptuel commun et une compréhension partagée par la communauté visée. [AHC 05].

Dans le contexte de l'informatique, une ontologie définit un ensemble de primitives de représentation qui permet de modéliser un domaine de connaissances. Les primitives de représentation sont généralement les classes (ou ensembles), les attributs (ou propriétés), et les relations (ou les relations entre les membres de la classe).

Une ontologie est une forme de connaissance et un moyen d'expression pour les êtres humains pour qu'ils puissent communiquer avec les machines.

Les ontologies permettent de décrire un ensemble d'axiomes et de règles qui permettent de générer de nouvelles connaissances. Cette caractéristique est absente dans les autres types de vocabulaires (taxonomies, thesaurus,...).

2.3.3.2. PROTÉGÉ-2000: outils de construction d'ontologie

Protégé[13] est un logiciel gratuit, c'est un éditeur d'ontologie open source et un système d'acquisition de connaissances. Protégé fournit une interface graphique pour définir des ontologies. Il comprend également des mécanismes pour valider la cohérence des modèles sont cohérents et en déduire de nouvelles informations sur la base de l'analyse d'une ontologie. Cette application est écrite en Java. Protégé est développé à l'Université de Stanford.

[13] www.**protege**.stanford.edu

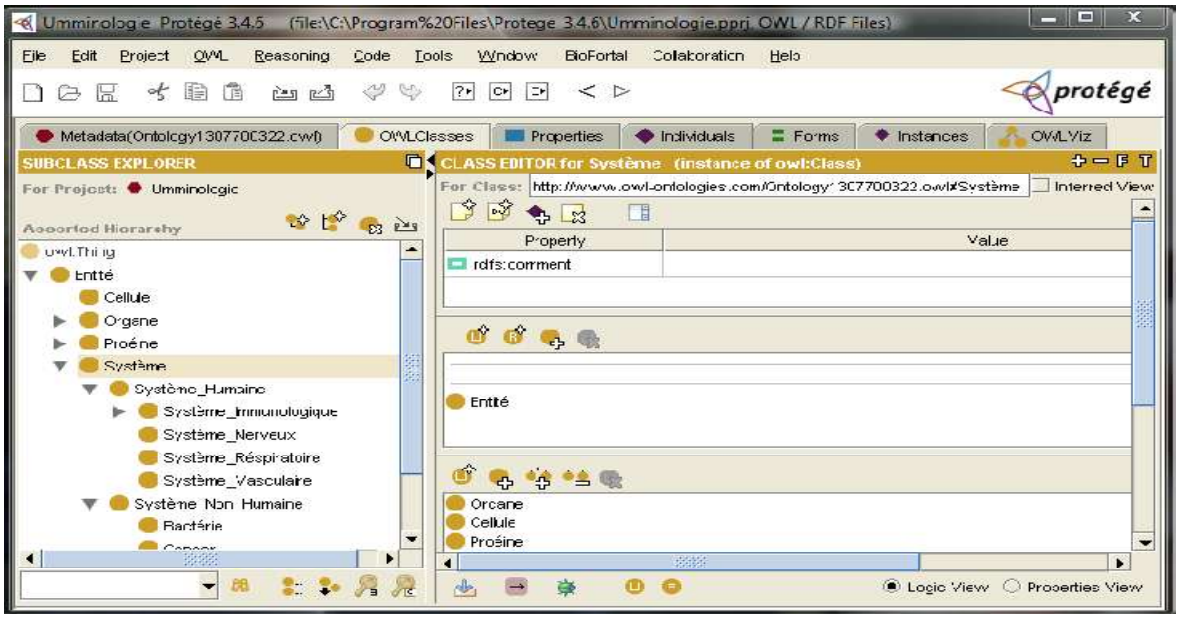


Figure 2.10: Interface graphique de Protégé 3.4.6

2.3.3.3. Outils de manipulation d'ontologies: Jena

Apache Jena[14] est une plate-forme libre et open-source en langage Java pour la construction de web sémantique et les applications de données liées. La plate-forme est composé de différentes API qui interagissent ensemble pour traiter les données RDF.

2.3.3.4. Exemple d'ontologies

2.3.3.5.1. YAGO

YAGO [SUC 07], est une ontologie de grande couverture et une précision très élevée (95 %). YAGO a été automatiquement extraite et combiné de Wikipédia et WordNet. elle comprend les entités et les relations, et contient actuellement plus de 2 millions d'entités (personnes, villes...), et 20 millions de faits sur les entités.

Elle contient la relation hiérarchique **Est-Un,** ainsi que les relations sémantiques entre entités. Les faits pour YAGO ont été extraits du système de catégories et les info_boxes de Wikipedia et ont été combinés avec les relations de WordNet. YAGO est basé sur un modèle logique qui permet de représenter les relations n-aire, avec un modèle de requête puissant facilite l'accès aux données du YAGO.

2.3.3.5.2. DEPEDIA

DBpedia est le résultat d'un effort d'une communauté pour extraire des informations structurées à partir de Wikipedia et de rendre cette information disponible sur le Web.

[14] https://**jena**.apache.org

DBpedia permet d'interroger, avec des requêtes complexes, les données de Wikipedia et permet aussi de lier avec Wikipedia, d'autres ressources de données qui se trouvent sur le Web. [ARN 12].

Les articles de Wikipédia sont disponibles dans plus de 250 langues, avec la version anglaise qui est la plus utilisée. Toutes les ressources stockées dans DBpedia décrivent environ 2,6 millions d'entités, (incluant 213 000 personnes, 328 000 lieux, 57 000 albums musicaux, 36 000 films, 20 000 entrepris). La base de connaissances totalise ainsi 274 millions de triplets RDF et représente 609 000 liens vers des images, 3 150 000 liens vers des pages Web externes, 4 878 100 liens vers des données RDF externes. Ces informations sont organisées dans 415 000 catégories de Wikipedia. [ARN 12]. DBpedia est le cœur du projet LOD (Linking Open Data) du W3C.

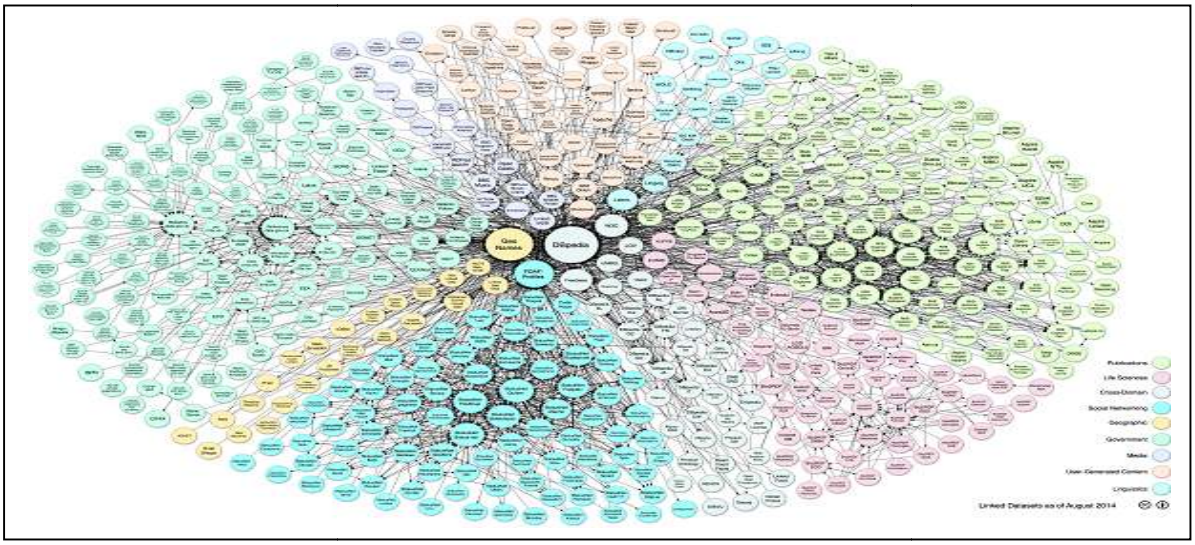


Figure 2.11: représentation des liens de DBpedia (LOD Cloud) [15]-

2.4. Traitement automatique de la langue

C'est une discipline de l'intelligence artificielle et de la linguistique qui se développe depuis les années 1960. Son objectif est de développer des techniques et des méthodes pour la compréhension automatique de la langue naturelle.

La langue naturelle, est un outil qi permet aux êtres humains de s'exprimer, elle possède des propriétés spécifiques qui sont la variation linguistique et l'ambiguïté. La variation linguistique nous donne la possibilité d'utiliser différents mots ou expressions pour communiquer la même idée. L'ambiguïté linguistique c'est quand un mot ou une phrase a plusieurs interprétations. Ces deux propriétés compliquent la tache des machines pour traiter les documents textuels. [MAR 07]

[15] www.dbpedia.org

2.4.1. Le traitement statistique de la langue naturelle

Le traitement statistique de la langue naturelle est la méthode classique pour modéliser et traiter le contenu textuel des documents. Le modèle de traitement de documents comprend deux étapes suivantes :

a) Le Prétraitement : est utilisé fondamentalement dans la préparation des documents, en éliminant tous les éléments considérés comme inutiles. Cette étape se compose de quatre phases principales.

- Elimination du document les éléments qui ne représente pas le contenu du texte, comme les balises dans un document XML.
- La standardisation du texte, qui consiste à homogénéiser l'ensemble du texte en éliminant les majuscules, les mots vides, ainsi qu'en identifiant les paramètres spécifiques comme chiffres ou dates ; sigles ou acronymes.
- La lemmatisation, qui tente de déterminer la base (lemme) de chaque mot dans un texte. La quatrième phase, consiste à identifier les N-Grams qui sont les mots composés, les noms propres, etc.. pour être en mesure de les traiter comme une unité conceptuelle unique (ex union européen)

b) Paramétrage: c'est le stade de complexité minimale une fois que les termes pertinents ont été identifiés. Cela consiste à quantifier des caractéristiques du document. Une des méthodes les plus utilisées pour estimer l'importance d'un terme c'est le système de tf.idf [MAR 07]

2.4.2. Traitement linguistique de la langue naturelle

Cette approche est basée sur l'application de différentes techniques et règles qui permettent d'extraire les connaissances linguistiques. Le traitement linguistique passe par différentes étapes : L'analyse morphologique qui est effectuée par les tagueurs qui attribuent chaque mot à une catégorie grammaticale (nom, verbe, adjectif...). La deuxième étape consiste à identifier les grandes unités grammaticales, expressions et phrases. La troisième étape a comme objectif l'obtention d'une représentation sémantique de la phrase à partir des éléments qui le composent. Un des outils les plus souvent utilisé dans le traitement sémantique est la base de données lexicographique WordNet. [MAR 07]

2.4.3. Etiquetage morphosyntaxique

L'étiquetage morphosyntaxique (part-of-speech tagging en anglais) est le processus qui permet de marquer pour chaque mot dans un texte, les informations grammaticales correspondantes, en fonction à la fois de sa définition, et de son contexte. Les formes grammaticales les plus simples dans l'identification des mots sont les : noms, verbes, adjectifs, adverbes, etc.

Ils existent plusieurs outils pour l'étiquetage morphosyntaxique parmi ces outils TreeTagger pour la langue française, et Stanford Tagger pour l'anglais.

2.4.4. désambiguïsation lexicale

C'st un problème dans le domaine de traitement des langues naturelles et de l'ontologie. C'est la détermination du sens d'un mot dans une phrase lorsque ce mot peut avoir plusieurs sens possibles. La désambiguïsation intervient dans plusieurs domaine comme la traduction automatique où un mot anglais comme grid, peut être traduit en français (grille, réseau, gâchette) selon le contexte. [NAN 98].

La désambiguïsation repose sur deux étapes principales, la première étape permet d'extraire la liste de sens pour chaque mot. La deuxième étape consiste à définir le sens exact du mot en étudiant le discours et le contexte dans lequel ce mot apparaît, par l'exploitation d'une ressource sémantique externe. Pour déterminer le sens exact d'un mot dans un contexte, il existe des techniques qui permettent de déterminer la similarité sémantique entre les mots. Ces techniques seront détaillées dans la section suivante.

2.5. Similarité sémantique

La similarité sémantique entre les concepts est une méthode de mesure de la distance sémantique entre deux concepts en fonction d'une ontologie. La similarité sémantique est utilisé pour identifier les concepts ayant des «caractéristiques» communes. Les méthodes de similarité sémantique sont intensivement utilisés dans la plupart des applications des systèmes à base de connaissances, dans la recherche sémantique d'information (identifier la meilleure combinaison de sens entre les termes de la requête), et dans la désambiguïsation du sens.

2.5.1. Types de mesure de similarité sémantique

Les mesures de similarité peuvent être affectées par les caractéristiques communes des concepts comparés. Les différences entre les concepts entrainent la diminution ou l'augmentation de la distance sémantique entre ces concepts.

En outre, les mesures de similarité sont liées à la taxonomie, la similarité est affectée par la position des concepts dans la taxonomie et le nombre de liens hiérarchiques. On peut regrouper les méthodes de calculs de similarité en trois catégories :

2.5.1.1. Calcul de similarité par le nombre d'arcs

C'est une technique de calcul basée sur les liens taxonomiques « est un ». Ils existent plusieurs études et méthode qui ont utilisé cette technique tels que :

2.5.1.1.1. La mesure de Wu-Palmer [WU 94]

Cette mesure de similarité observe la position de C1 et C2 dans la taxonomie par rapport à la position du concept commun C qui les subsume. Le concept parent a considéré est le concept commun le plus proche ancêtre commun (l'ancêtre commun lié avec le nombre minimum de liens est-A).

$$SIM(C1, C2) = 2*N /N1+N2+ 2*N$$

Où N1 et N2 sont la distance (nombre de liens est-A) qui sépare, respectivement, le concept C1 et C2 du concept commun spécifique et N est la distance qui sépare le plus proche parent commun de C1 et C2 à partir du nœud racine.

2.5.1.1.2. La mesure de Leacock et Chorodow [LEA 94]

La mesure proposé par [LEA 94] a la formule suivante :

$$SIM(C1, C2) = -Log (min Len (C1, C2) / 2* D)$$

Où min Len ($c1$, $c2$) est la longueur du plus court chemin entre c1 et c2 et D est la profondeur maximale de l'ontologie

Selon cette mesure, le plus court chemin entre deux concepts de l'ontologie restreinte aux liens taxonomiques est normalisée par l'introduction d'une division par le double de la profondeur maximale de la hiérarchie.

2.5.1.2. Calcul de similarité par le contenu informatif

2.5.1.2.1. La mesure de Resnik [RES 99]

Cette mesure utilise le contenu de l'information des parents partagés. Le principe de cette mesure est traduit par: deux concepts sont plus similaires si ils présentent une information plus partagée, et les informations partagées par les deux concepts C1 et C2 sont signalés par le contenu de l'information des concepts qui les englobe dans la taxonomie.

2.5.1.2.2. La mesure de Lin [Lin, 93]

La similarité entre deux concepts est mesurée par le rapport du contenu d'information partagées par les deux concepts C1 et C2, sur le montant du contenu d'information nécessaire pour décrire chacun des deux concepts.

2.6. Conclusion

Nous avons présenté dans ce chapitre les concepts fondamentaux du web sémantique et de la modélisation sémantique des documents. Nous avons donné un aperçu général sur toutes les notions liées à la sémantique et au texte, comme les ontologies, le traitement automatique de la langue et la similarité sémantique entre les termes d'un texte.

Ces concepts sont indispensables pour le traitement sémantique et automatique des documents numériques, notamment dans le domaine de recherche d'information, ou l'exploitation de la sémantique permet d'améliorer les résultats de la recherche.

Dans le chapitre suivant nous allons détaillé les technique utilisé pour intégrer la sémantique dans la recherche d'information.

Chapitre 3

Modélisation des documents numérique: Indexation et recherche

3.1. Introduction

Ce chapitre présente un état de l'art sur la recherche d'information, où nous allons introduire tous les concepts et notions utilisé dans ce domaine. Les systèmes de recherche d'information (RIS) permettent de modéliser le contenu des documents d'une manière automatique et efficace afin de retourner la réponse pertinente à la requête et aux besoins des utilisateurs. trois concepts essentiels autour des quels tournent ces systèmes sont: Documents, requêtes, et pertinences.

Le domaine de recherche d'information a pris un grand essor avec l'apparition et le développement de l'internet, où une quantité exponentielle de documents de tout genre (une partie de texte, page web, une image, une vidéo) est stocké chaque jour.

Les documents textuels peuvent êtres sans structuration (un texte plat) ou avec une structure tels que les documents XML. ce qui a aidé les systèmes de recherche à améliorer leurs résultats en prenant en compte cette structure.

Un axe récent dans le domaine de recherche d'information qui est la recherche sémantique a pris une place importante ces dernières années.

3.2. Définition de la recherche d'information

[CHR 09] a définit la recherche d'information (RI) comme suit : c'est l'opération de trouver un matériels (habituellement un document) de nature non structuré (habituellement un texte) qui satisfait un besoin d'information à partir d'une grande collection (généralement stockée sur les ordinateurs).

L'objectif des systèmes de recherche d'information est de retourner des informations (sous forme de document ou une partie de document) selon le besoin de l'utilisateurs. Ce besoin est formulé sous forme d'une requête.

Le domaine de (RI) traite les concepts suivants : la représentation, le stockage et l'accès aux documents [BAE 99]. La recherche d'information est devenu ces dernières années le moyen le plus utilisé pour l'accès à l'information, principalement en raison de l'augmentation considérable de documents dans le web.

3.3. Recherche d'information classique

Les systèmes de recherche d'information sont composés de trois fonctions principales: l'indexation et la représentation des documents, la représentation de la requête utilisateurs, et l'appariement requête-document. Le processus de recherche d'information sera détaillé dans la section suivante.

3.3.1. Processus de recherche d'information

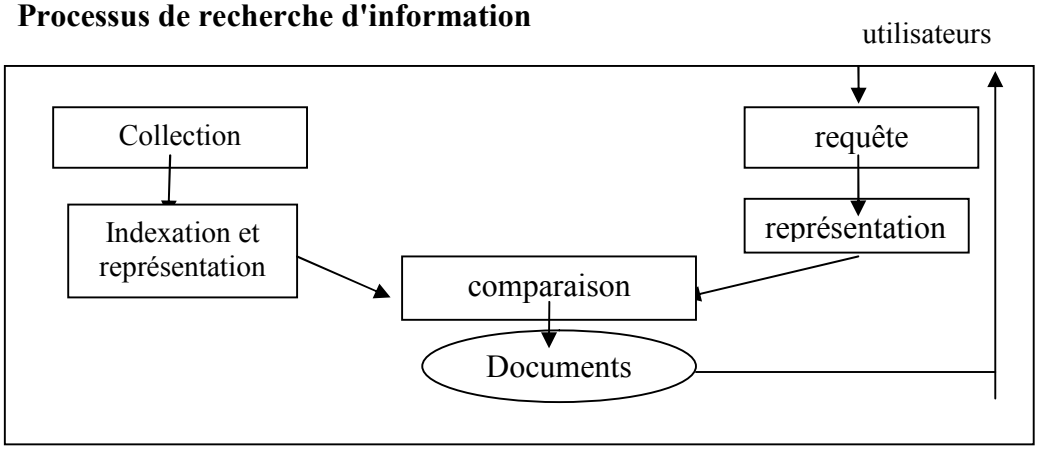


Figure 3.1: Processus de recherche d'information

3.3.1.1. Indexation et représentation

La représentation d'un document ou d'une requête permit d'analyser et de traiter le document (requête) afin d'extraire les mots clés (simple ou composé) qui représente bien le contenu du document. Pour extraire ces termes un processus de prétraitement est nécessaire. Ce processus contient les étapes suivantes: analyse lexicale, l'élimination des mots vides, et la lemmatisation.

Chaque terme extrait aura un poids qui définit son importance dans le document, et son importance dans la collection de document. Ce poids est calculé avec l'une des technique de pondération.

3.3.1.1.1. Analyse lexicale

L'analyse lexicale (tokenization en anglais) est le processus qui permet de couper un texte en un ensemble de mots, de phrases, ou de symboles. chaque mot ou phrase est un ensemble de caractère qui ont un sens.

3.3.1.1.2. L'élimination des mots vides

C'est la suppression des mots qui sont trop fréquents dans les documents de la collection et, par conséquent, n'affectent pas le sens du contenu. Ces mots sont ceux qui se répètent pratiquement dans tous les documents de la collection, ils ne sont pas considérés comme des termes à indexer. Des articles, prépositions et conjonctions sont des candidats naturels pour une liste de mots vides. [BAE 99].

Il existe deux techniques pour éliminer les mots vides: soit L'utilisation d'une liste de mots vides, ou bien l'élimination des mots qui dépassent un certain nombre d'occurrences dans la collection.

3.3.1.1.3. Lemmatisation

Elle se réfère à la transformation d'un mot (terme) à sa base ou sa racine. L' utilisateur spécifie Souvent, un mot dans une requête, mais un document pertinent ne sera pas retourné si une variante de ce mot existe dans le document. Les préfixes et les suffixes sont des exemples des variations syntaxiques qui empêchent un mappage parfait entre les termes de la requête et ceux du document correspondant. [BAE 99].

Les algorithmes de lemmatisation permettent de réduire les termes similaires à leur forme de racine commune. Par exemple Les termes suivant : «informé», «informer», «informe» seront représentés par le lemme «informe». [BAE 99]. L'algorithme le plus simple et le plus largement utilisée est l'algorithme de lemmatisation de Porter [POR 80].

3.3.1.1.4. Pondération des termes

La pondération permet d'associer à chaque termes un poids qui représente son importance dans la collection de documents. il existe deux type de pondération: locale et globale. La pondération locale permet de connaitre l'importance d'un terme dans un seul document, tandis que la pondération globale permet de connaitre l'importance du terme dans toute la collection. la pondération globale permet de réduire l'importance d'un terme non expressif et qui apparait fréquemment dans la collection

La plupart des techniques de pondération sont basées sur les facteurs TF et IDF*:*

TF (*Term Frequency*) : cette mesure permet de calculer l'importance locale du terme dans un document. la méthode la plus simple de calcule est celle qui permet de déterminer le nombre d'occurrence du termes dans le document.

IDF (*Inverse of Document Frequency*) : C'est la pondération globale des termes. Elle permet le calcule de l'importance d'un terme dans toute la collection. Elle permet de réduire l'importance des termes qui se trouvent souvent dans un document, mais qui ne représentent pas le sens globale du document. ces termes apparaissent fréquemment dans toute la collection

Exemple:

Voilà un tableau qui représente comment est calculé le TF*IDF des trois termes dans chaque document de la collection (10 documents dans la collection).

TF est le nombre d'occurrences d'un terme dans un document

IDF est calculé par la formule : (Log (N/n)) où N représente le nombre totale de documents dans la collection(dans l'exemple suivant 10 documents), et n est le nombre de documents qui contient le terme.

IDF permit de réduire l'importance d'un terme qui apparait fréquemment dans tous les documents. On remarque que le terme "Beaucoup", malgré son TF élevé dans le document 3, mais son TF*IDF sera réduit par rapport au TF*IDF du terme "XML" dans le document 8. Les documents qui contient fréquemment les termes "XML" et "recherche" seront mieux classé que les documents qui contient seulement le terme "Beaucoup"

N° Document	Beaucoup	xml	recherche	tf*idf Beaucoup	tf*idf xml	tf*idf recherche	SOMME	classement
1	20	1	1	0,91515	0,2218487	0,301029996	1,438029	10
2	5	4	15	0,228787	0,887395	4,515449935	5,631632	2
3	26	12	2	1,189695	2,662185	0,602059991	4,45394	5
4	5	10	10	0,228787	2,2184875	3,010299957	5,457575	3
5	32	0	0	1,46424	0	0	1,46424	9
6	35	0	0	1,601512	0	0	1,601512	8
7	80	0	0	3,660599	0	0	3,660599	6
8	15	22	0	0,686362	4,2151262	0	4,901489	4
9	20	20	2	0,91515	4,436975	0,602059991	5,954185	1
10	0	9	0	0	1,9966387	0	1,996639	7
idf (Log (N/n))	0,045757	0,2	0,30103					

Si un terme apparait dans toute la collection son IDF sera 0 alors ce terme sera éliminer (il devient mot vide)

3.3.1.2. Appariement document-requête

3.3.1.2.1. Le modèle booléen

Le modèle booléen est le modèle classique le plus ancien pour la recherche d'information. Ce modèle est basé sur l'utilisation de la logique booléenne traditionnelle et la théorie des ensembles pour identifier les documents qui correspondent à une requête. Les documents et les requêtes sont représentés sous formes des expression logique, ou les connecteurs logique sont utilisés (ET, OU, NON). [SAL 83]

Ce modèle permet de faire un appariement exacte entre la requête et les documents en déterminant si les termes de la requête existent ou non

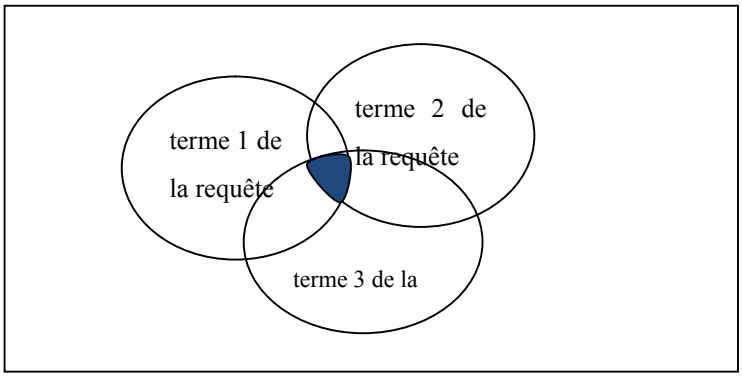


Figure 3.2 : appartenance des terme d'une requête dans un document [SAL 83]

Ce modèle est simple et très facile à réaliser, mais son inconvénient majeur c'est qu'il ne retourne pas des documents pertinents qui correspondent partiellement à la requête , comme il ne permet pas de classer les documents selon leur degré de pertinence.

3.3.1.2.2. Le modèle vectoriel

Le modèle vectoriel [SAL 75] est l'un des plus utilisé dans la recherche d'information. Dans ce modèle, les documents et les requêtes sont considérés comme des vecteurs dans un espace de terme.

Pour calculer la similarité entre le vecteur requête et le vecteur document, on utilise la fonction de distance euclidienne (similarité cosinus). Après le calcule de la distance entre chaque document et la requête, un document sera classé au premier rang s'il est plus proche du vecteur requête.

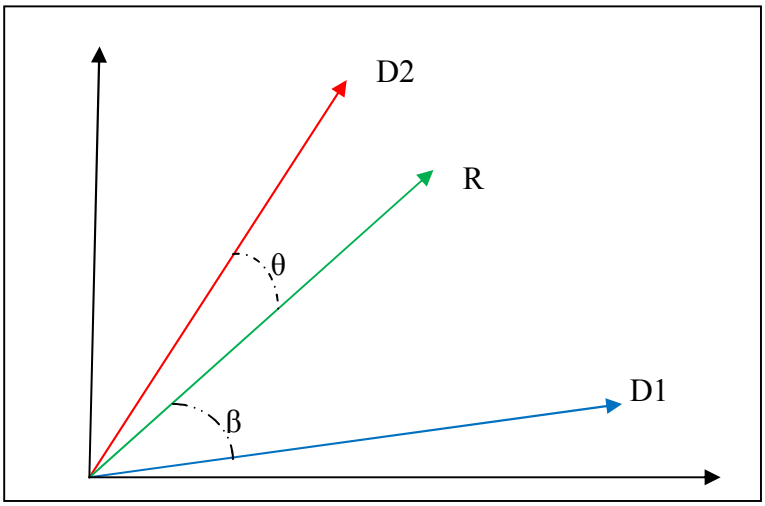


Figure 3.3. Le modèle vectoriel

Pour calculer la distance entre le vecteur du Document D2 et le vecteur de la requête R on doit calculer le cosinus de θ [BAZ 05]:

$\cos\theta = \frac{D2 \,.R}{||D2||.||R||}$ ou D2 .R est le produit scalaire entre les deux vecteurs (D2 et R)

et ||D2|| est la norme du vecteur D2

Chaque document est représenté par un vecteur: D_j (w_{1j}, w_{2j}, ..., w_{Nj}) ou w_{ij} représente le poids d'un terme i dans le document j, et N représente le nombre totale de termes dans le corpus. La requête R est représenté par le vecteur : q(w_{1R}, w_{2R}, ..., w_{NR})

Pour calculer le poids de chaque terme dans le document (w_{ij}) la méthode la plus utilisé est le TF*IDF [SAL 75] :

$$w_{ij} = TF_{ij} \cdot \log \frac{D}{d}$$

ou TF représente le nombre d'occurrence d'un terme i dans le document j et $\log \frac{D}{d}$ représente l'IDF où D est le nombre totale de document dans le corpus et d est le nombre de document qui contient le terme i

Alors pour calculer la similarité entre la requête R et un document D2 on doit utiliser la formule suivante :

$$\text{similarité (D, R)} = \frac{D2 \,.R}{||D2||.||R||} = \frac{\sum_{i=1}^{N} w_{i,D2} \,.w_{i,R}}{\sqrt{\sum_{i=1}^{N} w_{i,D2}^2}\sqrt{\sum_{i=1}^{N} w_{i,R}^2}}$$

3.3.1.2.3. Le modèle probabiliste

Le modèle probabiliste de recherche est basée sur le principe de la probabilité. il consiste à classer les résultats selon la probabilité de pertinence d'un document par rapport à une requête [HAL 07]. Les travaux les plus connus et les plus exploités dans ce modèle son ceux de [ROB 77] qui se basent sur la notion de "principe de classement probabiliste" qui indique que les documents sont classés selon leur probabilité de pertinence. Cette probabilité est estimé sur la base de la distribution des termes pertinents et non pertinents dans le documents:

[ROB 77] propose deux classes de documents D_j: pertinents (classe R) ou non pertinents (classe $\overline{R}$) par rapport à une requête Q :

- P(R/D) est la probabilité que D est pertinent pour Q

- P($\overline{R}$/D) est probabilité que D n'est pas pertinent pour Q.

Pour calculer le score d'un document le modèle utilise la formule suivante :

$$S = \frac{P(R/D)}{P(\overline{R}/D)}$$

Robertson [ROB 94] a proposé la fameuse formule BM25 :

Soit une requête Q qui content les terme q_1.....q_n :

$$S(Q,D) = \sum_{i=1}^{n} IDF(q_i) . \frac{TF_{i,D}\,(k+1)}{TF_{i,D} + k.(1-b+b.\frac{|D|}{avgdl})}$$

$$IDF(q_i) = \frac{N - n_i + 0.5}{n_i + 0.5}$$

ou : N représente le nombre totale de documents et n_i le nombre de document qui contient le terme i. |D| la taille du document et avgdl est la taille moyenne des documents de la collection. k et b sont des paramètres où $k \in [1.2, 2]$ et b=0.75

3.3.1.3. Reformulation de requêtes

Le problème de l'imprécision de la requête formulée par les déférents utilisateurs, pose un grand problème aux moteurs de recherche d'information. les utilisateurs souvent non expérimentés, qui n'arrivent pas à exprimer leur besoin, utilisent des termes qui ne correspondent pas aux documents pertinents. [BAZ 05]

Pour pallier à ce problème, des techniques sont utilisées pour l'expansion automatique de la requête en ajoutant des termes supplémentaire à la requête initiale afin d'améliorer le nombre de documents pertinentes dans le résultat final de la recherche.

Il existe plusieurs méthodes pour l'expansion de requête. parmi celles-ci il faut citer la méthode qui exploite le profil utilisateur afin de proposer d'autre termes proches aux termes initiaux de la requête. Une autre méthode utilise les requêtes déjà posées par d'autres utilisateurs et les proposé comme des requêtes proche de la requête initial (comme recherche associé de Google[16]). Une troisième méthode utilise la technique feedback (retour en français); ou l'utilisateur après un premier résultat de recherche, peut choisir, parmi les documents trouvés, les documents les plus pertinents. Le système analyse ces documents, il essaye d'extraire des termes et les injecte dans la requête initial.

3.3.2. Evaluation

Il existe plusieurs modèles de recherche d'information qui utilisent de différentes techniques et méthodes. il est difficile de connaitre quelle est la meilleure technique pour une telle collection de documents. Pour ce faire, un axe de recherche indépendant, est créé et qui s'intéresse à l'évaluation des systèmes de recherche d'information et comparer entre les résultats de chaque modèle. [CHR 09] [HAL 07]

L'évaluation est effectuée en utilisant une collection de documents, un ensemble de requêtes décrivant les besoins d'information d'un utilisateur et un ensemble de jugements de pertinence, indiquant, pour chaque requêtes, quels sont les documents (annotées manuellement) pertinents. [RON 08]

Alors pour mesurer la performance d'un modèle de recherche on a besoin de [CHR 09] :

1. Une collection de documents
2. une série de requêtes qui expriment le besoin d'un utilisateur

[16] www.google.com

3. Un ensemble de jugements pour chaque requête. qui détermine si un document dans la collection est ou non pertinent à la requête

3.3.2.1. Campagnes d'évaluation

La performance des systèmes de recherche d'information est mesurée par des collections de teste. Ces collections déterminent comment les systèmes répondent aux requêtes d'utilisateurs et permettent aussi d'évaluer et de juger ces réponses.

Les collections sont composés de trois parties: les documents, les requêtes et les jugements de pertinence qui prédétermine la pertinence des documents dans la collection par rapport à une requête. [RON 08]

Quelques collections les plus utilisées pour évaluer les système de recherche sont ci-dessous présentées [CHR 09]:

3.3.2.1.1. TREC

Text Retrieval Conference (TREC[17]) est une compagne annuelle d'évaluation des travaux dans le domaine de recherche d'information. Lancé par l'Institute National de Standards et de la technologie (NIST[18]) depuis 1992.

Dans ce cadre, il ya eu de nombreuses taches. chaque tache représente une collection de documents issues de déférentes sources: journaux (comme Wall Street Journal, Associate Press Newswire,...), web (comme Wt2G et Wt10g.....). chaque tache a ses propres documents et ses propres requêtes (topics)

3.3.2.1.2. GOV2

GOV2 représente une tache dans la compagne de TREC, NIST a créé cette collection à partir de documents provenant des site gouvernemental (.gov). Elle est parmi les grande collection de documents utilisé pour l'évaluation des systèmes de recherche. Elle contient plus de 25 millions de page web (.gov), de 426 Go de taille.

L'objectif de la création et de l'utilisation de telle collection était de fournir une analyse approfondie d'un domaine unique, où les documents de la collection sont bien liés entre eux. ces liens peuvent être exploitée. [CHR 09] [YOH 09]

[17] **trec**.nist.gov

[18] www.nist.org

3.3.2.1.3. CLEF

L'Initiative CLEF [19] (Conférence et les laboratoires de l'Evaluation Forum, anciennement connu sous le Cross-Language Evaluation Forum) est un organisme d'auto-organisée, dont la mission principale est de promouvoir la recherche, l'innovation, et le développement de systèmes d'accès à l'information multilingue (les langue européenne)

3.3.2.1.4. REUTERS

Reuters est la plus grande agence de nouvelles au monde. en 2000, elle a publié un corpus d'articles de nouvelles connues comme "Reuters Corpus Volume 1" (RCV1) pour utiliser librement dans la recherche d'information et les systèmes d'apprentissage. Actuellement ce corpus est la collection la plus largement utilisée pour la classification de textes.

Les article sont formatés à l'aide d'un schéma XML conforme qui est basé sur une première version de NewsML, un standard ouvert conçu dans Reuters. RCV1 est classé manuellement en trois catégories générales: le sujet, la région et l'industrie.

3.3.2.1.5. INEX

INEX (INitiative for the Evaluation of XML Retrieval) fondée en 2002 est une compagne d'évaluation des système de recherche de documents XML. elle offre aux chercheurs dans ce domaine la possibilité pour évaluer leurs méthodes et comparer leurs résultats. INEX est une collection de documents XML issue de pages Wikipédia en anglais.

Wikipédia est une encyclopédie multilingue collaborative et gratuite. En 2009, elle contient plus de 50 millions d'articles. La compagne propose aussi des requête (topics), et des jugements de pertinence.

Les documents (articles) XML d'INEX se compose de deux partie (en-tête et corps) et chaque partie contient plusieurs balises (tags). La partie corps contient le texte du document (contenu). Le texte est composé de sections et sous section (dans la balise <sec>) et chaque section à un titre <st> et des paragraphes <p>

[19] http://www.clef-initiative.eu/

<?xml version="1.0" encoding="UTF-8"?>

……..

<title>Computer science **</title>**

<id>5323 **</id>**

……

<bdy>

Computer Science is the scientific and practical approach to computation and its applications. It is the systematic study of the feasibility, structure, expression, ……

<sec>

<st> History </st>

the earliest foundations of what would become computer science predate the invention of the modern digital computer. Machines for calculating fixed numerical …….

…….

</sec>

<sec>

<st>Areas of computer science </st>

</bdy>

Figure 3.4 : le document "computer science" de Wikipédia en XML (INEX 2009)

Exemple de requête

<topic id="2009001" ct_no="186">

<title>Nobel prize</title>

<castitle>//article[about(., Nobel prize)]</castitle>

<phrasetitle>"Nobel prize"</phrasetitle>

<description>information about Nobel prize</description>

<narrative>

I need to prepare a presentation about the Nobel prize. Therefore, I want to collect information about it as much as possible. Information, the history of the Nobel prize or the stories of the award-winners for example, is in demand.

</narrative>
</topic>

Figure 3.5: exemple de requête INEX 2009

Jugement de pertinence

la compagne INEX propose plusieurs requêtes (115 en INEX 2009). Et pour chaque requête un jugement de pertinence qui contient les documents pertinents. la compagne d'INEX est caractérisé par le fait qu'elle ne détermine pas seulement si un document est pertinent, mais elle détermine quelle partie (nœud ou balise) est pertinente dans un document XML.

```
2009001 Q0 1916241 0 10777
2009001 Q0 1806870 0 10510
2009001 Q0 13651185 0 7317
2009001 Q0 7252287 0 6904
2009001 Q0 19442420 0 15853
2009001 Q0 474958 0 5256
```

Figure 3.6 : partie de jugement de pertinence (Qrel) de INEX 2009

La figure 3.6 montre une partie de jugement de pertinence de la requête 1(2009001) dans la collection INEX de l'année 2009. on trouve dans chaque ligne le numéro de la requête (ici 2009001), le numéro du document pertinent (comme 1916241) et les partie pertinentes de ce document (le numéro du premier caractère jusqu'au dernier caractère)

3.3.2.2. Mesure d'évaluation

Il existent plusieurs facteurs pour mesurer la performance d'un système de recherche d'information tels que : le temps de réponse du système, le nombre de documents pertinent retournés, la méthode de présentation des résultats,....[BAZ 05].

Mais les deux mesures les plus importante pour l'évaluation d'un système de recherche est la **précision** et le **rappel**. Ces deux mesures sont largement utilisé et permettent de bien connaitre les performance du système de recherche.

La précision permet de déterminer le nombre de documents pertinents retourner par le système de recherche comme résultats par rapport aux nombres totales de documents retournés. La précision parfaite est égale à 1, qui veux dire que tous les documents retournés par le système sont pertinents.

Le rappel est le nombre de documents pertinents retourner par le système par rapport aux nombre totale de documents pertinents. Le rappel parfait est 1, et qui veut dire que tous les documents pertinents sont retournés par le système.

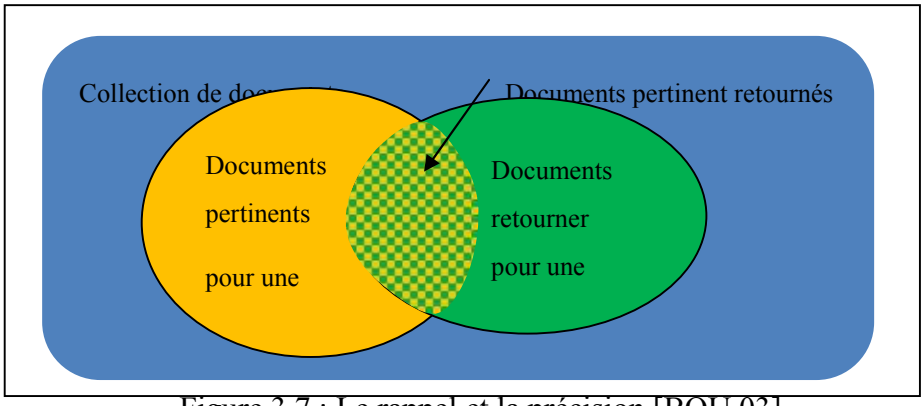


Figure 3.7 : Le rappel et la précision [BOU 03]

$$\text{Rappel} = \frac{Nombre\ de\ documents\ pertinents\ retournés}{Nombre\ total\ de\ documents\ pertinents}$$

$$\text{Précision} = \frac{Nombre\ de\ documents\ pertinents\ retournés}{Nombre\ total\ de\ documents\ retournés}$$

La courbe qui relie entre le rappel est la précision a l'allure générale suivante:

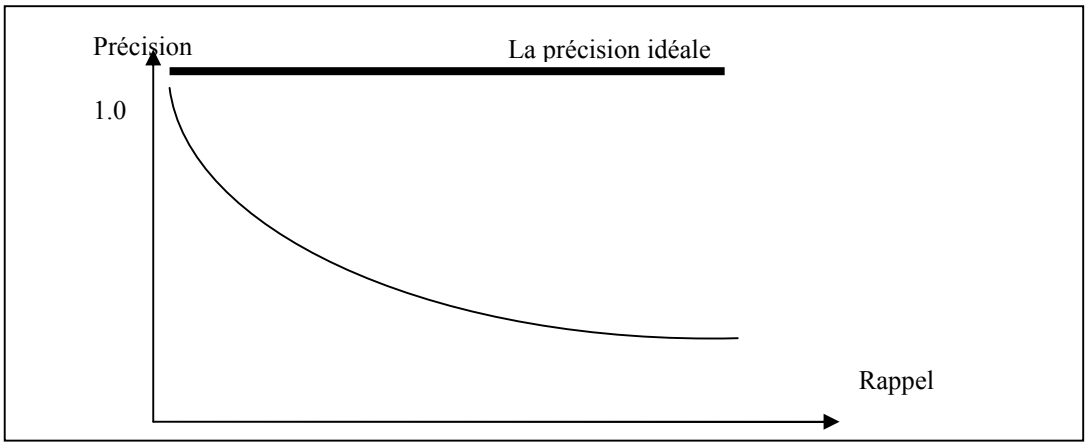


Figure 3.8: Courbe Précision/Rappel[BAZ 05]

Pour dessiner cette courbe on doit connaitre la valeur de la précision dans chaque point de rappel. les points de rappel sont le rapport entre les documents pertinent retourner et le nombre totale de documents pertinents. Le point 1.0 de rappel signifie que tous les document pertinent sont retournés.

Exemple

soit une requête "université de Biskra", ou le nombre totale de ses documents pertinents est 3 . Le système de recherche retourne 6 documents classés selon l'ordre suivants :

Documents Retournés	Pertinent / Non pertinent	Précision	Rappel
D1	P	1	0.33
D2	NP	0.5	0.33
D3	NP	0.33	0.33
D4	P	0.5	0.66
D5	P	0.6	1
D6	NP	0.5	1

Tableau : Exemple de rappel et précision

Le système sera idéale s'il ne retourne que les documents pertinents (D1, D4, D5) et il s'arrête. Ici le système a retourné comme premier document le documents D1 qui est pertinent, alors dans ce point la précision est 1 est le rappel est 0.33 (1 documents pertinent retourné sur trois existants). La précision commence a dégrader quand le système retourne des documents non pertinent.

Avec ces point de rappel et précision du tableau on peut pas dessiner une courbe qui aura l'allure de la figure 3.8. Pour ce faire on doit utiliser les technique d'interpolation linéaire .

Les systèmes d'évaluation issue des compagnes tels que TREC et INEX utilise des points de rappel (x=0.01, 0.05, 0.10, ..., 1) et calcule la précision dans ces points. ils utilisent aussi MAP (Median Average Precision) qui est la précision moyenne pour toutes les requêtes de la collection [HLA 07]

3.4. Recherche d'Information dans les documents semi-structurés

La recherche d'informations dans les documents semi-structurés (RIS) avait pris une place importante avec le développement du web, et notamment, après l'apparition et le développement des documents XML.

Le langage XML permet de modéliser n'importe quelle type d'information, grâce a son système de balise. ces balises permettent de décrire et structurer le contenu des documents.

La recherche d'information structuré a comme objectif de retourner, pour une requête, les parties et les unités les plus pertinentes d'un documents, et non pas tout le document. Pour ce faire le système prend en compte la structure logique du document modélisé par les balises. Les parties retournées comme résultats sont appelées *unité d'information*[HAL 07]. elle est caractérisée par deux notions:

*"On dit qu'une unité d'information est **exhaustive** à une requête si elle contient toutes les informations requises par la requête et qu'elle est **spécifique** si tout son contenu concerne la requête".[CHI 96]*

3.4.1. Modèle vectoriel pour les documents semi-structuré

Les modèles de la représentation classique de documents ont été étendues afin de s'adapter avec l'information structurelle. ils prennent en compte, maintenant, la structure et le contenu.

Le modèle vectoriel, qui prend en compte la structure, a été introduit en 2002 par [YAN 02]. où un vecteur d'un document D dans un espace de caractéristique à N dimensions a été présenté comme suit:

$$D= (w_1,w_2, ...,w_N)$$

où w_i dénote le poids des terme t_i calculé comme suit:

$$w_i= \sum_{j=1}^{M}(tf(t_i, e_j) * IDF(t_i)$$

où tf représente le nombre d'occurrence du terme t dans l'élément e du document

3.4.2. Pondération des termes dans les documents semi-structuré

Une extension de la formule TF-IDF, connu sous le nom TF-IEF [WOL 00] a été introduite pour inclure la notion de structure dans la fonction pondération.

Dans ce calcul, chaque nœud contenant le texte est traité séparément. TF est la fréquence d'un terme dans l'élément structurel correspondant du document et l'IEF est le fréquence inverse de l'élément :

$$tf(t_i, e_j) * IEF(t_i, e_j) = tf(t_i, e_j) * Log\frac{|E|}{|e_{tj}|}$$

où:

$tf(t_i, e_j)$ représente le nombre d'occurrence d'un terme dans un nœud.

$|E|$ est le nombre totale des nœuds dans la collection

$|e_{tj}|$ est le nombre de nœuds dans la collection où le terme t_i est apparu.

A partir de cette formule, plusieurs formules ont été proposé pour pondérer les termes dans les documents semi-structurés

3.5. Recherche sémantique d'information

Le problème majeur dans les systèmes de recherche classiques est que l'utilisation des mots clés simples comme requêtes conduit au problème de silence, où des documents

pertinents ne seront pas présentés comme résultats de la requête. les documents peuvent décrire le même contenu et le même sens en utilisant des termes déférents.

Les chercheurs ont pensé à utiliser la notion du concepts, au lieu d'un simple termes. un concept regroupe autour de lui plusieurs mots clés. Pour cela, les ressource sémantiques ont devenu un composant principale dans les moteur de recherche sémantique.

La ressource sémantique, intervient pratiquement, dans toutes les étapes de la recherche d'information: dans l'indexation des documents, dans l'analyse de la requête et dans l'étape d'appariement requête-documents

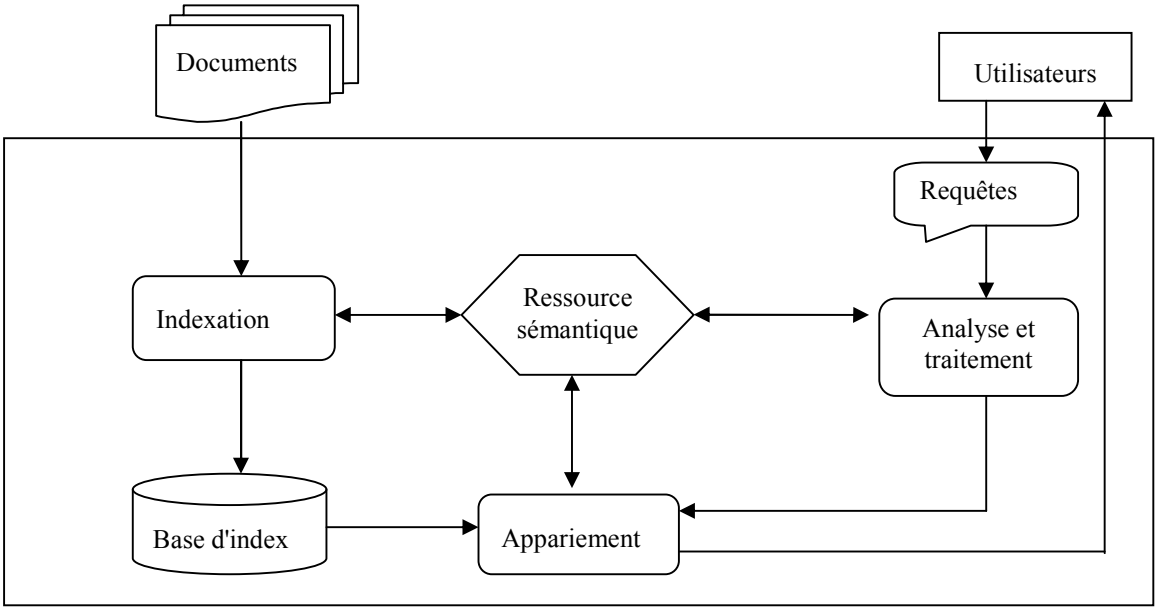


Figure 3.9: Ressource sémantique dans les systèmes de recherche d'information

3.5.1. Indexation sémantique

Dans la recherche classique, l'unité textuelle dans l'indexation est représenté par un mot clés. Cela pose deux problèmes : l'ambigüité et la disparité des mots [BOU 11]. L'ambigüité sémantique dû au fait qu'un terme peut avoir plusieurs sens selon le contexte ou il se trouve. Alors que la disparité des mots est dû au fait que les termes qui sont syntaxiquement différents ils ont le même sens. ces deux problème mènent à l'effet du silence où des documents pertinents ne sont pas retourné comme résultats pour une requête.

L'objectif de l'indexation sémantique est de désambigüiser le sens des termes. la tache consiste à représenter les documents et les requêtes par leurs sens. dans ce type d'indexation l'unité textuelle sera le concepts et non pas un simple terme.

Pour extraire le sens exact du terme, on doit utiliser les techniques de similarité sémantique pour trouver le meilleur concept qui représente le terme dans le contexte.

3.5.2. Traitement sémantique de la requête

Le traitement sémantique de la requête d'un utilisateur peut résoudre deux problèmes, le premier est les erreurs dans l'orthographe de la requête par l'utilisation d'un dictionnaire et le deuxième est l'expansion automatique de la requête.

Pour l'expansion de la requête. il existe plusieurs méthode regroupé en deux techniques locale et globale. les méthodes locale consiste à utiliser la notion de feedback, qui permet d'injecter de nouveaux terme dans la requête à partir des premiers documents retournés comme résultats.

La méthode globale permet de reformuler la requête en ajoutant d'autre termes proche sémantiquement aux termes initiaux de la requête (comme par exemple les synonymes). ces termes sont calculés et extraits à partir d'une ressource sémantique[CHR 09]

3.5.3. Appariement sémantique

L'appariement sémantique consiste à calculer la similarité sémantique entre les concepts de la requête et ceux des documents. Il permet de détecter les relations sémantiques entre les concepts , comme la relation de généralisation ou de spécialisation entre concepts. ces techniques permettent l'amélioration des résultats de recherche en retournant tous les documents qui ont une relation sémantique avec les concepts de la requête [BOU 11].

3.6. Outils pour la recherche d'information

3.6.1. Lucene

Lucene est une bibliothèque Java qui permet aux applications d'indexer et de rechercher un texte dans les documents. Lucene n'est pas une application, c.-à-d. elle ne peut pas fonctionner toute seule. Mais c'est un ensemble de classes et de méthodes qui sont utilisé dans des applications java.[MIC 10]

Lucene est développé par Doug Cutting, c'est un projet open source, disponible pour le téléchargement libre. Lucene a rejoint la famille de Jakarta de la Fondation Apache[20] depuis 2001. La bibliothèque de Lucene peut être intégré dans plusieurs environnements de programmation et dans différent langage tels que : C/C++, C#, Ruby, Perl, Python, PHP,…

Plusieurs projets ont utilisés Lucene comme un outils puissant de recherche d'information. parmi ces projets : LinkedIn, ifinder, blogdigger, ...[21]

[20] www.**apache**.org

[21] https://wiki.apache.org/lucene-java/PoweredBy

Lucene offre deux services principaux: indexation de texte et recherche de texte.

3.6.1.1. Les classes de Lucene

Lucene est constitué d'un ensemble de classes qui sont utilisés pour construire une application de recherche. Les principales classes sont:

- **IndexWriter** est utilisé pour créer et maintenir des index
- **IndexSearcher** est utilisée pour rechercher dans un index
- La classe **Analyzer** est une classe abstraite qui est utilisé pour prendre un document et le transformer en termes qui peuvent être indexés.
- La classe **Document** représente un document dans Lucene. Les documents sont l'unité de l'indexation et de la recherche. Un document est un ensemble de champs (Field)
- La classe **Field** est un champ qui représente une section d'un document. chaque champ a un nom et contient un texte comme données
- La classe **QueryParser** est utilisée pour construire un parseur qui peut analyser la requête pour chercher ensuite dans un index

3.6.1.2. Indexation dans Lucene

Le cœur de tous les moteurs de recherche est le concept de l'indexation; Indexation peut être définie comme le traitement des données d'une manière très efficace en vue de faciliter une recherche rapide

L'indexation est le cœur de Lucene, elle se fait par des analyseurs. Les textes inutilisables tels que les mots vides, les suffixes de mots ou préfixes sont rejetées par l'analyseur. A la fin de cette étape, un index est créée. .[MIC 10]

3.6.1.2.1. Structure de l'index dans Lucene

L'index de Lucene est stocké dans un répertoire dans le système de fichiers sur un disque dur. Les éléments de base d'un index Lucene sont des segments, des documents, des champs, et les termes. un index Lucene est constitué d'un ou plusieurs segments. Chaque segment contient un ou plusieurs documents. Chaque document a un ou plusieurs champs, et chaque champ contient un ou plusieurs termes. Chaque terme est une paire de chaînes représentant un nom de champs et une valeur. [MIC 10]

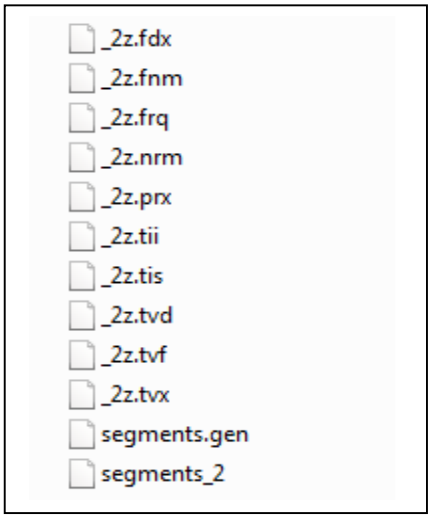


Figure 3.10 : les fichiers d'un index dans répertoire

Un index de Lucene est un index inversé. un index inversé est un ensemble de termes qui pointent vers les documents correspondant. pour créer l'index inversé, Lucene analyse le contenu des documents et extrait les termes importants et les stocke comme un couple constitué d'un nom de champs et une valeur. Les champs sont utilisés pour calculer les poids et le classement des résultats de recherche. .[MIC 10]

La figure suivante représente un index de Lucene qui se compose de plusieurs champs (Field) comme : sectitle, contents, filenames, ... et chaque champs contient des termes

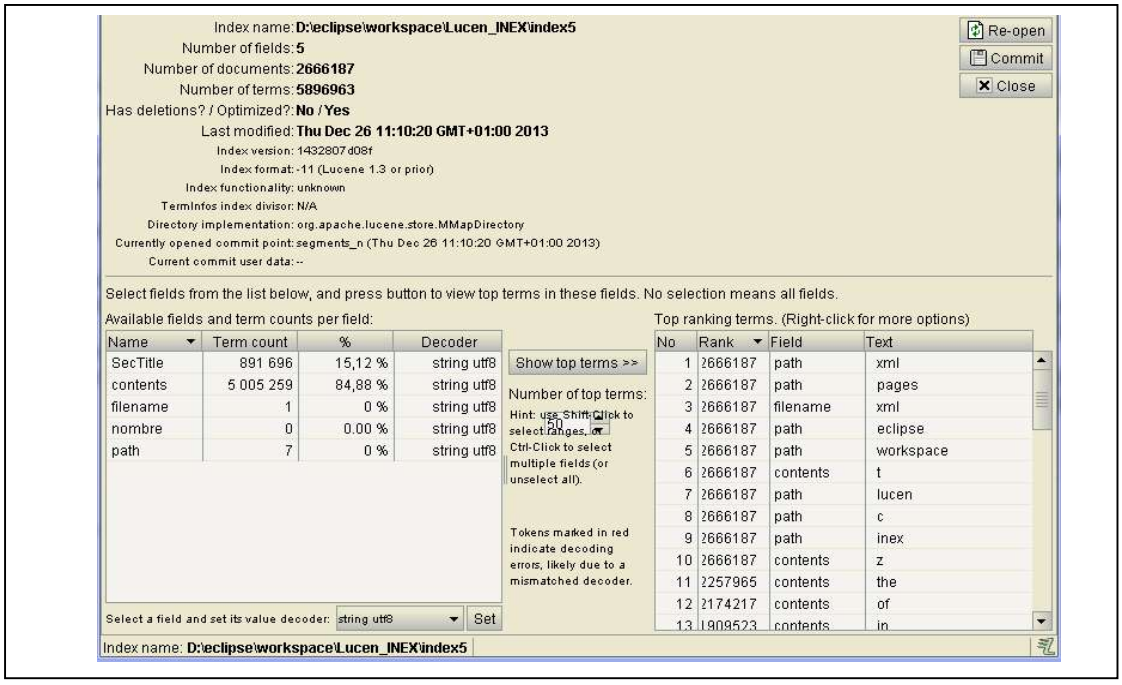


figure 3.11: Exemple d'un index dans Lucene

3.6.1.3. Recherche dans Lucene

La recherche consiste a retourner les documents pertinents pour une requête d'un utilisateur. La requête est un ensemble de termes. Lucene analyse la requête et recherche dans l'index les document qui lui correspondent. Pour se faire Lucene applique une fonction de calcule de similarité suivante[22] :

$$score(q,f) = coord(q,f) \cdot queryNorm(q) \cdot \sum(tf(t \text{ in } f) \; idf(t)^2 \cdot norm(t,f))$$

où:

tf (t en f) est la fréquence du terme t dans le champs f du document;

tf (t inf) = √ fréq

idf (t) représente l'inverse de la fréquence de terme t dans l'ensemble du document;

idf (t) = log (numDocs / (docFreq 1)) + 1

où numDocs représente le nombre total de documents dans le corpus, et docFreq, le nombre de documents qui contiennent le terme t.

coord (q, f) est un facteur de score basé sur le nombre de termes de la requête contenue dans un champ spécifié; Un champ contenant plusieurs termes de la requête aura un score plus élevé

coord (q, f) = tq / TQ

tq: Numéros termes de l'application qui sont dans le domaine

TQ: nombre total de termes de la requête

queryNorm (q) est un facteur de normalisation utilisé pour faire des demandes similaires;

queryNorm (q) = 1 / √ (Σidf (t ²))

norme (t, f) est utilisée pour normaliser la taille des champs (pour faire des domaines comparables) un champs plus courtes auront un score plus élevé

norme (t, f) = 1 / √ (nombre de termes dans le domaine)

[22] https://lucene.apache.org

3.6.2. Terrier ir

Terrier[23] (Terabyte Retriever), est un projet qui a été lancé à l'Université de Glasgow en 2000, dans le but de fournir une plate-forme flexible pour le développement rapide d'applications à grande échelle de Recherche d'information

C'est un projet open source, qui a été mis à la disposition du grand public depuis Novembre 2004. La version open source de Terrier est écrit en Java, permettant au système de fonctionner sur des systèmes d'exploitation différentes, et sur plusieurs plates-formes[ROD 11]

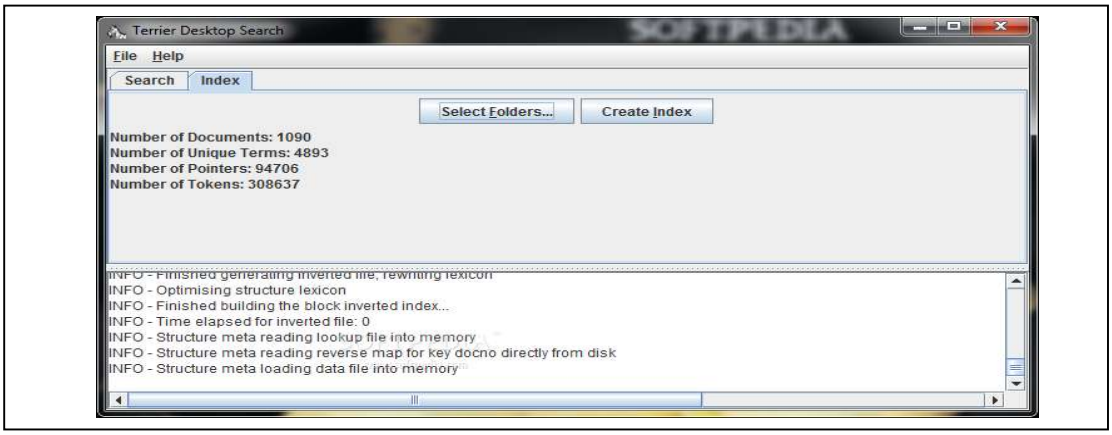


Figure 3.12: Interface de l'application bureautique de Terrier

3.7. Conclusion

Nous avons présenté dans ce chapitre un domaine qui modélise et traite les documents numériques, qui est la recherche d'information. Le cœur de ce domaine est l'indexation des documents, qui permet d'analyser le contenu des document et le présenter d'une manière efficace affin de faciliter la recherche.

Dans ce chapitre nous avons décrit tous les concepts liés à ce domaine, comme nous avons présenté toutes les techniques et méthodes utilisé afin d'améliorer la pertinence des résultats de la recherche.

Dans le chapitre suivant nous allons détaillé notre contribution qui permet de modéliser sémantiquement la structure des documents numérique. ce qui permet de faire progresser la performance de la recherche de documents.

[23] http://terrier.org/

Chapitre 4

Contribution : Une approche structuro-sémantique pour la recherche de documents

4.1. Introduction

Les systèmes de recherche et d'accès aux documents consistent à identifier les documents les plus pertinents par rapport à une requête donnée [SAL 83]. Il y a souvent de très longs documents qui traitent de nombreux sujets, qui sont répartis dans les sections logiques de ces documents, où chaque section a un titre et des paragraphes décrivant le contenu. Les titres de section ont une importance majeure pour modéliser la structure logique des documents, ainsi ils décrivent la sémantique contenu dans les paragraphes. Ils sont très utiles dans la recherche d'information.

Dans ce chapitre, nous allons présenter nos contributions et l'architecture générale de notre approche qui consiste à extraire la structure logique des documents pour modéliser la hiérarchie de ces documents, et ensuite modéliser et exploiter sémantiquement cette structure afin d'améliorer les résultats de la recherche documentaire.

4.2. Motivation

Lorsque l'auteur a choisi de mettre des mots dans les titres et d'appliquer sur ces termes un format de police spéciale, comme une taille différente, les lecteurs comprendront que ces termes sont particulièrement importants dans le texte. Ici, l'auteur a annoté les termes qui représentent bien le sujet du document.

Les documents numériques en différents formats tels que PDF, Doc, HTML, PS,... sont très nombreux dans les bases de documents. En effet, ces formats sont devenus un moyen privilégié de publication, dans le monde universitaire et dans l'industrie, notamment le format PDF qui est facile à échanger, à visualiser et à imprimer [LIA 11]. De plus, il est exploitable dans pratiquement tous les systèmes [ROS 11].

Cependant, la plupart des formats des documents sont protégés, la structure logique (section, paragraphe, chapitre, titre de sections…) du contenu des documents n'est accessible le plus souvent que par les lecteurs humains, et elle n’est pas exploitable par les applications, ce qui rend l’extraction des parties pertinentes du document très difficile [ROS 11].

Lorsqu’un auteur crée une hiérarchie de titre ; où des titres d’un niveau n généralisent les titres d’un niveau n+1, ceci revient à réaliser une segmentation du document par l’auteur. Cette segmentation peut être très utile dans le processus d’analyse et de traitement de ces documents et dans le domaine de recherche d’informations.

Par exemple un document qui a des titres de sections contenant les mots « recherche et sémantique » sera plus pertinent qu'un document qui contient ces mots dans les paragraphes inclus. Dans le premier cas, on comprend que toute une section parle de « recherche et sémantique », mais dans le deuxième, on devine qu'il s'agit seulement d'une partie de cette section.

Notre objectif est d'exploiter les titres des objets logiques (chapitres, sections, paragraphes) pour modéliser leur signification et leur importance dans le processus de recherche de documents. Et puis structurer ces documents et retourner à l'utilisateur uniquement les parties les plus pertinentes des documents.

4.3. Travaux existants

Dans ce qui suit nous allons présenter brièvement les travaux qui ont étudié l'importance des titres et la faisabilité de leur extraction**.** Approches pour la structuration des documents

Il existe des travaux comme [ROS 11] [HER 09] [LIA 09] qui ont essayé d'extraire la structure logique des documents en format PDF, pour l'utiliser dans la classification et la structuration. Les travaux de [HER 09] et [LIA 09] présentent des méthodes pour exploiter la table des matières dans la structuration et la classification. Ils ont proposé une approche pour détecter automatiquement la table des matières, puis extraire les titres avec leur niveau respectif dans la hiérarchie des objets logiques, et finalement, structurer les documents.

D'autres travaux comme [JOE 13] et [EDV 09] proposent des techniques pour localiser et extraire des métadonnées à partir des documents PDF.

[JEA 05] propose un "Framework" pour transformer les documents dans un format initial tel que PDF, PS, HTML..., en un format XML. Il présente une méthode pour l'analyse et l'extraction des éléments structurels du document. Cette méthode utilise si cela est possible la table des matières pour la structuration des documents. Une autre étape, appelée annotation sémantique, se réfère plutôt à la signification des éléments qu'a leur apparence sur une page, comme les noms des personnes, pays….

Pour la recherche focalisée ou ciblée, [MAT 10] a étudié l'impact des balises qui représentent la structure logique des documents codés en format XML dans la recherche ciblée des informations, afin de retourner la partie (élément) la plus pertinente du document XML. Il ajoute à chaque balise (balise titre, résumé, section…) un poids afin de bien déterminer les termes pertinents. [MAT 10] calcule aussi le poids de toutes les balises et pas seulement les balises qui contiennent les titres et les sous-titres.

4.3.1. Travaux étudiants l'importance des titres

Parmi les travaux qui ont étudié l'importance des titres de section et de sous section dans les documents on trouve : [JAC 06] et [HOD 04].

Les auteurs ont montré que les titres ont une importance sur deux plans : premièrement, en tant qu'objets d'organisation logique du texte qui sert à segmenter, hiérarchiser, et structurer le contenu d'un document ; *deuxièmement,* ils présentent le contenu sémantique des documents, pas d'une manière explicite, mais comme contenu structuré, qui permet aux lecteurs de construire un « modèle mental » pour comprendre la signification du texte au fur et à mesure qu'il lit le document. Ces travaux montrent donc l'utilité des titres dans la classification et l'extraction automatique des segments pertinents

4.3.2. Approches basées sur l'exploitation des titres d'un document

Dans la recherche d'informations, peu de travaux ont exploité les titres qui se trouvent dans les documents, la plupart n'exploitent que le titre principal des documents et ignorent les titres des objets logiques composants.

Par contre dans le web, ils existent plusieurs travaux qui exploitent les titres et lien hypertexte (anchor text), qui se trouvent dans les pages Html, pour améliorer la recherche d'informations. A titre d'exemple, citons le travail de [XUE 07] dont l'objectif est d'extraire les titres qui se trouvent dans le corps des pages web et de les utiliser dans l'indexation de ces pages, en proposant une nouvelle méthode de pondération, inspirée de la méthode Okapi BM25 [STE 09]. Il a montré que l'utilisation de ces titres améliore la recherche d'informations.

Le travaille de [HEN 08], propose une méthode pour la recherche dans des livres scannés, en exploitant leur structure. Il a créé un index à plusieurs champs pour chaque objet du livre (table des matières, index …). En utilisant le modèle de recherche BM25F [HER 09], il a montré que les résultats de récupération de livre s'améliorent avec cette méthode.

Le travail de [WAL 08], qui consiste à indexer les déférentes parties des livres scannés, et à réaliser une étude comparative, a montré que l'indexation des titres de livres est plus efficace que l'indexation du contenu. Il a ainsi montré que les titres ont plus de valeur que les autres parties de livres.

4.3.3. Approche qui exploite une ressource sémantique

Il existe plusieurs travaux qui ont utilisé une ressource sémantique dans le processus de recherche d'informations. Ces travaux se diffèrent par la manière d'exploitation de la ressource sémantique, et dans quelle étape intervient cette ressource, soit dans la représentation et l'indexation des documents, ou bien elle intervient dans l'étape et l'algorithme de recherche, ou dans la modélisation et l'expansion de la requête. Chaque travail se caractérise aussi par le type de la ressource qu'il utilise : Une ontologie, une taxonomie, un thesaurus, ...etc.

Parmi ces travaux [RAM 10], on peut citer les travaux de [BAZ 05], qui a proposé des méthodes de représentation des concepts des documents comme un réseau sémantique. Il utilise WordNet comme ressource sémantique afin de calculer la similarité sémantique entre les termes et détecter les meilleurs concepts pour les termes des documents. Les travaux de Khan [KHA 04], utilise une ontologie de domaine du sport pour indexer les documents. L'ontologie permet de désambiguïser les termes et représenter les documents sous forme d'un vecteur, par les concepts adéquats. Le travail de Radhouani [RAD 08], utilise la logique de description pour représenter le document et la requête. La ressource sémantique utilisée dans son travail est UMLS, une ontologie de domaine médical. Le travail de Maissonace [MAI 09] utilise aussi UMLS comme ressource sémantique, et il utilise aussi l'analyse morpho syntaxique pour détecter les noms et les adjectifs dans les documents pour les représenter ensuite comme vecteur de concepts.

4.4. Contexte et problématique

Les systèmes de recherche d'informations documentaires ajoutent pour chaque document des informations signalétiques, comme le titre, l'auteur, la date, le résumé, des mot-clésss clés …etc. Ces informations sont appelées les métadonnées. L'exploitation de ces métadonnées permet d'améliorer le résultat en minimisant le bruit (document non pertinent sélectionné) et le silence (document pertinent non sélectionné).

Cependant, ces métadonnées doivent être ajoutées manuellement dans le système documentaire afin qu'elle puissent êtres exploités. Dans les documents textuels en format "plat" comme les thèses en format PDF (figure 1), leurs métadonnées seront considérées comme de simples termes, et seront indexé comme le contenu textuel et non pas comme des termes importants.

Notre objectif est de proposer une approche qui permet de traiter un ensemble de documents homogène d'une manière automatique, afin de détecter les métadonnées et les parties importante pour les exploiter ensuite dans le processus de recherche

d'information . Les bases de documents qui utilisent uniquement des métadonnées, ont deux inconvénients majeurs. Le premier est qu'ils ne font la recherche que sur les métadonnées. Le deuxième, est qu'ils retournent le document entier, et non les parties qui répondent bien à la requête, ce qui oblige l'utilisateur à lire tout le document. [ABA 07].

Numéro d'ordre : 2010-ISAL-0073 Année 2010

Institut National des Sciences Appliquées de Lyon

INSA LYON

THÈSE

en vue de l'obtention du

GRADE DE DOCTEUR

SPECIALITE : INFORMATIQUE

délivré par

L'Institut Nationale des Sciences Appliquées de Lyon

présentée par

Rami HARRATHI

Recherche d'information conceptuelle dans les documents semi-structurés

Soutenue à Lyon le 29 Septembre 2010 devant la commission d'examen :

Jury

Gilles FALQUET	Professeur à l'Université de Genève	Rapporteur
Genevève LALLICH-BOIDIN	Professeur à l'Université Lyon 1	Rapporteur
Jérôme GENSEL	Professeur à l'Université Pierre Mendès France	Examinateur
Jean-Marie PINON	Professeur à l'INSA de Lyon	Examinateur
Sylvie CALABRETTO	Maître de Conférences HDR à l'INSA de Lyon	Directrice de thèse

Figure 4.1: Métadonnées dans la page de garde d'une thèse scientifique.

Par exemple si une personne cherche des documents qui traitent de « WordNet », si ce mot-là n'apparaît pas explicitement dans le titre principal ou dans les métadonnées d'un document, il ne sera pas retenu comme un document pertinent, même si une partie en traite.

L'exploitation du titre principal d'un document dans le processus de recherche documentaire peut améliorer les résultats retrouvés, mais reste insuffisante parce que le titre principal ne donne qu'une vision globale sur le thème général abordé par le document, et pas tous les sous-thèmes abordés dans les différentes parties du document.

Dans les documents numériques, assez long, comme les thèses, la structuration logique, les décompose en objets logiques (chapitres, sections, sous sections, titres de sections, paragraphes, etc.) où les titres de ces objets logiques décrient le thème ou le domaine abordé dans l'objet correspondant.

Quand l'auteur d'un document numérique, choisit de mettre des termes dans des titres de sections, et d'appliquer sur ces termes un format de police spéciale, comme par exemple, une taille différente, les lecteurs comprennent que ces termes ont une importance particulière dans le texte (Figure 2). Par exemple, l'auteur indique l'importance d'un terme par un type de police particulier ce qui revient à annoter avec ce terme les parties qui représentent bien le thème du document.

Dans l'exemple de la figure 2, le titre du document est "computer science", caractérisé par une taille de police très grande par rapport aux restes des titres et de texte. Ce document contient plusieurs section et sous section, où chacune a un titre avec une taille de police précise et une numérotation qui définit le niveau de la section. Cette structure permet de comprendre la hiérarchie du document et l'importance de chaque partie de ce document par rapport aux autre partie.

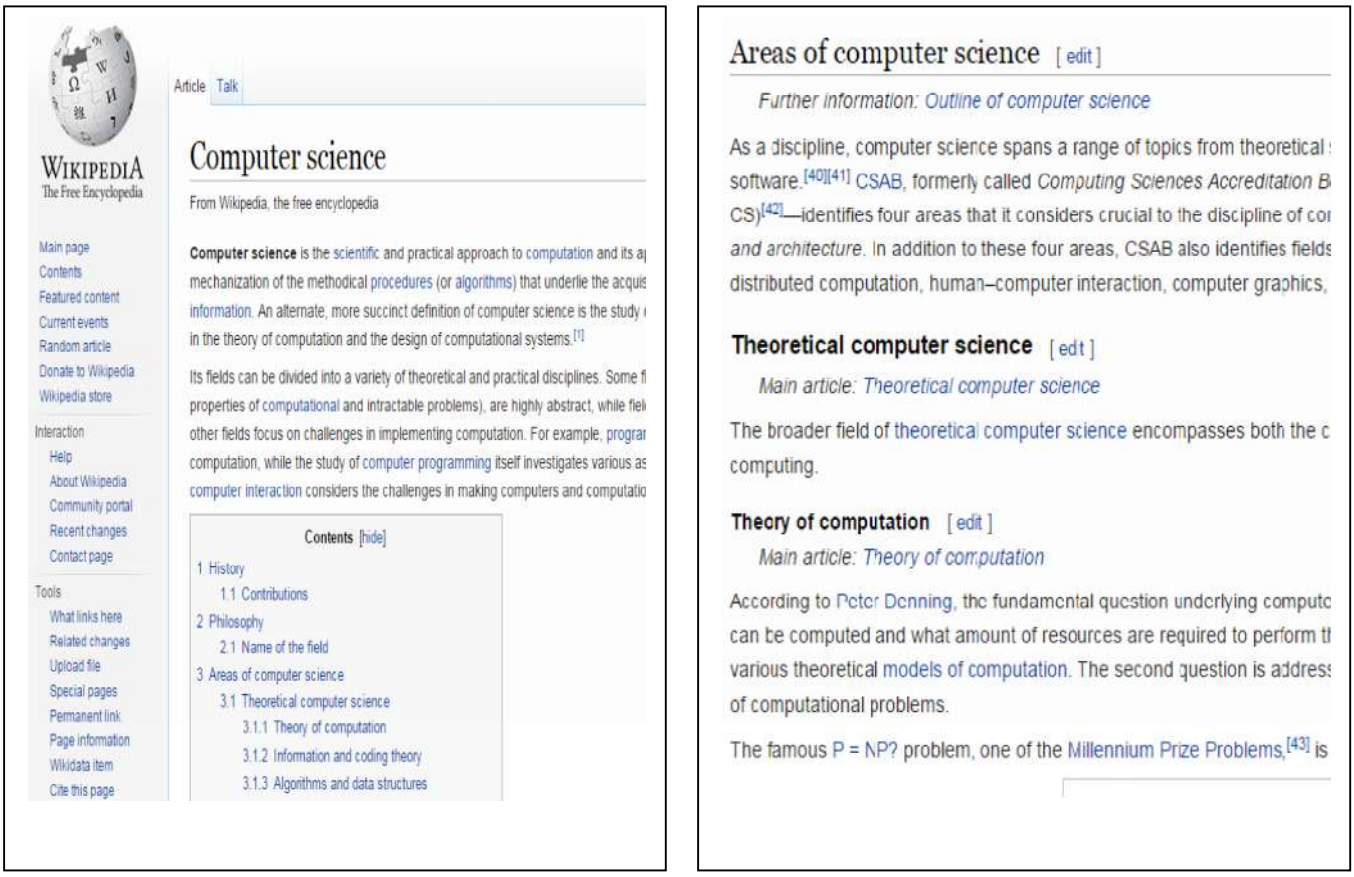


Figure 4.2: Un exemple d'un document Wikipédia structuré

Par exemple, la figure 2 ci-dessus, représente un document Wikipédia en format HTML (page web). La section 3 du document qui a comme titre, "Areas of computer science", cette section a une numérotation (ici 3), et elle content aussi des sous-sections sections, et une taille de police plus grande par rapport aux sous-titres el les paragraphes du texte.

Cette structuration réalisée par l'auteur du document permet aux lecteurs humains de connaître les termes et les parties importantes. Mais les machines voient et traitent le document comme un seul bloc de texte.

Notre objectif est de proposer une approche qui permet aux machines de comprendre la structuration des documents et l'exploiter dans l'indexation et la recherche des documents en donnant aux termes importants un poids élevé par rapport aux autres termes.

Un autre point important et intéressant dans l'analyse de ce type de documents est que les termes de titres que nous considérons importants et ont une signification majeure, et qui donnent une vision sur le fond et le contenu de leurs paragraphes, ci termes là , peuvent avoir des relations sémantique avec d'autres termes, qui se trouvent dans les paragraphes. se qui rend ces termes importants aussi. les synonymes et les hyperonymes sont un exemple de ces relations.

Dans l'exemple ci-dessus (figure 2), on trouve dans le texte des termes comme : processus, algorithmes, intelligence Artificiel,.... etc. Ces termes ont des relations sémantiques avec "computer science". ces relation peuvent êtres identifiés et extraire si on projette ce document sur une ressource sémantique comme WordNet.

Un autre exemple, les termes des métadonnées d'un document (figure 1), ces termes aussi sont importants et permettent d'améliorer la recherche et l'accès à ces documents.

Prenant l'exemple du terme " directeur de thèse", ce terme peut êtres écrit dans d'autres thèses comme encadreur, Co-encadreur, ou codirecteur. Tous ces termes signifie le même concept. Cependant, si quelqu'un qui pose une requête : "Thèse encadrée par Sylvie Calabretto", le moteur de recherche ne va pas retourner ce document dans les premiers résultats, parce qu'il ne fait pas de liaison sémantique entre le terme encadreur et directeur de thèse.

Un autre objectif de notre thèse est de proposer une méthode qui exploite une ressource sémantique pour trouver tous les termes importants dans les documents et qui ont des relations sémantiques avec les termes des titres et les métadonnées du document.

4.5. Approche proposée

Partant du fait que les documents dits "plats" ont implicitement une structure, nous voulons exploiter cette structure pour améliorer la recherche et l'accès a ces documents. Ces documents contiennent des sections et des sous-sections avec différents titres et sous-titres correspondants. La pluparts des systèmes traditionnels de recherche n'utilisent pas cette structuration dans leur processus.

L'exploitation de la structure des document devient très importante dans la recherche sémantique d'information, parce que ces documents ont un contenu diversifié avec des titres qui explicitent le sens de chaque partie du document

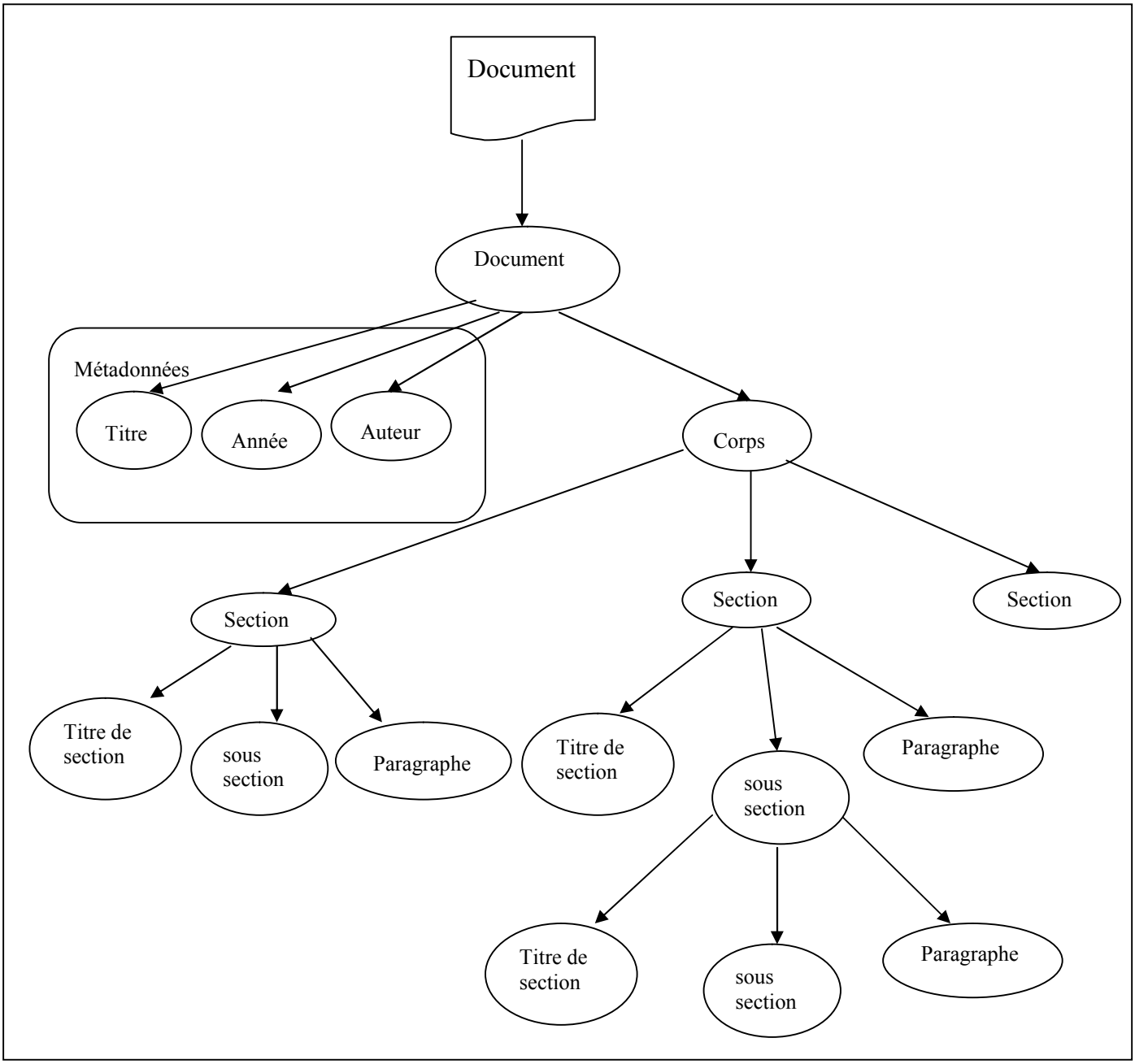


Figure 4.3 : Modélisation d'un document sous forme d'un arbre.

Dans notre système proposé, nous allons exploiter les informations structurelles, notamment les hiérarchies de section. Afin de, déterminer premièrement les sections les plus importantes pour une requête utilisateur.

Deuxièmement, certaines informations structurelles telles que les titres et les sous-titres englobant les termes les plus important dans le document et pouvant être exploités sémantiquement pour détecter d'autres termes importants dans les documents, ou bien ajouter d'autres termes à ces documents à partir d'une ressource sémantique.

Notre approche se résume par les points suivants :

1. Identifier et détecter les Métadonnées dans les documents homogènes comme les thèses et les articles scientifiques en formats plat (PDF, DOC, ...)
2. Identifier les sections et les sous-sections des documents, ainsi que leurs titres et sous-titres correspondants.
3. Restructurer la hiérarchie des sections en modélisant les documents sous format XML pour détecter ensuite les parties importantes des documents.
4. Extraire les termes les plus importants dans le documents à partir de la structure logique et physique des documents. Puis exploiter ces termes pour améliorer la pertinences des documents dans les résultats de recherche
5. Détecter d'autres termes importants dans le contenu textuel de documents en projetant les termes extraits de la structure sur une ressource sémantique (WordNet).
6. Elargir les termes importants dans les documents par la suggestion d'autres termes proche sémantiquement aux termes importants existent déjà.

4.6. Architecture du système

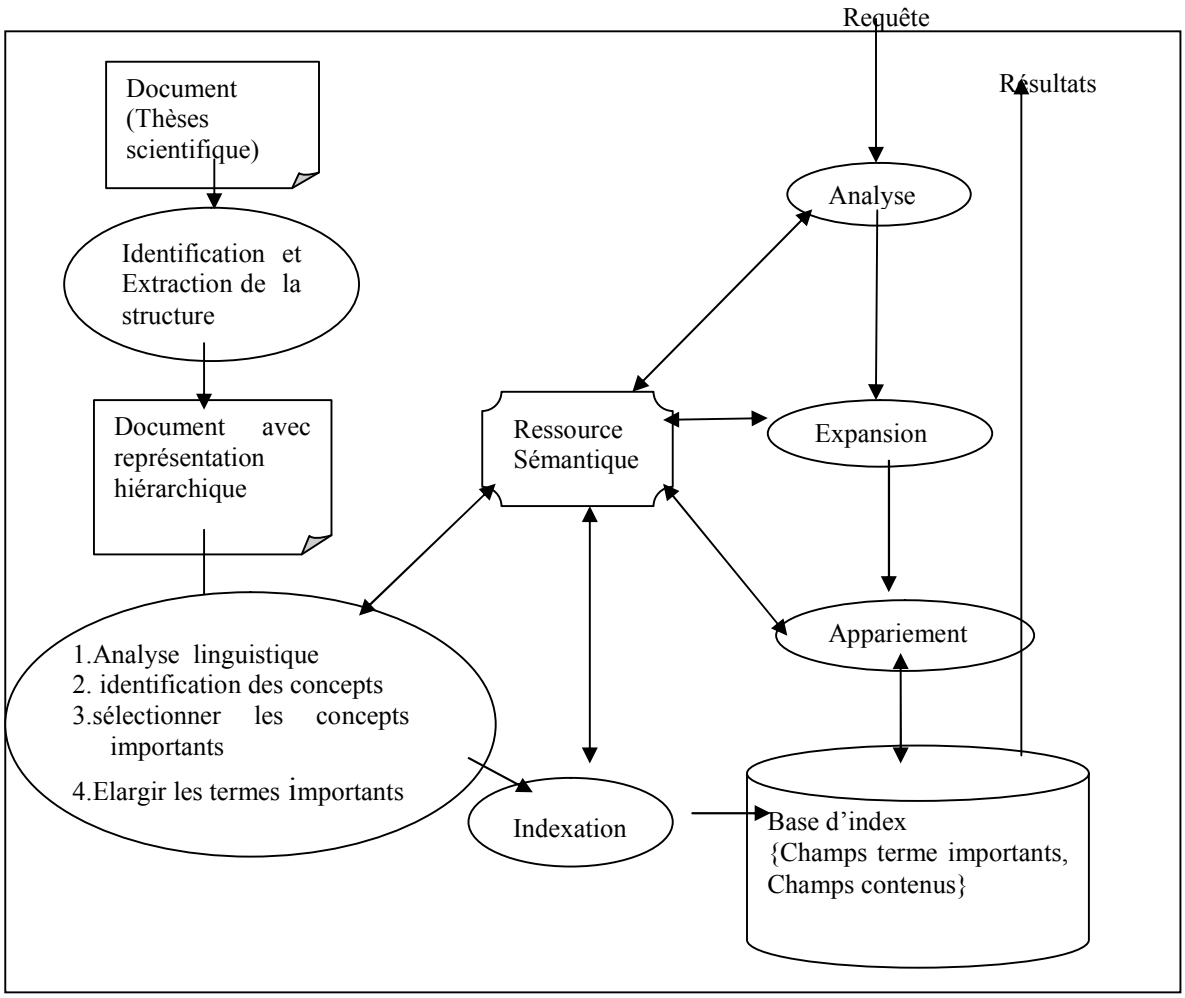


Figure 4.4 : Approche générale du système [ABD 15]

4.6.1. Identification et Extraction de la structure

L'objectif de la transformation structurelle est l'identification de la hiérarchie du document. Les thèses en format PDF, en général, ont une structure qui peut être considérée comme une hiérarchie, où chaque document peut avoir des sections, chaque section peut avoir un titre et des paragraphes et des sous-sections ainsi de suite.

4.6.1.1. Extraction des titres [ABD 14b]

Nous nous limitons aux thèses qui ont une table des matières. Celle-ci se trouve généralement dans les premières pages. Elle est caractérisée par un début (un nom tel que « table des matières »), et une ligne qui détermine la fin. Entre ce début et cette fin, il y a une succession de ligne ou chaque ligne (parfois deux) représente le titre d'un objet logique.

Chaque ligne de titre est composée de trois champs (ou triplet) , titre, numéro de page > [JAY 12], ou : division représente la position hiérarchique de l'objet correspondant dans la structure logique, titre : représente le titre l'objet logique (chapitre, section …). Par exemple <chapitre 1, Introduction générale, 10> ou bien <2.2.1 Recherche d'information …….15>

Pour extraire les titres à partir de ces documents PDF, dans un premier temps, on extrait la totalité du texte. En utilisant la plate-forme **itextepdf**[24].

Ensuite, on doit détecter l'emplacement de la table des matières, en identifiant le début dans le document.

Le début de la table des matières est la ligne qui contient la phrase « Table des matières » ou toutes ses variations telles que « Sommaire » ou bien d'autres mots utilisés pour annoncer une table de matière.

Nous extrayons les lignes suivantes, jusqu'à la ligne qui contient le titre qui représente la « Conclusion Générale », avec toutes ses variations (conclusion générale, conclusion et perspectives, …etc.) parfois, la conclusion générale a comme titre «conclusion », ce qui confond avec les « conclusions » des chapitres, alors si c'est le cas on s'arrête à la ligne « références bibliographiques ».

Entrées de la table	Fin de la table
Table des matières	CONCLUSION GENERALE
TABLE DES MATIÈRES	CONCLUSIONS ET PERSPECTIVES
Sommaire	REFERENCES BIBLIOGRAPHIQUES
Avant propos	BIBLIOGRAPHIE
Introduction générale	PERSPECTIVES
Chapitre 1	…….

Tableau 4.1 : différent type du début et fin de la table des matières [ABD 14b]

L'algorithme proposé pour l'extraction est donné ci-dessous :

Algorithme :

Entrées : Fichier PDF

Sorties : F (fichier de titres)

1. Détecter le début de la table des matières

[24] http://itextepdf.com/

2. Si début trouvé
3. Répéter
4. Extraire ligne
5. Si du texte existe dans ligne
6. extraire titre à partir de ligne
7. sauvegarder titre dans F
8. Jusqu'à ligne qui contient la fin de la table
9. Retourner F

4.6.1.2. Difficulté dans l'extraction de texte

Pour extraire un texte à partir d'un document PDF, on doit toujours le transformer en un format texte, il existe plusieurs outils qui font cette opération (comme itextpdf, pdfbox…), et là on perd des informations sur le format et le style de texte, et des fois il y aura des changements mêmes sur les lignes et les mots comme par exemple le titre suivant :

2.3.4. Les différentes approches de l'indexation multilingue 42

Devient : *2.3.4. Les différentes approches de l'*
indexation
multilingue 42

Ce type d'erreurs génère des problèmes dans l'extraction de titres et de mots

D'autres types d'erreurs sont représentées dans le tableau ci-dessous [ABD 14b]:

Avant conversion	**Après conversion**
sémantique	se mantique
définis	dé?nies
Mécanismes et Implications,	MécanismesetImplications
Introduction	I n t r o d u c t i o n

Tableau 4.2 : Erreurs de conversion d'un document PDF au format texte

4.6.1.3. Représentation hiérarchique du document

Après l'extraction des titres, on passe à l'extraction de texte qui se trouve dans les sections et les sous-sections sections du document. Les titres et les sous-titres titres permettent de définir et détecter les blocks de texte qui leur correspondent. Le résultat final de l'opération sera un arbre qui représente la hiérarchie des sections, ou les titres et les sous-titres titres seront les nœuds intermédiaires et le texte de chaque section

sera une feuille. L'arbre du document sera stocké comme un fichier bien structuré sous format XML

4.6.2. Analyse linguistique

C'est une étape principale de prétraitement des documents avant la phase de l'indexation. Cette étape est composé de plusieurs opérations telles que : Analyse lexicale, l'élimination des mots vides, lemmatisation, racinisation (stemming) ...etc.

4.6.2.1. Analyse lexicale

Cette étape permet de découper le document en unités lexicales. Chaque unité lexicale est une séquence de caractères entourée par des séparateurs d'unités. Ces unités peuvent êtres des mots simples ou des mots composés. Cette opération se compose de 3 étapes :

1) Détecter les lettre majuscule et les ponctuations... etc.

2) Transformer les lettres majuscules en minuscules et éliminer les ponctuations.

3) Découper le document en unités lexicales.

4.6.2.2. L'élimination des mots vides

Les mots vides (« stop words » en anglais) sont des mots qui permettent de lier entre les mots d'une phrase pour la structurer. Parmi ces mots : les articles, les conjonctions de coordination, les verbes auxiliaires, etc. Ces mots ne portent pas de sens, alors ils seront donc éliminés.

Exemple

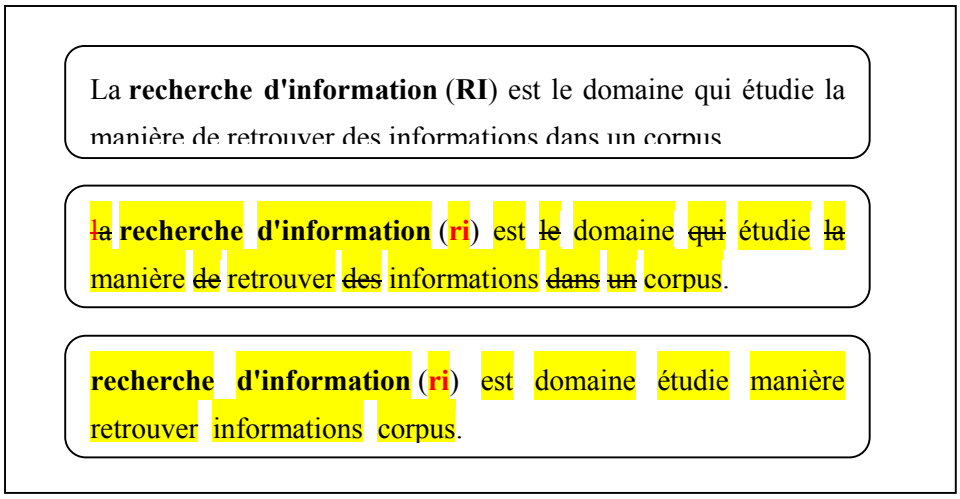


Figure 4.5: Analyse lexicale et suppression des mots vides

4.6.2.3. Lemmatisation

La lemmatisation consiste à remplacer un mot par son lemme. les mots *informatique ,informer ,informés ,informez ,informé* et *information* seront remplacés par leurs lemmes : *informe*

4.6.2.4. Racine d'un mot

La racinisation (« stemming » en anglais) consiste à rechercher la forme restante tronquée d'un mot après la suppression de son suffixe et son préfixe. Les mots : ***étude, étudiant*** et ***étudier***, sont construits à partir de la racine ***étud***. Le résultat de le racinisation peut être une forme qui ne corresponde pas à un mot réel dans la langue. Elle permet d'augmenter le rappel, mais peut diminuer la précision.

4.6.2.5. Etiquetage morpho-syntaxique [ABD 15]

Dans cette étape, nous utilisons l'outil d'étiquetage morpho-syntaxique du discours (POS Tagger) [TOU 00] qui permet d'identifier chaque terme comme : nom, verbe, adjectif, etc. Par exemple, la phrase suivante (en anglais) : "Model fitting data analysis" sera étiquetée comme suit: *Model/NNP fitting/JJ data/NNS analysis/NN,* où NN est nom, NNP est un nom propre, NNS est nom pluriel, JJ est adjectif.

Après le marquage, nous filtrons les mots. WordNet contient quatre types: verbe, adverbe, adjectif, nom, mais nous sommes seulement intéressés par les noms. Les relations sémantiques entre les autres types ne sont pas très bien conçues [MAL 11].

4.6.3. Identification des concepts

Dans cette étape, nous détectons le meilleur concept pour chaque terme. Chaque terme peut avoir plusieurs sens, nous allons essayer de déterminer le meilleur sens (concept) dans le contexte (phrase ou paragraphe) ou se trouve ce terme. Pour cela, une opération de désambiguïsation est nécessaire. [ABD 15]

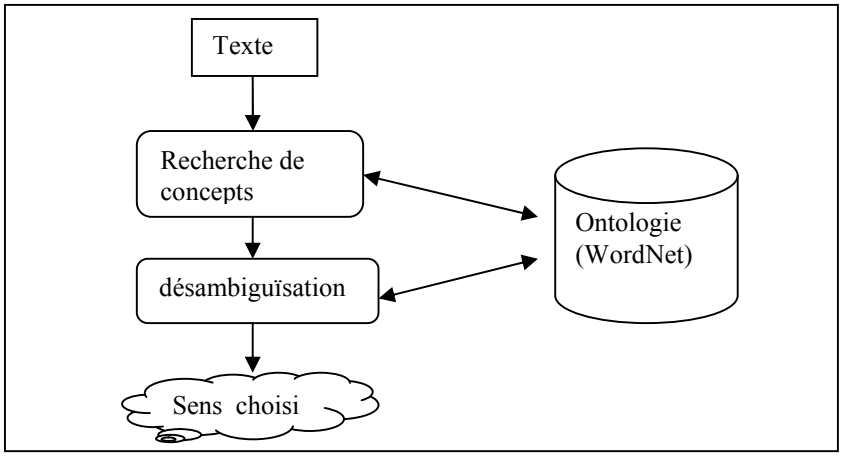


Figure 4.6: Processus d'identification des concept

4.6.3.1. Recherche de concepts

Dans cette étape, nous cherchons si un terme possède une entrée dans l'ontologie de WordNet ; le terme qui a un ou plusieurs sens (1 ou plusieurs synsets) sera conservé, les autres seront éliminés.

Par exemple, pour la phrase " *BBC Radio 4 Poirot radio drama*", nous constatons que les termes BBC et Poirot n'ont pas un synset dans WordNet.

Les deux autres noms ont plusieurs synset (sens). "Radio" a trois sens (figure 7) et "drama" a quatre sens. Ici nous sélectionnons ces termes, afin de les transmettre à la prochaine étape pour lever l'ambiguïté et choisir le meilleur sens du terme, par le calcul de la similarité entre eux.

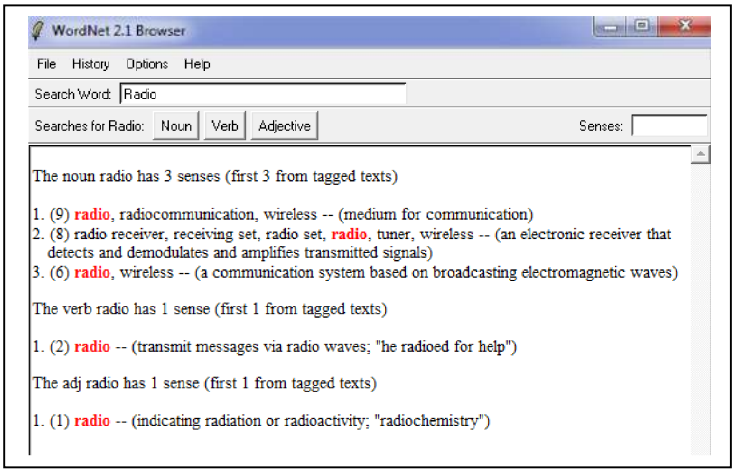


Figure 4.7 : Les sens de "Radio" dans WordNet

4.6.3.2. Désambiguïsation des termes

Après sélection des termes, nous devons maintenant choisir le sens approprié pour chaque terme. Il est fort probable que chaque terme a des significations multiples (synsets) dans WordNet. Ce qui nous amène à calculer la similarité entre les termes afin de choisir le meilleur sens, dans le contexte. Le contexte est défini comme une unité de texte dans un document, il peut être une phrase, un paragraphe, section, etc. [BAZ 05].

Nous définissons un texte comme un ensemble de termes :

$$T = \{t_1, t_2 \dots t_n\},$$

Et nous définissons $C(t_i)$ comme un ensemble de sens pour les t_i dans l'ontologie WordNet :

$$C(t_i)=\{S_i^1, S_i^2, \dots\dots\dots, S_i^n\}$$

Nous voulons définir pour chaque terme $t_i \in T$, le meilleur sens $Sj \in C$ dans le contexte, par le calcul de similarité entre tous les termes de Titre. Ici, nous avons problème combinatoire [RAM 10], où nous devions trouver la meilleure combinaison des concepts parmi les combinaisons possibles d'un concept. La désambiguïsation est un problème combinatoire.

Le nombre de combinaisons possibles est $\prod_1^i | C(\text{ti})|$

Exemple

Nous prenons la phrase suivante (en anglais), qui est un titre d'un document [RAM 10] :

An Information Retrieval Driven by Ontology: from Query to Document Expansion

Nous avons six noms qui ont des synsets dans WordNet (Driven n'a pas une entrée dans WordNet comme nom). T = {Information, Retrieval, ontology, Query, Document, Expansion }.

Chaque terme de T a plusieurs sens (concept) dans WordNet : {Information (5 sens), Retrieval (sens), ontology (2 sens), Query (1 sens), le document (sens) et Expansion (4senses)}.

Pour choisir le meilleur sens pour chaque terme, nous avons à choisir une combinaison parmi les 480 combinaisons possibles (480 = 5 * 3 * 2 * 1 * 4 * 4). Exemple de combinaison:

CB1={Information #08347159, Retrieval#13376715, Ontology#06081744, Query#07094985, Document#06384226, Expansion#00361798}.

CB2={Information #08347159, Retrieval#05690643, Ontology#06081744, Query#07094985, Document#03184230, Expansion#07074115}.

Le tableau suivant représente les différents synset pour chaque terme dans WordNet (version 2.1) :

Terme	**Nombre de sens (synset)**	**Identifiant de chaque synset**	**Glossaire[25] de chaque synset**
Information	5	{06546125}	a message received and understood
		{08347159}	a collection of facts from which …
		{05743526}	knowledge acquired through study or …
		{05031765}	a numerical measure of the ……
		{07138400}	formal accusation of a crime
Retrieval	3	{13376715}	the operation of accessing information computer's memory
		{05690643}	the cognitive operation of accessing information in memory
		{00044242}	the act of regaining or saving something lost
ontology	1	{06081744}	the metaphysical study of the nature...
Query	1	{07094985}	an instance of questioning
Document	4	{06384226}	writing that provides information
		{03184230}	anything serving as a representation of a person's thinking by means of ……
		{13230588}	a written account of ownership
		{06424377}	(computer science) a computer file that contains text
Expansion	3	{00361798}	the act of increasing (something) in….
		{07074115}	a discussion that provides additional information
		{00367403}	adding information or detail

Tableau 4.3: Les différents concepts pour les terme de l'exemple

Pour chaque combinaison CB_K, nous calculons la moyenne des similarités entre tous les synsets, avec la formule suivante [RAM 10]

$$S(\ CB_k\ ) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \boldsymbol{sim}(\ \boldsymbol{c_i}, \boldsymbol{c_j})$$

Où sim (c_i, c_j) est la similarité sémantique entre le concept c_i et c_j de combinaison CB_K calculé sur la base de l'un des mesures de similarité basée WordNet comme (Leacock et Chodorow [LEA 98] ou Resnik [RES 99]....)

[25] Glossaire : est la définition du concept (synset) dans WordNet

Nous devons également sélectionner la combinaison qui a une forte similarité:

$$\mathbf{max=ArgMax(S\ (\ CB_k\ ))}$$

En raison de la complexité du problème de la désambiguïsation, si le nombre de mots des titres est très grand, nous ne pouvons pas calculer les similarités de tous les termes dans le même temps. Nous devons limiter le nombre de mots que nous voulons calculer leurs similitudes, en utilisant une fenêtre de contexte (Crestan 2003), [RAM 10]

L'utilisation des fenêtres réduit le problème de la complexité des calculs de la désambiguïsation où les termes ont une haute ambiguïté et le nombre des termes est très grand.

Comme dans le travail de [RAM 10], nous prenons une taille de fenêtre de 3 termes, puis nous identifions leurs meilleurs concepts dans WordNet. Si nous prenons l'exemple précédent de la figure 4, nous avons une phrase avec 6 termes T = {t1, t2, t3, T4, T5, T6}, le processus de désambiguïsation est comme suit :

1. Pour la fenêtre : {t1, t2, t3}, nous calculons la similarité sémantique, et nous identifions à partir de WordNet le meilleur concept (C1, C2, C3) pour {t1, t2, t3}. Après que nous choisissons (C1 et C2). Le meilleur concept (c3) de t3 sera sélectionné dans l'étape suivante.2. sélectionner le meilleur concept de { t3 } à partir de la fenêtre: {c2, T3, T4}
2. Sélectionner le meilleur concept de {t4} à partir de la fenêtre : {C3, T4, T5}
3. Sélectionner le meilleur concept de {T5, T6} de la fenêtre: {c4, T5, T6}

Nous appliquons ces étapes sur l'exemple précédent, nous aurons :

1. Pour (Information, Retrieval, Ontology), nous identifions :{Information# 08347159, Retrieval#13376715}.
2. Pour (Retrieval#05761380, Ontology, Query) nous identifions: { Ontology#06081744}
3. From (Ontology#06081744, Query, Document} nous identifions: { Query#07094985}
4. Pour (Query#07094985, Document, Expansion) nous aurons :{ Query#07193596, Document#06424377, Expansion# 00367403}

4.6.4. Sélectionner et élargir les concepts importants [ABD 15]

Dans cette étape, nous allons sélectionner les concepts importants qui se trouvent dans les parties importantes de la structure du document, comme par exemple les concepts des titres.

Après la sélection des ces concepts, nous allons chercher d'autre concepts important qui ne se trouvent pas dans les parties importantes de la structure du

document, mais ils ont une relation sémantique avec les concepts importants déjà sélectionner .Pour cela nous prendrons la liste des concepts importants et nous allons la projeter sur la ressource sémantique WordNet pour extraire tous les concepts qui ont une relations avec ces concepts comme les synonymes, les hyperonymes, les Hyponymes...etc.

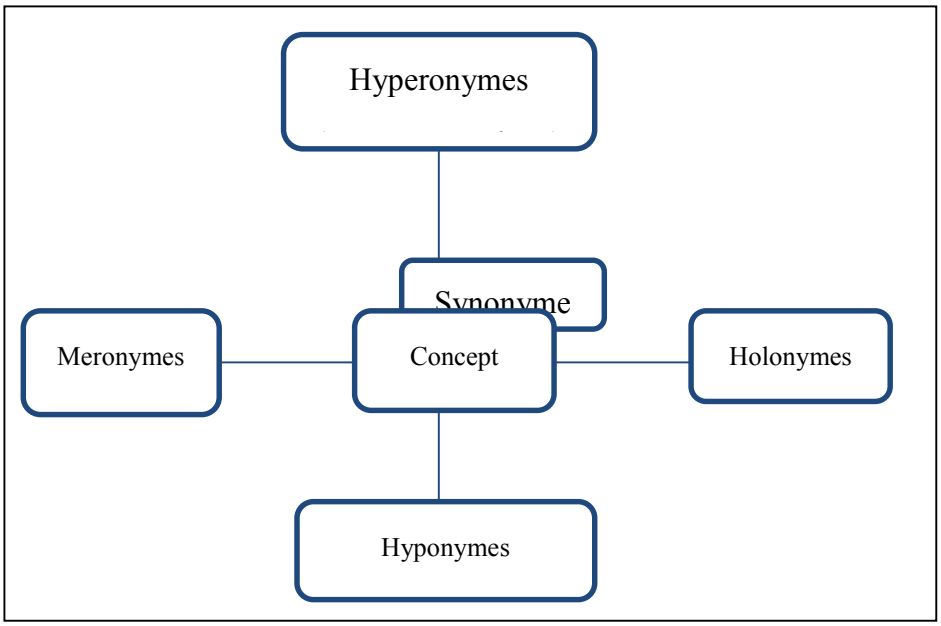


Figure 4.8. Relations Sémantique dans WordNet. (Baziz 2005)

Après la sélection de tous les concepts à partir de WordNet, nous allons rechercher ces concepts dans le contenu textuel du document. Si le concept existe, il sera considéré comme important et il sera ajouté à une liste des concepts importants.

Par exemple le concept "Document #06424377" de l'exemple précédent a comme {text_file} et comme Hyperonymes :{computer_file} et comme Hyponymes :{ web_page, webpage }

Alors on a comme liste de terme à ajouter aux termes importants :{ text_file, computer_file, web_page, webpage}

Nous allons rechercher dans le document ces termes. Chaque terme trouvé sera ajouté à la liste des termes importants. Si le terme n'est pas trouvé dans le document, il sera ajouté à une deuxième liste, qui sera annexé aux documents afin d'élargir le nombre de termes importants. Ces termes auront un poids d'importance moins élevé par rapport à la première liste [ABD 15].

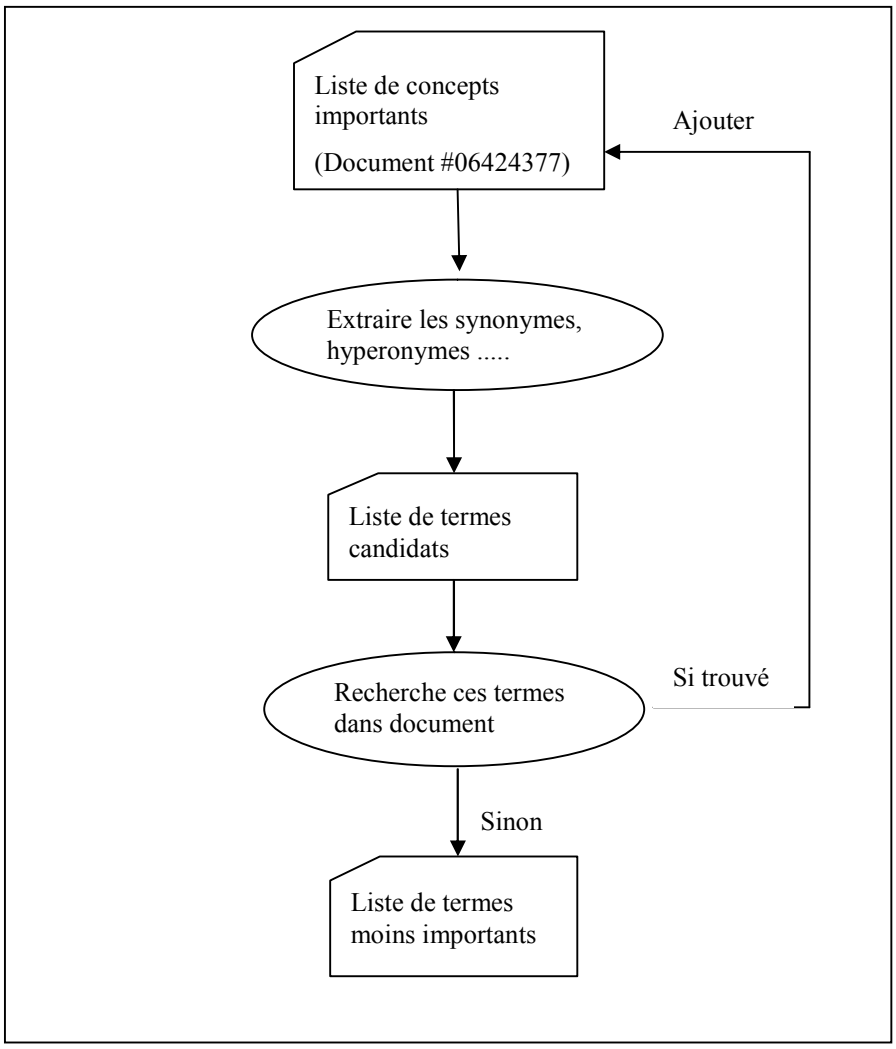


Figure 4.9 : processus de détection des termes important.

Algorithme :

Algorithme qui permet de trouver les termes importants

Algorithme trouver (Liste_de_concepts LC, Ontologie WN, Document D)

{

Pd:= Parser_le_document (D) ;

// Sélectionner un concept important à partir de la liste LC

$\forall$ $c_i \in$ LC {

TC := extraire_terme(c_i , WN)

// extraire les synonymes, hyperonymes,... de C_i

}

$\forall$ $tc_i \in$ TC { Si Rechercher (tc_i , Pd) Alors

Ajouter (tc_i , LC);

Sinon

Ajouter (tc_i , LCM);

}

}

Ou :

LC : Représente la liste des concepts les plus importants dans le document extrait à partir de la structure du document D. ces termes un poids très élevé

D : le document que nous analysons afin d'extraire d'autres termes importants

WN : est l'ontologie WordNet.

Pd: c'est le document D parsé, c.-à-d. une forme du document rendable par la machine

LCM : représente la liste des concepts moins importants. Ce sont les termes qui ont une relation avec les termes qui se trouvent dans LC Mais ne se trouvent pas dans le document D. Ces termes auront un poids moins élevé pendant la phase de pondération des termes.

4.6.5. Indexation

Le processus d'indexation consiste à définir une représentation pour chaque document, pour qu'il soit facile à exploiter par les machines pendant le processus de recherche. Le résultat de l'indexation est un ensemble de termes à pondérer par des poids. Le poids de chaque terme détermine l'importance du terme localement dans le document et globalement dans la collection des documents.

Dans notre système le document sera segmenté en plusieurs champs (Field en anglais), ou chaque champ représente les concepts d'une partie significatif du document (Le titre du document, les titres de sections, les paragraphes...)Chaque champs aura un nom, et un contenu, et chaque champs a un poids d'importance, qui sera attribué aux concepts qui se trouvent dans ce champs.

L'index de notre système est constitué d'un champs pour l'identificateur de chaque document de la collection, un champs pour les concepts importants dans le document, un champs pour les concepts ajoutés et qui ont une relation sémantique avec les concepts importants, et un champs qui contient les concepts non important du document.

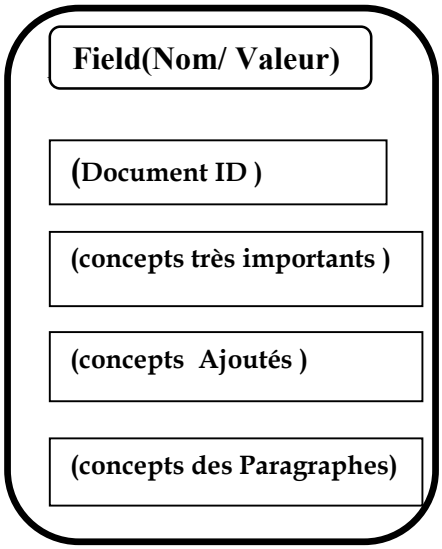


Figure 4.10 : La représentation d'un document indexé

4.6.5.1. Pondération des concepts

Nous considérons, dans notre travail, que les concepts des documents ont une importance différente dépendante de leurs emplacement structurel. Un concept qui se trouve dans le titre principale du document est très important par rapport aux reste des concepts, et les concepts qui se trouvent dans les titres et les sous-titres des sections et sous sections sont important par rapport aux concepts qui se trouvent dans les paragraphes.

Pour calculer le poids des concepts, nous utilisons une extension de la méthode TF-IDF en prenant en compte l'emplacement du concepts dans le document. La pondération du concept ne repose pas seulement sur la pondération locale et globale du terme, qui détermine la distribution du terme dans le document et dans la collection. Mais nous ajoutons aussi un facteur d'importance du concepts dans la structure du document.

Par conséquent, le poids d'un terme sera calculée comme suit [ABD 15] :

$$W_{i,d}=TF_{i,d} * IDF_{i,d} * P$$

Où :

$TF_{i,d}$ Représente la fréquence d'occurrence du concept C_i dans le document d

$IDF_{i,d}$ (en anglais : Inverse Document Frequency) représente l'importance globale du concept dans la collection

P: Représente le poids d'importance du concepts.

4.6.5.2. Vecteur de concepts

Dans notre travail, nous avons utilisé le modèle vectoriel pour représenter les concepts des documents, par un vecteur de poids.

Après la construction du vocabulaire de la collection et le calcule du poids de tous les concepts d'un document, le vecteur sera construit pour chaque document. Il est présenté comme suit :

$$\overrightarrow{d_i} = (w_{1,i} , w_{2,i} , w_{3,i}, ..., w_{n,i})$$

où $w_{k,i}$ est le poids du concepts k dans le document d_i

4.6.6. Appariement requête-documents

Dans notre travail, nous avons utilisé la notion de champs pour fragmenter le document pendant l'indexation.

Notre index inversé est composé de trois champs pour les concepts du document [ABD 14a] [ABD 15]:

- Un champs pour «**concepts importants »** qui sont les concepts importants du document.
- un champs pour « **concepts ajoutés»** contient les concepts qui ont une relation sémantique avec les concepts importants ajouté à partir d'une ressource sémantique [ABD 14a] [ABD 15].
- Un champs pour «**paragraphe»** qui contient le reste des concept du document.

Dans ce travail nous avons utilisé la méthode de calcule de score d'un terme utilisé par le moteur de recherche Lucene de la fondation Apache[26].

Pour rechercher un concept d'une requête dans un document, nous utilisons un modèle simple basé sur la méthode de pertinence TF-IDF.

Nous calculons le score pour tous les champs d'un document, puis nous calculons le score final par la somme des scores de les champs, comme suit:

$$score(q,D) = score(q,f_{\textbf{importants}}) + score(q,f_{\textbf{ajoutés}}) + score(q,f_{\textbf{paragraphe}})$$

$$score(q,f) = coord(q,f) \cdot queryNorm(q) \cdot \sum \Big(tf(t \text{ in } f) \;\; idf(t)^2 \cdot norm(t,f).\, Boost(f) \Big)$$

ou:

tf (t in f) est la fréquence du terme t dans le de champ f du document;

$$tf(t \text{ inf}) = \sqrt{} \text{ freq}$$

[26] https://**lucene**.apache.org/

idf (t) représente l'inverse de la fréquence de terme t dans la collection des documents;

$$idf(t) = \log (numDocs / (docFreq + 1)) + 1$$

où numDocs représente le nombre total de documents dans le corpus, et docFreq, le nombre de documents qui contiennent le terme t.

coord (q, f) est un facteur de score basé sur le nombre de termes de la requête contenue dans un champ spécifié; Un champ contenant plusieurs termes de la requête aura un score plus élevé

$$coord(q, f) = tq / TQ$$

tq : nombre de termes qui sont dans le champ.

TQ : Nombre Totale des termes de la requête

queryNorm (q) est un facteur de normalisation de requête

$$queryNorm(q) = 1 / \sqrt{(\Sigma idf(t)^2)}$$

Norme (t, f) est utilisée pour normaliser la taille des champs (pour faire des champs comparables) un champ plus court aura un score plus élevé.norm (t, f) = 1 / √ (nombre de termes dans le champ)

Boost(f) est le facteur d'importance donné pour chaque champ f.

4.6.7. Représentation de la requête

Après que l'utilisateur a saisi sa requête, elle est analysée par notre système, par le même processus d'analyse des documents. Une phase d'analyse linguistique qui permit de purifier la requête et supprimer les mots intitules, suivi par une étape d'extraction des termes simples et composé.

Notre système permit aussi l'expansion des termes de la requête en suggérant d'autres termes proches aux termes initial de la requête.

Il existe plusieurs travaux d'expansion de requête initial, mais on peut les grouper en trois familles de travaux, les travaux qui utilisent la notion du feedback qui interagit avec l'utilisateur, la deuxième technique utilise une ontologie pour extraire de nouveaux termes. Et une troisième technique qui repose sur l'exploitation de

l'historique des moteurs de recherche en utilisant les termes déjà posés par d'autres utilisateurs.

Dans notre travail on utilise deux méthodes, la première qui exploite une ontologie [BAZ 05], et une deuxième qui exploite les termes importants qui se trouvent dans les titres et les sous-titres titres des documents.

4.6.7.1. Suggestion des termes à partir d'une ontologie

Après l'extraction des termes simples et composés de la requête, ceux-ci sont projetés sur une ontologie afin de les désambigüiser et choisir les meilleurs concepts pour chaque terme. Cette étape de désambiguïsation est déjà expliqué précédemment.

Après avoir choisi les meilleurs concepts, on extrait pour chacun d'eux, à partir d'une ontologie, tous les concepts qu'ils lui sont proches comme les synonymes, les hyperonymes, les hyponymes, les Meronymes et les Holonymes. Ces concepts seront ajoutés aux concept initiales de la requête

Exemple

La requête : "data base query" est composé de deux concepts : data base et query. Après la désambiguïsation, les concepts extraits pour chaque concept sont .pour "query" : question, inquiry, enquiry, interrogation, questioning, inquiring

Pour "data base": information, info, list, listing

Alors la nouvelle requête sera comme suit : data base, information, info, list, listing
query question, inquiry, enquiry, interrogation, questioning, inquiring

4.6.7.2. Suggestion des termes à partir des titres de documents

La deuxième technique qu'on a développée dans notre système, est l'exploitation les concepts importants qui se trouvent dans les titres de documents.

Nous allons créer une base de concepts extrait, et quand un nouveau utilisateur pose ça requête, on compare cette requête avec les titres et on lui propose les concepts d'un titre le plus proche.

4.7. Conclusion

Dans ce chapitre, nous avons proposé deux contributions principales, la première consiste à exploiter les titres et les sous-titres titres des documents afin de structurer ces derniers pour améliorer leur recherche en ne retournant qu'une partie et non pas tout le document.

La deuxième contribution, c'est la modélisation sémantique des titres et les sous-titres titres de document en donnant une importance plus élevée à leurs concepts pendant la phase d'indexation pour avoir des résultats pertinents.

Dans le chapitre suivant nous allons appliquer nos propositions, puis étudier les résultats expérimentaux.

Chapitre 5
Expérimentation et évaluation

5.1. Introduction

Dans ce chapitre, nous présentons une mise en œuvre de nos contributions citées précédemment, afin de juger nos propositions et d'évaluer l'efficacité de notre système de recherche. Nous étudions les deux propositions principales de notre travail à savoir :

- L'extraction des titres et les sous-titres titres qui se trouvent dans des documents non structurés, puis la transformation de documents en format structurel.
- La modélisation sémantique de la structure des documents afin d'améliorer la pertinence des résultats de recherche.

Dans notre travail, nous avons utilisé deux corpus de documents pour implémenter notre système. Un corpus de thèses en format PDF, pour montrer l'importance de la structuration des documents dans la recherche. Et un deuxième corpus de la collection de documents en format XML ; INEX 2009, qui nous permet d'évaluer nos résultats en utilisant des requêtes proposées par cette compagne.

5.2. Environnement Technologique

5.2.1. Langage java

Nous avons mis en place un prototype en utilisant le langage java afin de démontrer la faisabilité de nos contributions. JAVA développé dans les laboratoires de SUN Microsystems[27], présente plusieurs avantages. La plate-forme JVM (Java Virtual Machine) peut effectuer le même code dans plusieurs environnements. Il est fourni gratuitement par Sun Microsystems. En termes de l'environnement de développement, nous avons utilisé Eclipse[28]. Java est choisi parce qu'il existe plusieurs API qui sont utiles dans notre travail. Ces API[29] seront décrites dans ce qui suit.

5.2.2. iText

iText[30] est une API qui permet de créer, adapter, inspecter et entretenir les documents dans le format Portable Document Format (PDF). iText est utilisé par les développeurs Java, .NET[31], et Android[32] afin d'améliorer leurs applications qui traitent le document en format PDF.

[27] https://www.oracle.com/**sun**/
[28] https://**eclipse**.org/
[29] Application programming interface

[30] **itext**pdf.com/
[31] www.microsoft.com/**net**
[32] https://www.**android**.com

5.2.3. Lucene

Apache Lucene[33] est une bibliothèque de logiciels de recherche d'information. Elle est une API libre et open-source, publiée sous la licence Apache Software. Lucene est devenue très populaire grâce à sa simplicité, l'API est utilisé par plusieurs application dans le web ou dans les logiciels bureautiques.

5.2.4. Luke

L'API Luke[34] de la fondation apache, permet d'accéder aux indexes crées par Lucene. Elle permet d'afficher, modifier, et recherche le contenu des indexes .

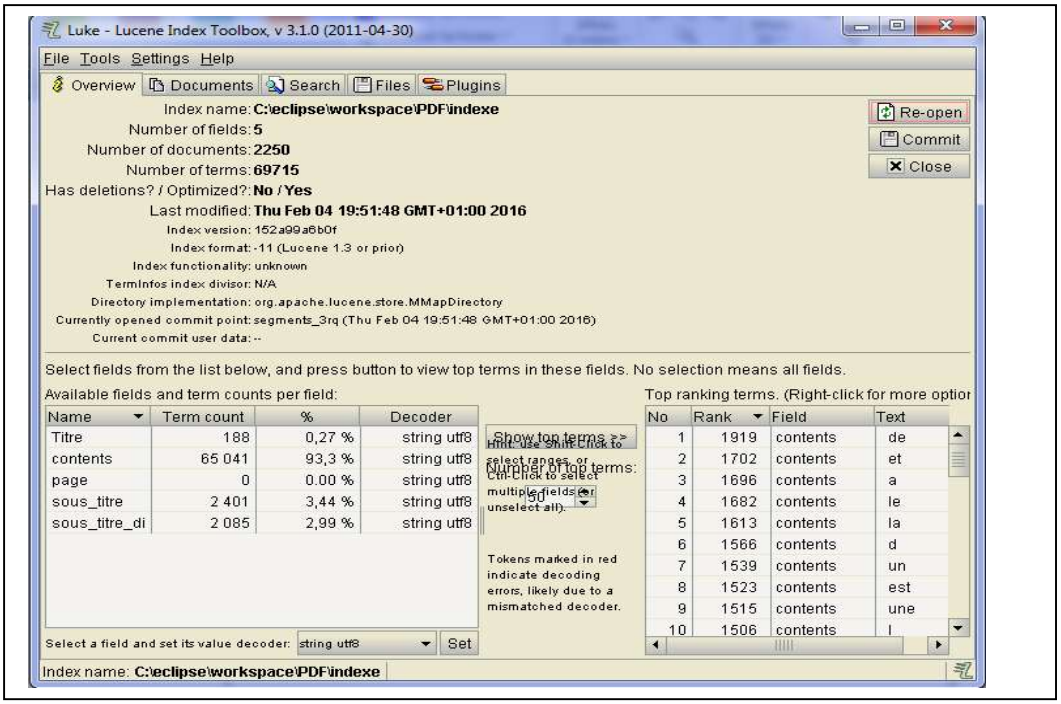


Figure 5.1 : Interface de l'API Luke

5.2.5. POS Tagger

Pour analyser syntaxiquement le contenu textuel des documents, nous avons utilisé l'API Stanford POS Tagger[35]. Cette API permet de lire un texte et extraire des catégories grammaticales des mots comme les verbes, les noms, les adjectifs, les adverbes... etc.

5.2.6. WordNet

Nous avons utilisé l'API JWI[36] (Java Wordnet Interface) qui permet d'accéder à la base de données de la ressource sémantique WordNet.

[33] https://**lucene**.apache.org/core/
[34] http://www.getopt.org/**luke**/
[35] nlp.stanford.edu/software/**tagger**.shtml
[36] projects.csail.mit.edu/**jwi**/

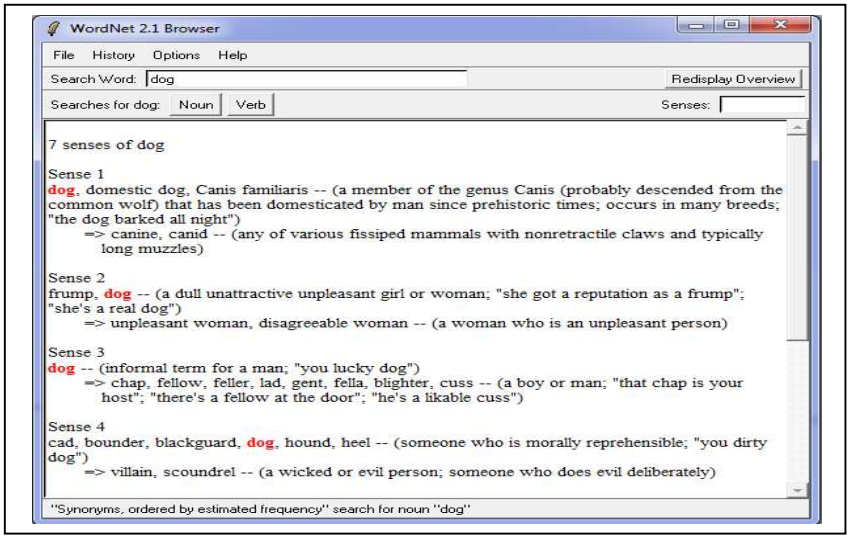


Figure 5.2 : version bureautique de WordNet.

5.2.7. WS4J

WS4J[37] -- (WordNet similarité pour Java) fournit une API Java pour le calcul de similarité sémantique entres les termes. Celle-ci propose plusieurs métriques de calcule de similarité avec la version 3 de WordNet.

5.2.8. XML SAX

Afin d'analyser les documents XML, et extraire le contenu textuel, nous avons utilisé le parseur SAX de Xerces[38].

5.2.9. INEX_Eval

L'API INEX_Eval[39] permet d'évaluer les résultats de recherche de documents XML. Cette API est fourni par la compagne INEX avec une collection de documents XML et un ensemble de requêtes. INEX_eval permet d'évaluer les résultats retournés pour chaque requête, en les comparant avec le fichier de jugement de pertinence fourni par la compagne.

5.3. Modélisation structurelle des documents PDF

Pour la validation de notre approche, nous avons réalisé un prototype de recherche de documents, qui permet en première étape, l'extraction des titres à partir des documents qui sont des thèses en PDF, puis l'indexation et la recherche dans ces documents, en exploitant les titres extraits.

Dans notre système, le résultat est un ensemble classé de parties pertinentes des documents, et qui comporte pour chaque partie : le titre général du document, le ou les titres

[37] https://code.google.com/p/**ws4j**/
[38] https://**xerces**.apache.org/**xerces**2-j/**sax**.html
[39] www.**inex**.otago.ac.nz/

des objets logiques du document correspondant à la requête avec numéro de la page où il se trouve, et les objets logiques (sections, paragraphes,) contenus.

Par exemple : si on a comme requête : « recherche d'informations » on aura comme résultats :

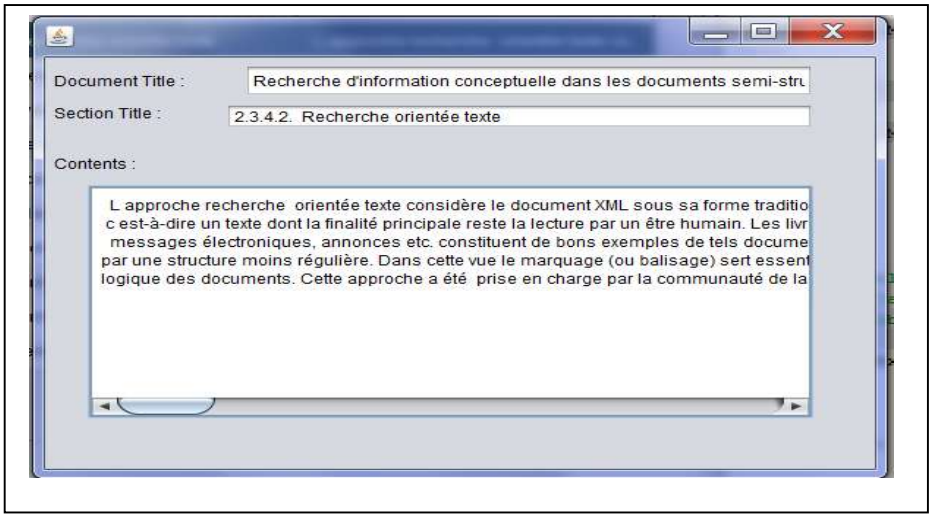


Figure 5.3 : une partie de résultat retourné

Notre prototype utilise la plate-forme open source L*ucene* d'Apache, pour profiter de sa puissance dans l'indexation et la recherche dans les documents textuels.

Nous avons comparé nos résultats avec les résultats du système de recherche thèses scientifiques utilisé par INSA de Lyon DOCINSA ((http://scd.docinsa.insa-lyon.fr/thèses)

Figure 5.4 : Interface du moteur de recherche de DOCINSA

Pour la recherche d'une thèse dans ce site, on peut utiliser soit la recherche par année, par Auteur, ou bien une recherche avancée pour affiner les critères de recherche. La recherche avancée nous permet de choisir le type de document (thèses, ressources pédagogiques, ou publications scientifiques), de définir aussi les critères documentaires tels que, la recherche dans le titre, dans le résumé, l'auteur, la langue, l'année... etc. Ainsi de retrouver les documents avec accès libre ou restreint.

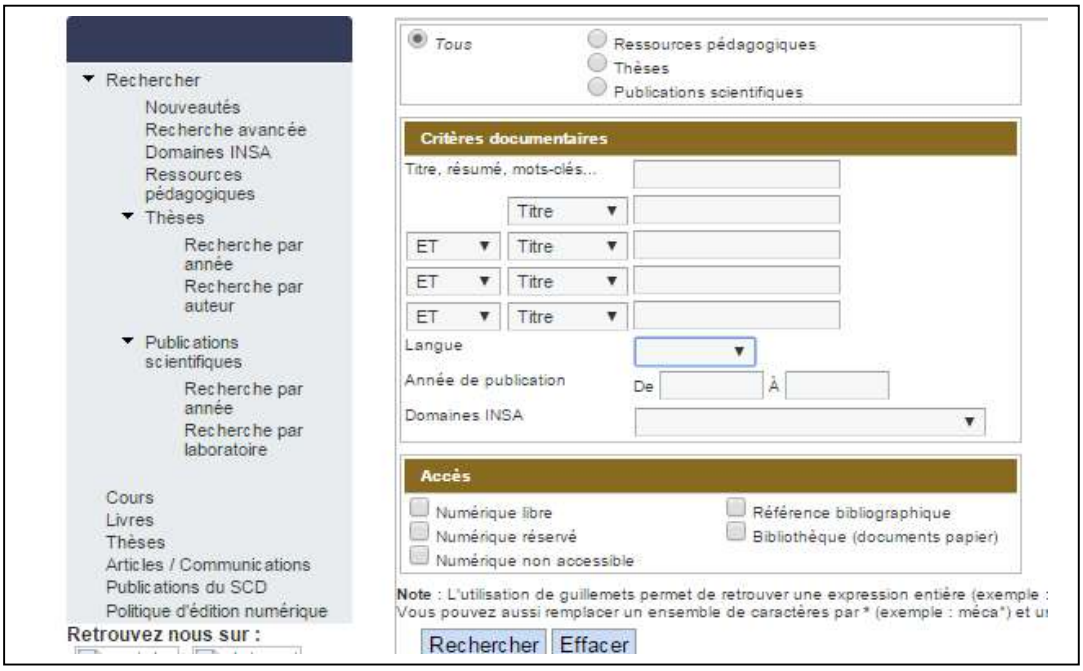


Figure 5.5 : Fenêtre de Recherche avancée de DOCINSA

Après le lancement d'une recherche, des documents seront retournés, on voit dans les résultats, l'auteur de la thèse, le titre, l'année de soutenance, le type d'accès libre ou non), et des liens vers: le résumé de la thèse en français, la notice de la thèse qui est fenêtre de description des métadonnées de la thèse (titre, résumé en anglais et en français, domaine, auteur, date),...., lien vers droit d'utilisation, et un lien vers le texte intégrale qui permet de voir tout le document ou bien le télécharger en format PDF.

Figure 5.6: Une Notice d'une thèse dans DOCINSA.

5.3.1. Corpus

Pour une comparaison objective entre notre système et le système de DOCINSA, nous allons utiliser le même corpus, qui est l'ensemble de thèses déposé dans la base du système DOCINSA dans la période entre l'année 2009 et l'année 2013. [ABD 14b].

Pour récupérer ces thèses que nous utiliserons dans notre expérimentation, nous avons lancé une recherche de thèses dans le système (en précisant la période (2009 - 2013), et en précisant aussi comme critères : les thèses en langue française, et les thèses qui sont en accès libre. Nous avons récupéré avec cette recherche 313 thèses en format PDF. dans tous les domaines couverts par ce système (dans le tableau qui suit, le nombre de thèses el leur domaine) :

Domaine de thèses	Nombres de thèses	Domaine de thèses	Nombres de thèses
Biosciences	21	Chimie	2
Electronique	42	Génie Industriel	7
Energétique	13	Physique	2
Environnement	22	Urbanisme	6
Signal Images et Télécommunications	33	Informatique	34
Automatique	7	Matériaux	51
Génie civil	17	Mécanique	56
		Totale des thèses	313

Tableau 5.1 : thèse extraite de DOCINSA. [ABD 14b]

5.3.2. Evaluation de l'extraction

Après la récupération des thèses, nous avons lancé, la fonction d'extraction de titres à partir de la table des matières, après avoir converti ces thèses vers le format texte en utilisant la plate-forme Itextepdf. Le tableau suivant montre le résultat de l'extraction :

Nombre total de thèses.	313
Nombre de thèses bien converties en format texte.	295
Pourcentage d'erreurs de conversion.	5,75%
Nombre de table de matière bien extrait	287
Pourcentage d'erreurs	8,30%

Tableau 5.2 : Résultat d'évaluation de l'extraction des titres [ABD 14b].

Le nombre de thèses dont nous n'avons pas pu extraire la table des matières est 26. Ces échecs sont dus à des erreurs dans la phase de conversion.

Comme le montre le tableau 2, 18 thèses n'ont pas pu être converties en totalité : la conversion s'est arrêtée après quelques pages (voir, n'a pas commencé), les 8 autres échecs sont causés par des erreurs de conversion du texte telle que la perte d'espaces séparateurs dans le texte extrait, comme nous l'avons déjà évoqué dans la section 4.1, où le titre « table des matières » devient : « tabledesmatière » où « t a b l e d e s m a t i è r e ».

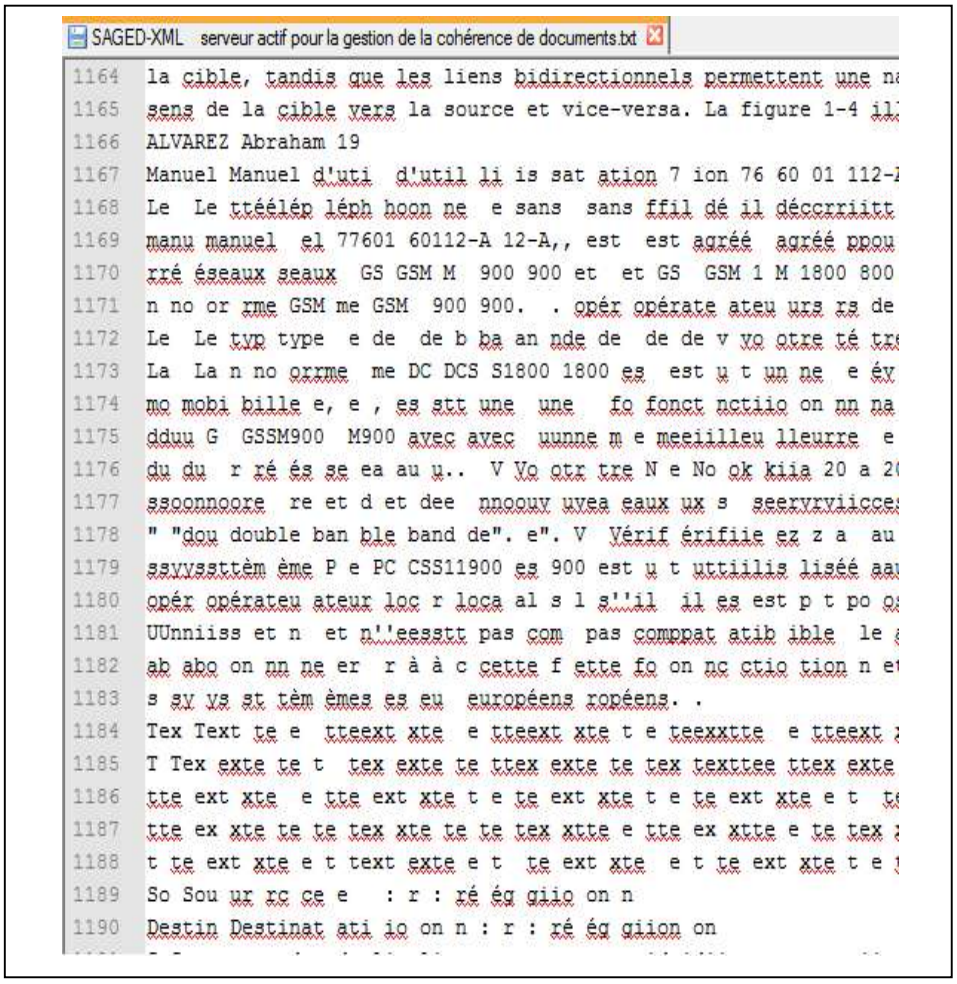


Figure 5.7 : Exemple d'erreurs de l'extraction de texte à partir d'un PDF

Finalement nous prenons en considération dans la phase d'indexation : 313 titres principaux de documents, 295 contenus textuels, et 287 hiérarchies de titres d'objet logique (titres de la table des matières).

5.3.3. Transfert des Documents PDF en format XML

Pour mieux indexer le fragment textuel de chaque document PDF, il faut transformer ces documents en format XML en exploitant le format structurel des titres et les sous-titres de sections.

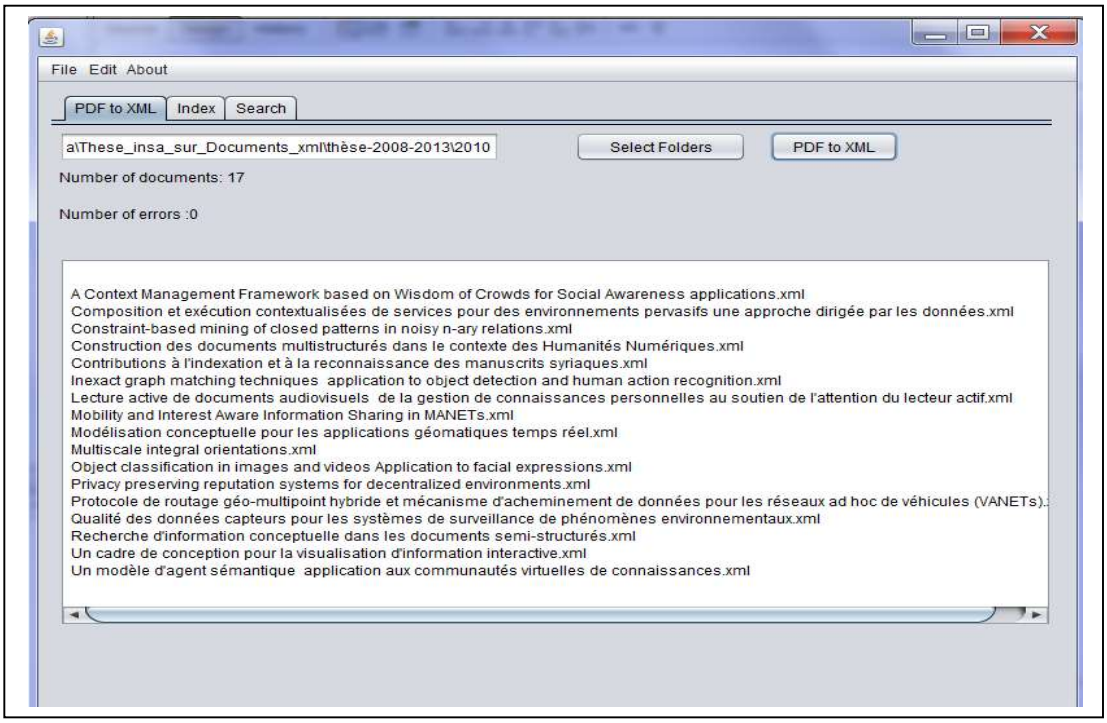


Figure 5.8 : Interface du Prototype pour convertir les thèses PDF en XML

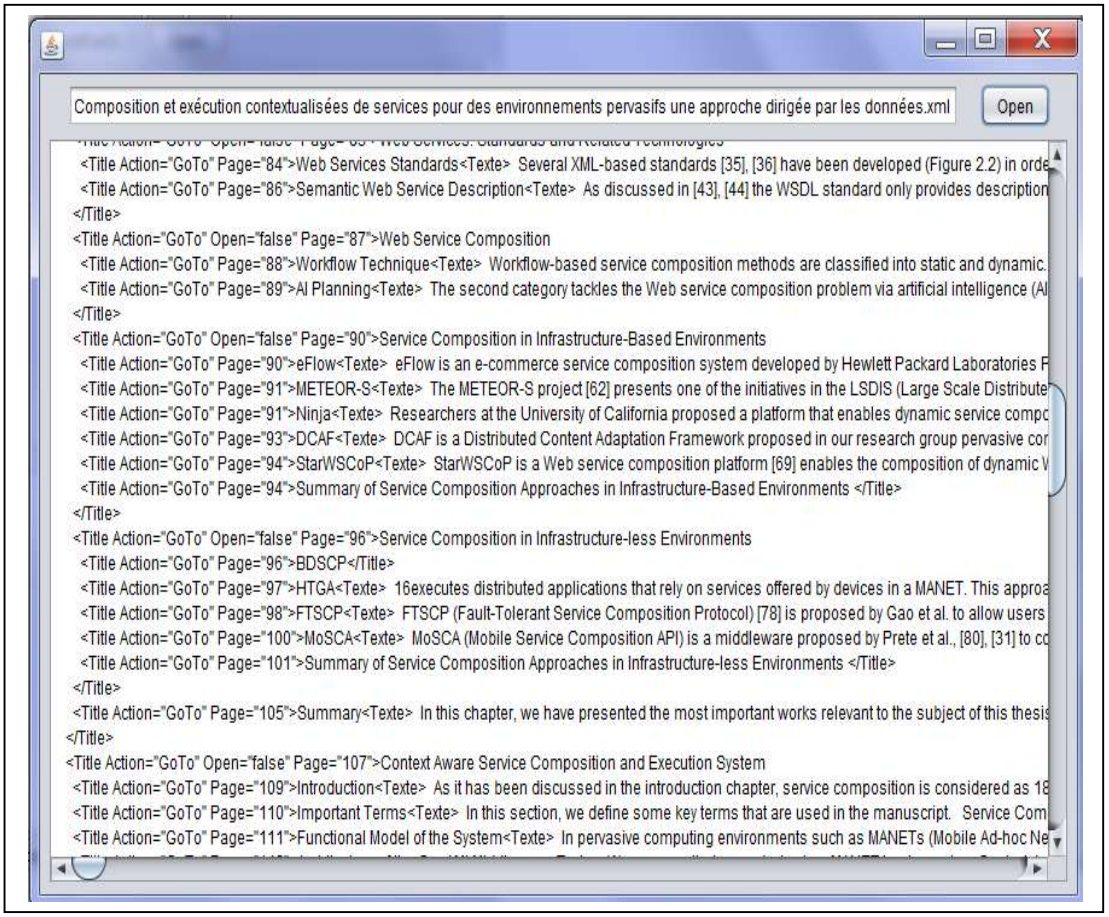


Figure 5.9: Une partie d'un document XML générée à partir d'une thèse PDF

5.3.4. Evaluation de la recherche

5.3.4.1. Indexation

Dans l'étape précédente nous avons extrait le texte à partir du document PDF puis modélisé sa structure logique sous forme d'un document XML. l'étape suivante consiste à indexer l'ensemble du document. à la phase d'indexation le document sera représenté sous forme de trois champs (Field en anglais). Le premier champ pour les termes du titre principale du document. Le deuxième pour le titre de section et le troisième pour les termes du texte.

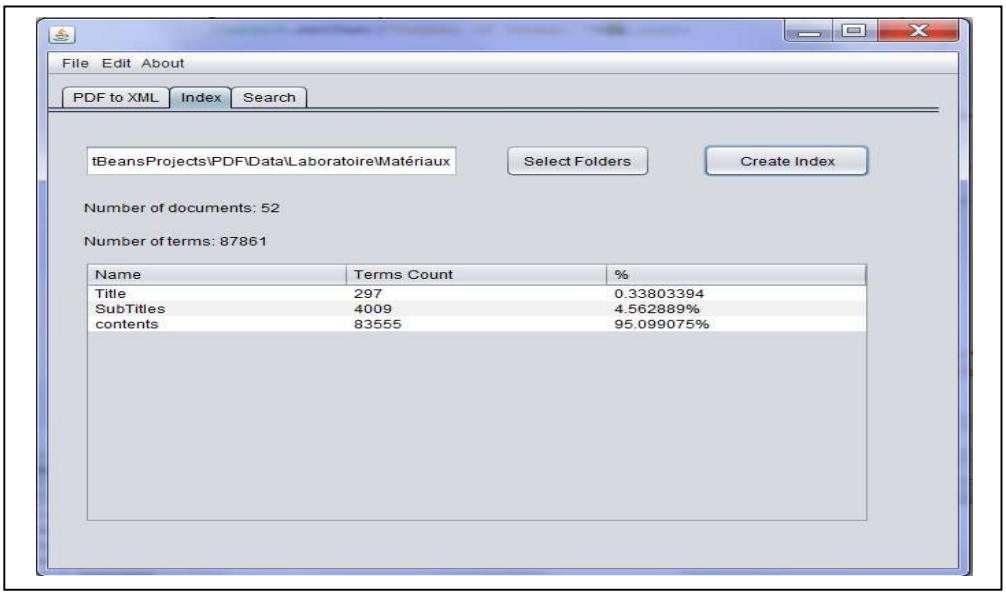


Figure 5.10 : la phase de l'indexation.

5.3.4.2. Recherche

Afin de comparer les performances de notre système à celles du système de DOCINSA nous avons utilisé les deux requêtes suivantes comme exemple de teste :

« WordNet » et « recristalisation alliage» [ABD 14b].

Pour la première requête « wordnet », le système DOCINSA ne retourne aucun document qui contient ce mot, tandis que notre système retourne 5 documents, dont 2 contiennent une section qui parle de wordnet et dont le titre contient ce mot. Les 3 autres documents citent le mot sans qu'il se trouve dans le titre. Ceci montre que notre système diminue le **silence.**

Figure 5.11 : Résultats du moteur DOCINSA pour la requête : WordNet

De plus, notre système retourne les fragments avec les titres d'objets logiques (chapitre, section , paragraphe) qui comportent des termes de la requête (voir la Figure suivante :

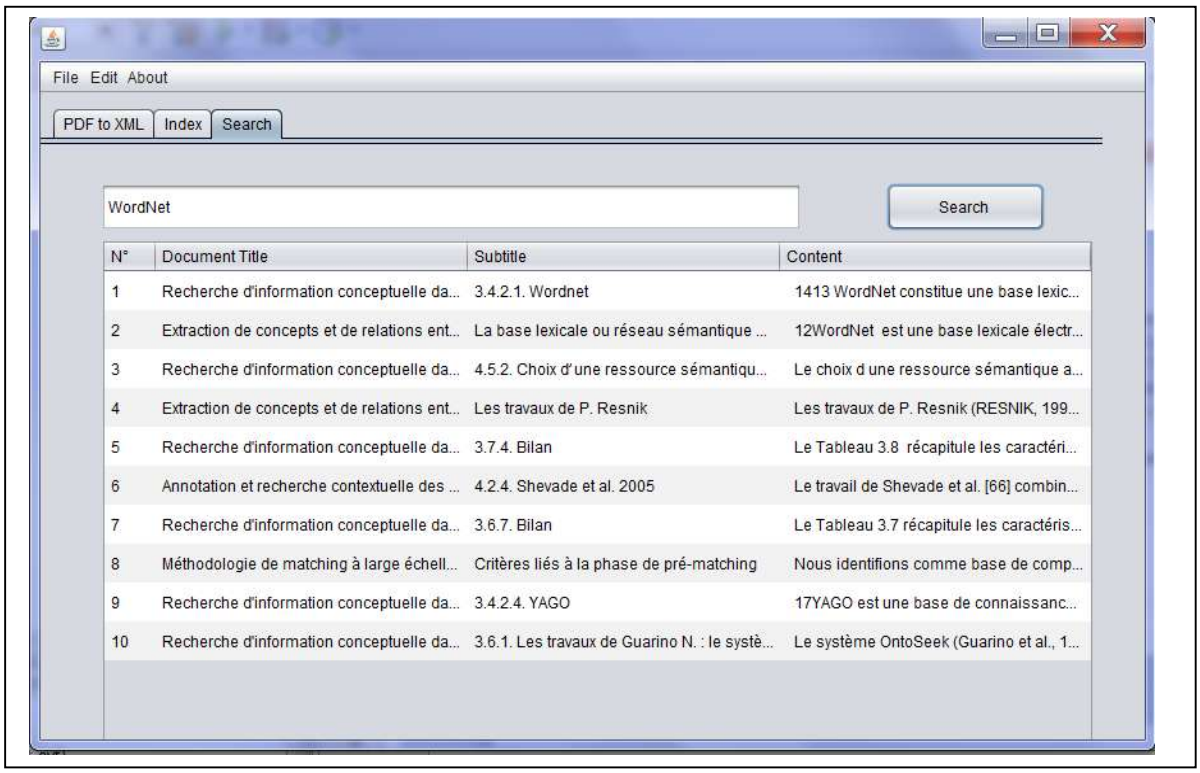


Figure 5.12 : résultats de notre système pour la requête « wordnet ».

En ce qui concerne la deuxième requête « recristalisation alliage» , dans les deux systèmes on lance une requête avec l'opérateur logique **OU** « recristalisation **OU** alliage» c.-à-d. on demande aux deux systèmes de trouver des documents qui contiennent au minimum un mot parmi les mots de la requête. Notre système retourne 45 documents trouvés sur 313 existants alors que le système DOCINSA retourne 12 documents, et le classement de 10 premiers documents n'est pas le même : le chiffre entre parenthèse à la fin des titres rappelle le classement de notre système.

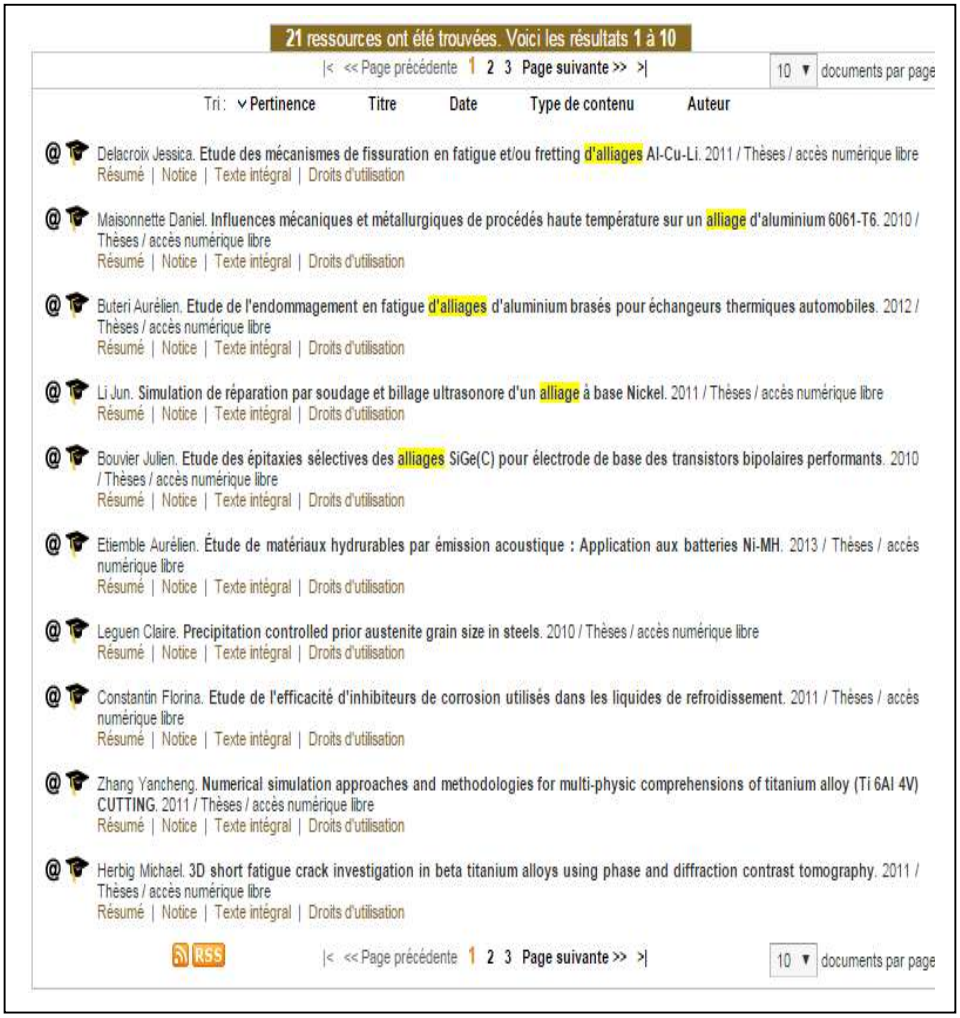


Figure 5.13 : Classement des 10 premiers résultats de DOCINSA

Le tableau suivant montre le classement des 10 premiers documents dans notre système : le chiffre entre parenthèse à la fin des titres rappelle le classement de Doc'INSA

1	Etude des épitaxies sélectives des alliages SiGe(C) pour électrode de base des transistors bipolaires performants **(4)**
2	Simulation de réparation par soudage et billage ultrasonore d'un alliage à base Nickel **(3)**
3	Influences mécaniques et métallurgiques de procédés haute température sur un alliage d'aluminium 6061-T6 **(2)**
4	Etude des mécanismes de fissuration en fatigue et/ou fretting d'alliages Al-Cu-Li. **(1)**
5	Caractérisation du couplage mécano-électrochimique en pointe de fissure lors de la fissuration assistée par corrosion (pas trouvé par DOC'INSA, pourtant pertinent)
6	Caractérisation des films passifs pour la définition de nouveaux matériaux application aux plaques bipolaires métalliques **(10)**
7	Caractérisation de matériaux composite polyacide lactique-bioverre pour application dans la réparation osseuse (pas trouvé par

	DOC'INSA, pourtant pertinent)
8	Elaboration de pseudosubstrats accordables en paramètre de maille à base de silicium mésoporeux pour l'hétéroépitaxie (pas trouvé par DOC'INSA, pourtant pertinent)
9	Systèmes hétérogènes lyophobes Influence de la température et de la vitesse sur les cycles d intrusion extrusion forcées **(8)**
10	Analyse du mouvement d'accessibilité au poste de conduite d'une automobile en vue de la simulation (pas trouvé par DOC'INSA, pourtant pertinent)

Tableau 5.3 : Classement des 10 premiers résultats de notre système [ABD 14b].

Nous remarquons que le classement fourni par le système de Doc'INSA diffère du classement fourni par notre système : Après vérification par un expert humain, le sixième document trouvé par le système de doc'INSA n'est pas pertinent par rapport à la requête. En effet, son contenu ne contient aucun mot de la requête, sauf le résumé qui contient le mot « alliage », mais utilisé une seule fois dans un autre contexte. On peut considérer ce deuxième document comme un **bruit.**

Notre système diminue le bruit dans les premiers résultats par rapport au système de DOCINSA. Dans son classement notre système prend en considération non seulement le titre principal mais aussi les titres des sections et le texte intégral du document. En ce qui concerne les documents trouvés par notre système (tableau 6), ils sont tous pertinents, bien qu'ils ne contiennent pas dans leur titre principal un mot de la requête.

On notera aussi, que le système de Doc'INSA retourne la totalité du document, alors que notre système retourne avec le titre principal, uniquement les titres et les contenus des objets logiques (section ou paragraphe), qui répondent à la requête.

5.4. Evaluation de l'effet de la structure logique sur la recherche

Dans la deuxième partie de notre travail, nous proposons un système de recherche d'information qui prend en compte la signification de la structure logique. Pour évaluer nos résultats, nous étions obligé de s'orienter vers une compagne d'évaluation, Parce que notre corpus de thèses en format PDF, ne nous permet pas de confirmer l'amélioration de résultats. Malgré les tests et les comparaisons avec le système de DOCINSA qui montrent les performances de notre système.

Nous utilisons la compagne d'évaluation INEX (INitiative for the Evaluation off XML Retrieval), qui a été créer en 2002 et qui propose une collection de documents extrait page web en anglais de Wikipedia en format XML. INEX offre la possibilité aux chercheurs d'évaluer leurs méthodes et de comparer leurs résultats.

5.4.1. Corpus

Notre prototype est évalué sur la collection de documents du track ad-hoc de la campagne d'évaluation INEX de 2009, elle est d'environ 50,7 Go de taille et avec plus de 2.600.000 articles.

```
Tony Randall</link></actor>
, <actor wordnetid="109765278" confidence="0.950892767680
<link xlink:type="simple" xlink:href="../068/612068.xml">
Alfred Molina</link></actor>
, and <person wordnetid="100007846" confidence="0.9508927
<actor wordnetid="109765278" confidence="0.91735530291647
<link xlink:type="simple" xlink:href="../210/150210.xml">
David Suchet</link></actor>
</person>
.</p>

<sec>
<st>
Overview</st>

<ss1>
<st>
Influences</st>
<p>

His name was derived from two other fictional detectives
Marie Belloc Lowndes</link>' Hercule Popeau and <link>
Frank Howel Evans</link>' Monsieur Poiret, a retired Fren
<writer wordnetid="110794014" confidence="0.9508927676800
<doctor wordnetid="110020890" confidence="0.9173553029164
<link xlink:type="simple" xlink:href="../335/18951335.xml
Arthur Conan Doyle</link></doctor>
</writer>
</person>
. In <it>An Autobiography</it> Christie admits that "I wa
<link xlink:type="simple" xlink:href="../159/27159.xml">
Sherlock Holmes</link></detective>
 tradition – eccentric detective, stooge assistant, with
<policeman wordnetid="110448983" confidence="0.8">
<person wordnetid="100007846" confidence="0.8">
<preserver wordnetid="110466918" confidence="0.8">
<fictional_character wordnetid="109587565" confidence="0.
<lawman wordnetid="110249459" confidence="0.8">
<imaginary being wordnetid="109483738" confidence="0.8">
```

Figure 5.14 : Un document XML de INEX (titre : Hercule Poirot)

Les documents contiennent des balises qui modélisent la structure logique, comme la balise **<title>** pour indiquer le titre principal du document, la balise pour indiquer les sections, pour sous section de niveau 1 (il existe 4 niveaux), la balise pour indiquer un titre de section, et la balise pour les paragraphes. Il contient d'autres balises qui représentent le format de texte, ou bien des liens vers d'autres pages Wikipédia.

5.4.1.1. Modélisation des documents XML

Il existe Dans les documents XML, du corpus INEX, deux types de balises qui représentent la structure logique du document.

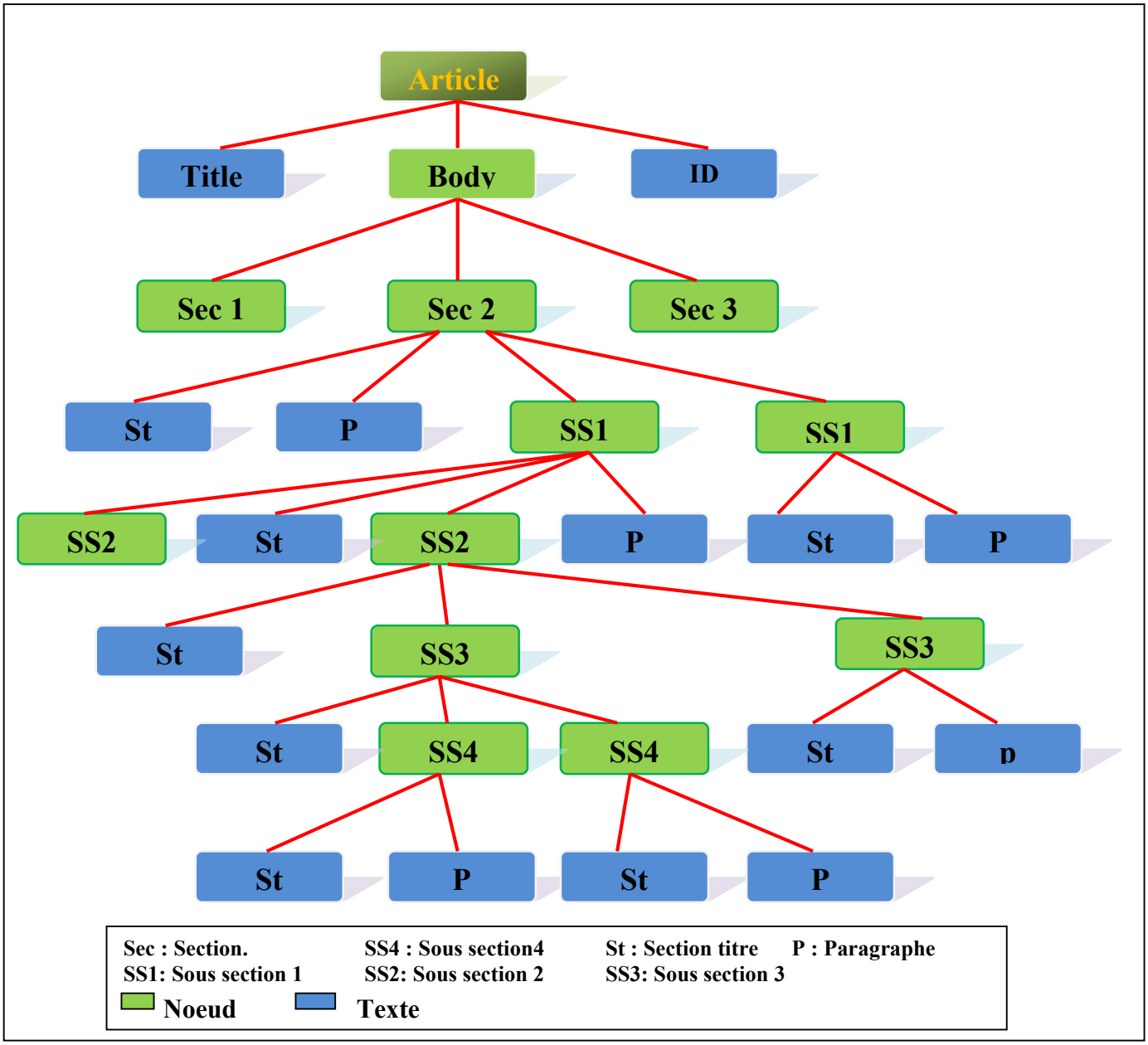


Figure 5.15 : Modélisation d'un document XML sous la forme d'un arbre

Le premier type est constitué de balises nœuds, qui contiennent d'autres balises, tandis que le deuxième type est constitué de balises qui contiennent le texte. Dans notre travail nous avons essayé de modéliser cette structuration afin d'extraire le contenu textuel, mais en gardant leurs chemins dans la structure.

La figure suivante montre une modélisation du document XML sous la forme d'un arbre de nœuds (La structure logique de document), où les relations structurelles entre les balises du document sont bien explicité.

Le résultat de l'extraction de la structure logique de document XML est un ensemble de chemins où se trouvent les éléments textuelle du document. Ces chemins sont exploités pendant la phase d'indexation et la phase de recherche, pour trouver et retourner les éléments les plus pertinents du document par rapport aux besoins de l'utilisateur (Requête), au lieu de retourner tout le document durant la recherche.

Article/Title	**Article/Body/Sec2/SS1/P**
Article/Body/Sec2/St	**Article/Body/S ec2/SS1/SS2/St**
Article/Body/Sec2/P	**Article/Body/Sec2/SS1/SS2/SS3/St**
Article/Body/Sec2/SS1/St	**Article/Body/Sec2/SS1/SS2/SS3/SS4/St**
Article/Body/Sec2/SS1/P	**Article/Body/Sec2/SS1/SS2/SS3/SS4/P**
Article/Body/Sec2/SS1/St	**.....**

Figure 5.16 : Les chemins des nœuds textuels d'un document XML.

5.4.1.2. Création d'un Identificateur

Comme nous l'avons vu dans la figure(12) il existe une relation structurelle entre les sections. Pour sauvegarder ces relations nous avons proposé l'utilisation d'un identificateur spécifique, composé de 7 parties comme suit:

- L'identificateur de document : C'est une valeur extraite de balise <id> du document.
- Le classement de document dans la base documentaire.
- Le numéro de section (Sec) : Le document XML peut en contenir plusieurs sections, dont chacune d'elles est indiquée par un numéro.
- Le numéro de la sous-section section (SS) et son niveau (x). Il existe 4 niveaux de section, la section (SSx) Peut être composée de plusieurs sous sections.

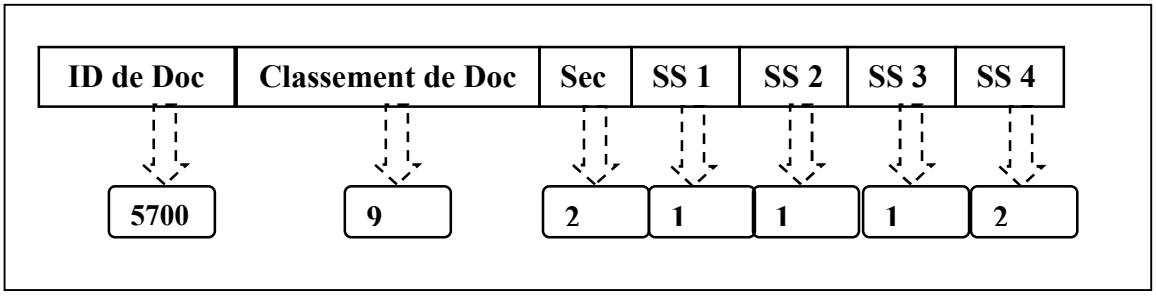


Figure 5.17 : Structure d'un Identificateur

5.4.2. Requêtes

Pour évaluer notre prototype nous avons utilisée des requêtes (topics) proposées par INEX, La compagne a fourni avec la collection de 2009, 115 requêtes (115 topics). Et un jugement de pertinence pour chaque requête. Chaque requête a un identificateur <id> qui détermine son numéro dans l'ensemble des topics, une balise <title> où se trouve le contenu textuel de la requête, une balise < castitle> qui décrit la structure de la requête, et des balises :

Descriptive et narrative, qui contiennent une description détaillée du besoin de l'auteur de la requête.

```
<topic id="2009001" ct_no="186">
  <title>Nobel prize</title>
  <castitle>//article[about(., Nobel prize)]</castitle>
  <phrasetitle>"Nobel prize"</phrasetitle>
  <description>information about Nobel prize</description>
  <narrative>I need to prepare a presentation about the Nobel prize. Therefore, .....
..... </narrative>
</topic>
```

Figure 5.18 : Exemple de requête extrait de INEX 2009

Nobel prize	applications bayesian networks bioinformatics
best movie	olive oil health benefit
yoga exercise	vitiligo pigment disorder cause treatment
mean average precision reciprocal rank references precision recall proceedings journal	native american indian wars against colonial americans
chemists physicists scientists alchemists periodic table elements	content based image retrieval
opera singer italian spanish -soprano	Voice over IP
financial and social man made catastrophes adversity misfortune -"natural disaster"	cycle road skill race
election +victory australian labor party state council -federal	rent buy home

Tableau 5.4 : Partie de reqeute de INEX 2009

5.4.3. Evaluation de la recherche

La compagne INEX fournie un ensemble de jeu de requêtes (topics) et pour chaque requête, des jugements de pertinences, qui seront utilisés pour évaluer nos résultats.

L'objectif d'utilisation de cette collection, est celui de permettre l'utilisation de son outil d'évaluation : inex_eval ; qui permet de calculer la précision iP[x] dans des points de rappel x, où x = {0.00 ; 0,01 ; 0,02…… ; 100}

Nous avons utilisée la tache Ad hoc d'INEX qui est considérée comme une simulation de l'utilisation d'une bibliothèque [HAR 10], quant aux métriques utilisées dans cette tâche elles sont :

1. La précision interpolée selon quatre niveaux de rappel : [$\boldsymbol{r}$]

   ($\boldsymbol{r} \in [\mathbf{0}.\mathbf{00}, \mathbf{0}.\mathbf{01}, \mathbf{0}.\mathbf{05}, \mathbf{0}, \mathbf{1}]$).

2. La moyenne des précisions moyennes (AiP) interpolées selon 101 niveaux de rappel : **MAiP**. Elle est calculée comme suit :

   a. Pour une requête **r** la moyenne des précisions interpolées AiP qui mesure la performance globale, est calculée selon les 101 niveaux de rappel ([0.00, 0.01, 0.02……,1,00]] :

   $$\text{AiP(r)}= \frac{1}{100} \sum_{x=0.00\,,0.01\ldots..,1.00} iP[x]$$

Ou : iP[x] est la précision dans le point de rappel x

   b. La performance globale MAiP pour n requêtes est calculée comme suit :

   $$\text{MAiP}= \frac{1}{n}\sum_{r=1,2,\ldots\ldots n} AiP(r)$$

La campagne INEX utilise la précision interpolée 1 % de rappel (iP [0.01]] comme mesure officielle. Nos résultats, sont évalués avec l'outil d'évaluation inex_eval, sur la tache focused[40]. Les résultats (cf § 5.3) montrent une amélioration dans MAiP et aussi dans la précision, pour les premiers points de rappel (ip[0.01]), quand on exploite les titres avec le contenu dans la recherche d'information.

L'évaluation dans INEX privilèges la précision au détriment du rappel, dont le classement est basé sur la mesure iP [0,01] : la précision dans le point 0,01 [MAT 12]

Organisation	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
Université de Waterloo	0.7657	0.6873	0.5700	0.4879	0.2071
Institut Max-Planck d'Informatique	0.6804	0.6795	0.5807	0.5265	0.2967
LIG Grenoble	0.7114	0.6665	0.5210	0.4216	0.1441
Université de Lyon3	0.6664	0.6664	0.6139	0.5540	0.3065
Université de Saint Etienne	0.6918	0.6640	0.5800	0.4986	0.2342

Tableau 5.5 : Résultats officiels de la tâche Ad hoc d'INEX 2008 [HAR 10]

[40] La sous tache « focused » est une tache ou le système RI doit trouver le meilleur élément ou passage d'un document XML

5.4.3.1. Résultats

Dans notre expérimentation, nous essayons de montrer l'effet d'exploitation des titres de section dans la recherche d'information, ainsi donc nous avons créé un index qui contient trois champs (contenu, titre de section, et titre principal), puis nous avons lancé une recherche sur ces trois champs. Après nous avons comparé les résultats obtenus.

Nous avons utilisé le model TF-IDF de Lucene. Ce model utilise un facteur de normalisation des champs, un champ est plus court aura un score plus élevé. Et comme il existe une grande différence de taille entre les champs (tableau 1), nous avons essayé de diminuer l'écart entre ces champs en modifiant ce facteur pour chaque champ :

Après plusieurs essais, nous avons trouvé une valeur pour le facteur qui permet de donner de bonne résultats, la valeur du facteur est de : 5 pour le champ titre (5 est le résultat de la division de la taille du champ contenu sur la taille du champ titre principal : 37,6% / 7,64%), et 10 (37,6% /2,87%) pour le champ titre de section

Alors la formule de calcul de facteur de normalisation de champs se présente comme suit : le champ contenu, pas de changement :

norm(t,f)=1 / √ (nbr terme dans champ)

Pour le champ Titre principal : norm(t,f)=1 / (5* √ (nbr terme dans champ))

Pour le champ Titres de section : norm(t,f)=1 / (10* √ (nbr terme dans champ))

5.4.3.1.1. Extraction des titres

Dans le tableau suivant sont montré le nombre de termes dans chaque champ, ainsi que leur taille par rapport à la taille totale de l'index, sachant qu'il y a d'autres champs qui représentent le nom des fichiers et le chemin d'accès [ABD 14b].

Le champ	Nombre de termes	Pourcentage dans l'index
Contenu	3 889 525	37,6%
Titre principal	790 045	7,64%
Titres de section	297 340	2,87%

Tableau 5.6 : la taille de chaque champ dans l'index

Il est à remarquer que le corpus contient moins de termes de titres de section par rapport aux termes de titre principal.

Cela s'explique, par le fait que tous les documents du corpus ont un titre principal, mais beaucoup d'entre eux n'ont pas de titres de sections.

5.4.3.1.2. Effet de Titres de sections sur la recherche

Dans notre expérience, nous essayons de montrer l'effet des titres de sections dans la récupération de l'information, donc nous avons créé un indice qui contient trois champs (contenu, les titres de section et titre principal), et puis nous avons lancé la recherche sur ces trois domaines.

Le tableau suivant montre les résultats obtenu, en lançant la recherche sur un seul ou sur plusieurs champs :

Recherche dans le champ :	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
Contenu seul	0.5267	0.5146	0.4575	0.4108	0.1678
Titre principal	0.5642	0.5274	0.3842	0.2949	0.09925
titres de section	0.3423	0.2979	0.2198	0.1787	0.06264
Contenu et titre principal	0.6266	**0.5971**	0.5195	0.4275	0.1577
Contenu et titres de section	0.5524	0.5425	0.4726	0.4291	**0.1753**
Contenu et titre principal et titres des sections	**0.6090**	**0.5777**	**0.5157**	**0.4538**	**0.1772**

Tableau 5.7 : Résultat de la recherche d'information dans les champs d'index.

Le tableau ci-dessus, montre les résultats, dans un, ou plusieurs champs d'index. Comme on a déjà parlé que les métriques officielles d'INEX sont : (iP[0.01]) qui représente la précision, et (MAiP) qui représente la performance globale [ABD 14a].

En comparant les différents résultats avec ceux de la recherche dans le champ contenu seul, on peut constater ce qui suit :

- La recherche dans le titre principal seulement (le champ Titre principal), montre une amélioration, de (2,48 %) dans la précision (iP [0.01]], par rapport à la recherche dans le contenu. Mais une dégradation majeure dans le (MAiP), de (-41 %).
- Mais la recherche dans les deux champs ensemble ; titre principal avec le contenu, améliore mieux la précision (+16,03%), mais n'améliore pas le MAiP, (-6,01%).
- La recherche dans les titres de sections seulement (le champ titres de sections) dégrade la précision (-42,11), et le MAiP (-62,67%).

- Mais la recherche, dans les titres de sections et dans le contenu ensemble, montre une amélioration dans la précision (+5,42), qui reste moins bonne que la précision de (titre principal + contenu : +16,103), et nous constatons aussi uns meilleure amélioration dans le MAiP (+4,46 %)
- La recherche dans les trois champs donne une amélioration dans la précision (+12,26%) et un bon MAiP de (+5,60%) qui est légèrement moins bien que celui de (titres de section + contenu)

Le tableau suivant résume la comparaison de différents résultats avec les résultats de recherche dans le champ (contenu seul) :

Recherche dans le champ :	iP[0.01]	Amélioration	MAiP	Amélioration
Contenu seul	0.5146	-	0.1678	-
Titre principal	0.5274	+2,48%	0.09925	-41%
titres de section	0.2979	-42,11	0.06264	-62,67%
Contenu et titre principal	0.5971	+16.03%	0.1577	-6,01%
Contenu et titres de section	0.5425	+5,42%	0.1753	+4,46%
Contenu et titre principal et titres des sections	0.5777	+12,26%	0.1772	+5,60%

Tableau 5.8 : comparaison des résultats des champs avec le champ contenu

Bilan :

- La recherche en exploitant, le titre principal seul, ou les titres des sections n'améliore pas la recherche, il faut les exploiter en liaison avec le contenu des documents
- On peut conclure que le titre principal donne une meilleure précision dans les premiers résultats. Tandis les titres de section améliorent la performance globale du système de recherche.

Pour bien comprendre notre méthode, nous présentons les courbes de précision aux différents points de rappel, que nous avons obtenus comme résultats de nos expériences.

La figure 1 ; montre que la recherche de document en exploitant leur titre principal donne une meilleure précision dans les premiers résultats retournés, mais elle devient légèrement moins bien quand à la recherche dans le contenu seul, à partir du point de rappel (0.20).

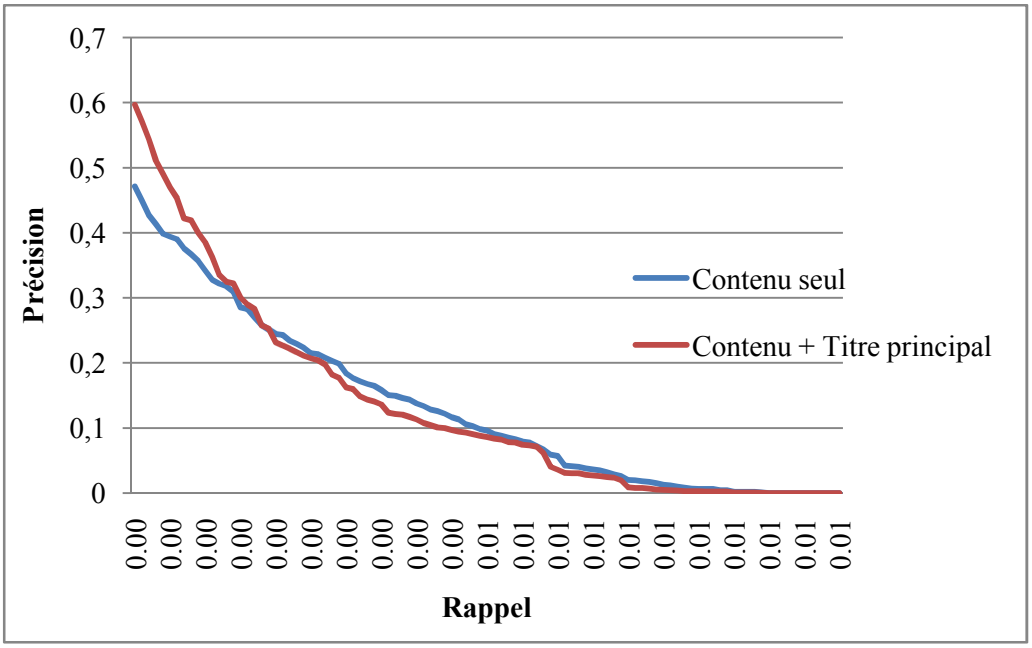


Figure 5.19 : la précision aux différents points de rappel, pour la recherche dans le contenu seul, et dans contenu +titre principal

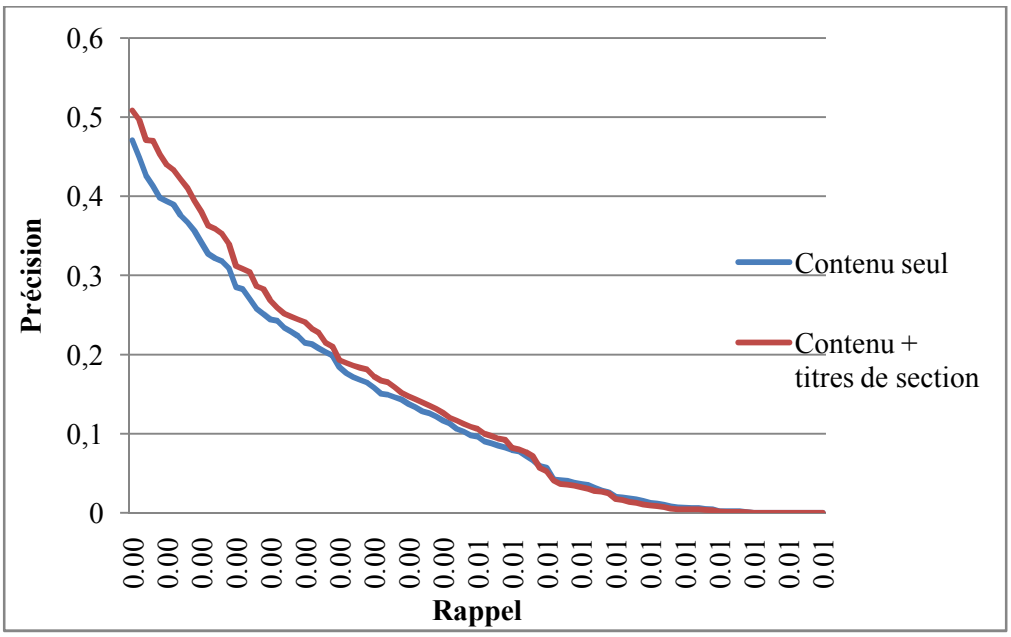


Figure 5.20 : la précision aux différents points de rappel, pour la recherche dans le contenu seul, et dans contenu +titres de sections

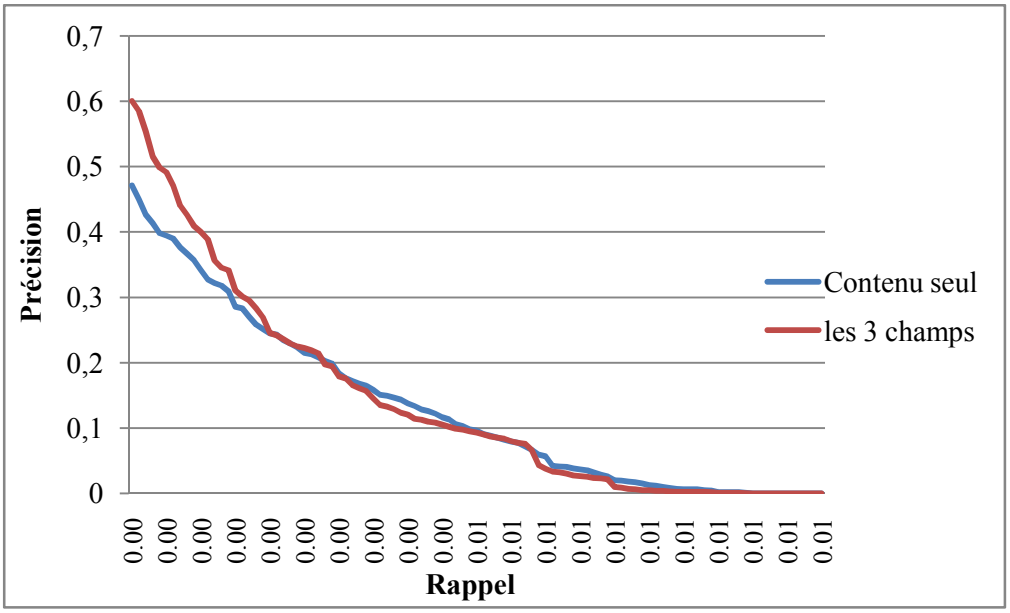


Figure 5.21 : La précision aux différents points de rappel, pour la recherche dans le contenu seul, et dans contenu +titre principal+ titres de sections.

Discussion.

Le nombre de titres de section dans le corpus est trop faible (2,87 % de l'index), et il est plus petit que le nombre de titres principaux (7,64 %). On peut expliquer ça par le fait que tous les documents dans le corpus ont un titre principal, mais elles n'ont pas des titres de sections, pour plusieurs d'entre eux. [ABD 14a]

Malgré ce taux faible, l'exploitation des titres de section dans la recherche d'information, montre une amélioration dans la précision et la performance globale de notre système (MAiP).

5.5. Evaluation de la modélisation sémantique des documents

Dans cette section, nous montrons l'effet de la modélisation sémantique de la structure logique des documents. Pour cela nous avons projeté dans une première étape les termes des titres et les sous-titres des sections sur une ressource sémantique ; WordNet, afin de trouver le meilleur concept qui représente le terme dans son contexte.

Puis dans une deuxième étape, nous avons extrait de nouveaux concepts à partir de WordNet qui ont des relations sémantiques avec les concepts des titres, pour enrichir et augmenter le nombre de termes importants dans l'index.

Comme nous avons montré dans le tableau de la section (44.1) (tableau 7), le nombre des termes des titres principales et les titre de sections est très réduit par rapport aux termes du contenu des documents. Ce taux faible avait un effet sur la précision des résultats, comme nous avons montré dans la section précédente.

Pour combler ce problème nous avons ajouter d'autre concepts proches sémantiquement aux concepts titre. Nous avons ajouté à l'index deux nouveaux champs; le premier est pour les titres principaux étendus, et le second est pour les titres de section étendue. Le tableau suivant montre la comparaison des résultats de récupération obtenus à partir de chaque champ.

Recherche dans le champ :	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
Contenu seul	0.5267	0.5146	0.4575	0.4108	0.1678
Contenu & titre principale	0.6266	**0.5971**	0.5195	0.4275	0.1577
Contenu & titre principale enrichi	**0.6280**	0.5900	**0.5251**	**0.4424**	**0.1736**
Contenu & titres de sections	0.5524	0.5425	0.4726	0.4291	0.1753
Contenu & titres de sections enrichi	0.5502	0.5408	0.4683	0.4283	0.1752
Contenu & titre principale & titres de sections	0.6090	0.5777	0.5157	0.4538	0.1772

Tableau 5.9: résultats de la recherche après l'extension des titres.

Les résultats montrent une amélioration de la précision (iP [0,01]) et la précision moyenne (MAP) lors de l'ajout des termes sémantiquement proches des titres initiaux.

Discussion

Le nombre de titres de sections dans le corpus est trop faible (2,87 % de l'index), il est plus petit que le nombre de titres principaux (7,64 % de l'index). Nous pouvons expliquer cela par le fait que tous les documents dans le corpus ont un titre principal, mais beaucoup d'entre eux n'ont pas les titres des articles. Malgré ce faible taux, l'utilisation des titres de section dans la recherche d'information, montre une amélioration dans les résultats de précision.

Nous voyons aussi que l'expansion de titres, améliore la précision l'iP (0,01) et la précision moyenne, malgré le faible nombre de synonymes ajoutés à ces titres, ce qui nous encourage à obtenir de bons résultats si l'on utilise un autre type de documents qui ont un nombre de titres et de sous-titres plus élevé comme dans les thèses scientifiques ou rapport technique [ABD 15].

```
<eval run-id="2009Run1" file="content_Title_TitleWithAll_0.07.txt">
num_ret         2009001     3000
num_rel         2009001     22
num_rel_ret     2009001     22
ret_size        2009001     6921535
rel_size        2009001     260694
rel_ret_size    2009001     223933
iP[0,00]        2009001     0.9869088603007792
iP[0,01]        2009001     0.9869088603007792
iP[0,05]        2009001     0.9869088603007792
iP[0,10]        2009001     0.9366696191319752
AiP             2009001     0.6741321925038539
num_ret         2009002     3000
num_rel         2009002     73
num_rel_ret     2009002     61
ret_size        2009002     7005697
rel_size        2009002     308522
rel_ret_size    2009002     205788
iP[0,00]        2009002     0.3355130968472583
iP[0,01]        2009002     0.3355130968472583
iP[0,05]        2009002     0.3355130968472583
iP[0,10]        2009002     0.05419853278885192
AiP             2009002     0.05731176765770663
num_ret         2009003     3000
num_rel         2009003     72
num_rel_ret     2009003     71
ret_size        2009003     8987164
rel_size        2009003     297306
rel_ret_size    2009003     239360
```

Figure 5.22 : une partie du résultat d'évaluation.

5.6. Conclusion

Nous avons réalisé un prototype qui permet d'extraire les titres des documents, puis de l'utiliser dans les phases d'indexation et de la recherche d'informations.

En comparant les résultats obtenus, quand on exploite les titres avec ceux réalisé sans obtient l'exploitation de titres, on constate que les résultats obtenus avec l'exploitation de titres sont meilleurs.

Conclusion Générale

Avec les publications en ligne, le Web actuel est devenu une très grande source de documents numériques, souvent stockés sous les formats HTML, PDF ou DOC. Parmi les caractéristiques de ces documents, notons plus particulièrement leur structure logique, qui représente les composants des documents comme les chapitres, les sections, les paragraphes, le titre du document, les titres des chapitres, des sections, …etc.

La structuration logique des documents est porteurs de sens, elle indique une bonne représentation du contenu. Pour cette raison nous avons fait attention particulière à cette représentation au cours du processus d'indexation et de recherche.

Notre objectif est de permettre un accès pertinent aux documents numériques, et cela par l'extraction des parties les plus importantes de la structure des documents, particulièrement les titres et les sous-titres. Puis de présenter ces parties sous un format compréhensible par les machines et facile à le modéliser.

Des expérimentations sur deux types de corpus ont été mené, pour montrer la faisabilité de nous contributions. Le premier corpus est un ensemble de thèse scientifique en format PDF. Nous avons modélisé ces documents pour extraire sa structure et la représenter en format XML. Puis nous avons proposé une méthode d'indexation qui a pris en compte cette structuration, pour retourner ensuite des parties pertinentes des documents pendant la phase de recherche.

Un deuxième corpus de grande taille, qui est INEX 2009, est utilisé pour permettre une évaluation efficace de nos contributions. Le système d'évaluation d'INEX a montré une amélioration dans la précision et la performance globale MAiP quand on exploite les titres et la structure logique des documents. Ces résultats sont meilleur par rapport aux résultats obtenus en exploitant uniquement le contenu des documents.

L'utilisation d'une ressource sémantique WordNet, pour désambiguïser le sens des termes, et calculer la similarité sémantique entre les concepts, a permis aussi d'améliorer les performances de notre système.

La première perspective à tirer de notre travail, c'est de concevoir et de créer une ontologie de domaine dédiée aux thèses scientifiques, pour permettre l'extraction d'autres métadonnées, à partir de la page de garde, comme par exemple le nom de l'auteur, de l'encadreur, des jurys.

Une deuxième perspective c'est de proposer une méthode pour extraire les références et les citations des thèses, et de créer des liens et des relations avec les documents cités.

Une troisième perspective est de perfectionner nos algorithmes d'indexation et de recherche sémantique, en utilisant la notion de concepts au lieu de termes.

Nous pensons aussi à exploiter nos contributions sur des corpus de documents arabes et à réaliser et utiliser des ontologies en langue arabe.

Références de l'auteur

[ABD 15] Abdelli B, Kazar O, Pinon J-M «The impact of titles expansion based on ontology in document retrieval», International Journal of Metadata, Semantics and Ontologies. pp 170-181. 2015

[ABD 14a] Abdelli B, Kazar O, Pinon J-M, «The impact of sections headings on the document retrieval», International Conference on Digital Information Management, ICDIM, Phitsanulok, Thailand. 2014

[ABD 14b] Abdelli B, Kazar O, Pinon J-M «Impact of section headings of a PDF document on information retrieval », International Symposium on Concepts and Tools for Knowledge Management, ISKO-Maghreb, Alger, Algerie. 2014

[ABD 12] Abdelli B, Kazar O, Pinon J-M « An agent-based approach for ontology and semantic search digital documents», International Conference on Information System and Technologies, ICIST Sousse, Tunisie. 2012

Bibliographie

[ABA 05] ABASCAL MENA, «Nouveau modèle de documents pour une bibliothèque numérique de thèses accessibles par leur contenu sémantique», thèse de doctorat, INSA de Lyon, 2005

[ABA 07] Abascal R., Rumpler B, « Accès au contenu des thèses numériques par leur structure sémantique » , Lavoisier *Document* numérique, 2007, Vol. 10, pp 9-35

[AHC 05**]** Ahcene BENAYACHE, « Construction d'une mémoire organisationnelle de formation et évaluation dans un contexte e-learning : Le projet memorae », université de technologie de Compiègne, décembre 2005

[ALA 04] Alain Nossereau « Le document comme contenant, contenu et médium. Les reformulations du numérique »– Education Nationale – 10/11/2004

[ALB 02] Albert Sitruk. La mise en œuvre de la numérisation. Paris :Tec et Doc, 2002. P. 66-67

[AMY 11] Amy Arntson, *Graphic Design Basics* (6th ed.). Cengage Learning. 2011

[ANS 05] ANSI & NISO 2005, Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies, NISO, Maryland, U.S.A

[ARC 12]ARCHANA A. SHINDE, D**.**Text Pre-processing and Text Segmentation for OCR. International Journal of Computer Science Engineering and Technology, pp. 810-812. 2012

[ARN 12] Arnaud Renard, S. Calabretto, B. Rumpler. Une approche de recherche d'information structurée fondée sur la correction d'erreurs à l'indexation des documents, Dans *19ème conférence sur le Traitement Automatique des Langues Naturelles (TALN 2012)*, Grenoble, France. 2012.

[BAZ 05] Mustapha Baziz, "Indexation conceptuelle guidee par ontologie pour la recherche d'information", thèse de doctorat, IRIT, Toulouse. France. 2005

[BAC 98] Bachimont B. *Bibliothèques numériques audiovisuelles. Des enjeux scientifiques et techniques*. Revue Document numérique, 2:3-43-4, 219-242, Hermès, 1998.

[BAE 99] R. A. Baeza-Yates and B. A. Ribeiro-Neto. Modern Information Retrieval. ACM / Addison-Wesley, 1999.

[BAK 05] Bakshi, K., and Karger, D.R. Semantic Web Applications. Proceedings of the ISWC 2005 Workshop on End User Semantic Web Interaction, Galway, Ireland, November 7, 2005

[BEN 08], J. (2008). Préparation des données. In J. Beney, *Classification supervisée de documents* (pp. 69-87). Paris: LAVOISIER

[BEL 00] A. Belaïd et H. Cecotti, La numérisation de documents : Principe et évaluation des performances, Université Nancy 2 - LORIA, 2000

[BEN 08], J. (2008). Préparation des données. In J. Beney, *Classification supervisée de documents* (pp. 69-87). Paris: LAVOISIER

[BER 01] Berners-Lee, Tim; James Hendler; Ora Lassila."The Semantic Web". Scientific American Magazine. May 2001

[BON 04] : Bonnailie Christine, Courcol juliette, Panien Jean, « Le web sémantique », Université de Lille, 2004.

[BOU 03]Mohand Boughanem, Outils de validation en recherche d'information - La campagne d'évaluation TREC, Inforsid Nancy France. 2003

[BOU 11] BOURAMOUL Abdelkrim Recherche d'Information Contextuelle et Sémantique sur le Web. thèse de doctorat. Université MENTOURI de Constantine. Algerie. 2011

[BRO 97] Borst W. N. (1997). *Construction of Engineering Ontologies*. Center for Telematica and Information Technology, University of Tweenty, Enschede, NL.

[CAT 02] Catherine Lupovici. Les choix techniques de la numérisation des documents imprimés dans Conduire un projet de numérisation Paris : Tec et Doc, 2002. P. 133.

[CHA 98] Charlette Buresi. "A propos de la numérisation : Notions et conseils techniques élémentaires" bibliothèques et de la documentation. décembre 1998.

[CHR 09] Christopher D. Manning, Prabhakar Raghavan Hinrich Schütze: Introduction to information Retrieval, Livre, Cambridge University Press. 2009

[CHI 96] Y. Chiaramella, P. Mulhem, and F. Fourel. A model for multimedia information retrieval. Technical report, Technical report, FERMI ESPRIT BRA 8134, University of Glasgow, 1996.

[EDV 09] Edvardsen, L.F.H, Solvberg, I.T, Aalberg, T, et al, « Automatically Generating High Quality Metadata by Analyzing the Document Code of Common File Types », In JCDL, 2009, pp 29-38.

[FEL 05] FELLBAUM, C. WordNet and wordnets. In: K. Brown (Ed.); Encyclopedia of Language and Linguistics. p.665–670. Oxford: Elsevier. 2005

[GRU 93] Gruber, T.: A Translation approach to portable ontology specifications. International Journal of Knowledge Acquisition for Knowledge based Systems, Vol. 5, No.2, 1993

[GUA 98] GUARINO, N. Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy. 1st ed. Amsterdam, The Netherlands, The Netherlands: IOS Press, 1998.

[HAL 07] Lobna Hlaoua, "Reformulation de Requêtes par Réinjection de Pertinence dans les Documents Semi-Structurés". thèse de doctorat. université Paul Sabatier. Toulouse. 2007

[HEB 09] Hebeler, John; Fisher, Matthew; Blace, Ryan; Perez-Lopez, Andrew. Semantic Web Programming. Indianapolis, Edition John Wiley & Sons. 2009

[HEN 08] Hengzhi W, Gabriella K, Michael T, « Book search experiments: Investigating IR methods for the indexing and retrieval of books », Advances in Information Retrieval Lecture Notes in Computer Science, 2008, Vol 4956, pp 234-245

[HER 09]Hervé D · Jean-Luc M «On tables of contents and how to recognize them » International Journal of Document Analysis and Recognition (IJDAR), May 2009, Vol 12, N° 1, pp 1-20

[HOD 04]Ho-Dac, L.-M., Jacques, M.-P. & Rebeyrolle « Sur la fonction discursive des titres » In *L'unité texte* S. Porhiel & D. Klingler (Eds), Pleyben, Perspectives, 2004. pp. 125-152.

[ISO 13] ISO 25964-1 - Thésaurus pour la recherche documentaire, Edition Janvier 2013

[JAC 06] Jacques, M.-P. & Rebeyrolle, J. « Titres et structuration des documents », International Symposium: Discourse and Document, Caen (France), 15-16 juin 2006, pp1-12

[JAY 12]Y Jayabal, C. Ramanathan, et M. J. Sheth ,« Challenges in generating bookmarks from TOC entries in e-books» DocEng , 2012, pp 37-40.

[JEA 05] Jean-Pierre C, Boris C, Hervé D, et al, « From Legacy Documents to XML: A Conversion Framework » Research and Advanced Technology for Digital Libraries, Lecture Notes in Computer Science, 2005, Vol 3652, pp 92-103

[JOE 13] Joeran B, Stefan L, Marcel G, et al, «Docear's PDF Inspector: Title Extraction from PDF Files » in Proceeding of the 13th ACM/IEEE-CS joint conference on Digital libraries, 2013, pp 385-386

[KHA 04] Khan et al. (2004). Khan L., McLeod D., Hovy E.,. Retrieval effectiveness of an ontology-based model for information selection. TheVLDB Journal (2004) 13:71–85.

[KNU 09] Knublauch, Holger; Oberle, Daniel; Tetlow, Phil; Wallace, "A Semantic Web Primer for Object-Oriented Software Developers". W3C, 2009

[LAL 05] Lallich-Boidin, G., & Maret, D. (2005). Recherche d'Information et traitement de la langue. enssib.

[LAL 09] Lalmas, M., & Mary, Q, Structured Document Retrieval. *Encyclopedia of Database Systems* , 2009, PP 2867-2868.

[LEA 13] LEÃO, F.; REVOREDO, K.; BAIÃO, F. A. Learning Well-Founded Ontologies through Word Sense Disambiguation. Proceeding of the Brazilian Conference on Intelligent Systems. p.195–200. Fortaleza - CE - Brazil: IEEE. doi: 10.1109/BRACIS.2013.40, 2013.

[LEA 94] Leacock, C and Chodorow, M. Filling in a sparse training space for word sense identification. March. 1994

[LEA 98]Leacock C. and Chodorow M. Combining local context and WordNet similarity for word sense identification. In Fellbaum, C., editor, WordNet: an electronic lexical database, volume 11 of Language, Speech and Communication, pages 265–283. The MIT Pr, , Cambridge, Massachusetts. 1998

[LIA 11] Liangcai G , Zhi T, Xiaofan L, et al, « Structure extraction from PDF-based book documents », Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries, June 13-17, 2011, Ottawa, Ontario, Canada, pp 11-20

[LIA 09] Liangcai G, Zhi T ; Xiaofan L, et al, «Analysis of Book Documents' Table of Content Based on Clustering» , International Conference on Document Analysis and Recognition, 26-29 July 2009, pp 911 – 915.

[LIN 93] Lin, D.. Principle-Based Parsing Without Overgeneration. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL'93)*, pages 112-120, Columbus, Ohio 1993

[MAL 11] Mallak, I. (2011) De nouveaux facteurs pour l'exploitation de la sémantique d'un texte en RI, Thèse de doctorat, Université Paul Sabatier, Toulouse, France.

[MAI 09]Maisonnasse et al. Maisonnasse L., Gaussier É, Chevallet J-P. Model Fusion in Conceptual Language Modeling. ECIR 2009: 240-251 . 2009

[MAR 07] Mari Vallez; Rafael Pedraza-Jimenez. Natural Language Processing in Textual Information Retrieval and Related Topics , 2007

[MAT 10] Mathias G, Christine L, Franck T, « impact précoce du poids des balises pour la recherche d'information ciblée » CORIA, 2010, 5-7 mai. Toulon, pp xx-yy

[MIL 95] MILLER, G. A. WordNet: A Lexical Database for English. Communications of the ACM, v. 38, n. 11, p. 39–41, 1995

[MIC 10] Michael McCandless, at al, Lucene in action, Livre , Manning Publications 2010

[NAN 98] Nancy Ide, et al, Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art. Computational Linguistics, 24(1):1–40, 1998.

[NAV 99] Navarro, G. (1999). Query languages. In R. Baeza-Yates, & B. Ribeiro-Neto, *Modern Information Retrieval* (pp. 99-116). New York: ACM Press.

[NAV 12] NAVIGLI, R. A quick tour of word sense disambiguation, induction and related approaches. Proceedings of the 38th international conference on Current Trends in Theory and Practice of Computer Science. SOFSEM'12.. p.115–129. Berlin, Heidelberg: Springer-Verlag. 2012.

[NIS 04] NISO press ,Understanding metadata. National Information Standards Organization Press, 2004

[NOU 06] Noureddine CHATTI, Documents multi-structurés : De la modélisation vers l'exploitation; thèse de Doctorat, INSA de Lyon, 2006

[NOY 01] Noy, N. F. and D.L. McGuinness. 2001. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05

[OFF 15] Office québécois de la langue française, *Le grand dictionnaire terminologique*. [en ligne], 2015. Disponible sur : http://www.granddictionnaire.com,

[PET 15] Peter Baofu, The Future of Post-Human Meta-Data: Towards a New Theory of Structure and Process in Information Science, livre , edition lulu.com. 2015

[POR 80] Porter M (1980) An algorithm for suffix stripping. Program 14(3):130–137

[RAM 10]Rami HARRATHI, Recherche d'information conceptuelle dans les documents semi-structurés, thèse de doctorat, INSA de Lyon, 2010

[RAD 08] Radhouani S. "Un modèle de Recherche d'Information orienté précision fondé sur les dimensions de domaine". Thesis de Doctorat, Université de genève 2008.

[RES 99] Resnik O.. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity and Natural Language. Journal of Artificial Intelligence Research, 1999

[ROB 76]Robertson, S. E., & Jones, K. S. (1976). Relevance weighting of search terms. *Document retrieval systems , 27* (3), 129-146

[ROB 77] Robertson SE The probability ranking principle in Journal of Documentation pp 294–304. 1977

[ROB 94] S. Robertson, S.Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC 3. In *Proceedings of the 3rd Text REtrieval Conference (TREC-3)*, pages 109-126, 1994.

[ROD 11] Rodrygo Santos, Richard McCreadie, Vassilis Plachouras. Large-scale Information Retrieval Experimentation with Terrier. CIKM, Glasgow, Scotland, 2011

[ROI 99] Roisin Cécile. *Documents structurés multimédia*, Habilitation à diriger les recherches, Institut National Polytechnique de Grenoble, septembre 1999, 82 p.

[RON 08] Ronan Cummins. The Evolution and Analysis of Term-Weighting Schemes in Information Retrieval, thèse de Doctorat, université de Galway Ireland. 2008

[ROS 11] Rosmayati M, AbdulRazak H, Zulaiha A, « Automatic Document Structure Analysis of Structured PDF Files », International Journal of New Computer Architectures and their Applications, Aug – 2011, Vol 1, n°2, pp 404-411

[SAL 75] Salton G, Wong A, Yang CS (1975) A vector space model for automatic indexing. Commun ACM 18(11):613–620

[SAL 83] Salton G, Fox EA, Wu H (1983) Extended boolean information retrieval. Commun ACM 26(11):1022–1036

[STE 09] Stephen R, Hugo Z «The Probabilistic Relevance Framework: BM25 and Beyond» Foundations and Trends in Information Retrieval, April 2009, Vol.3 N.4, p.333-389.

[STE 11] Stéfane Fermigier , Introduction à la GED et à l'ECM, Cours à l'EPITA, 2011.

[SUC 07]. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. Proceedings of the 16th international conference on World Wide Web, Vol. 6. ACM, Banff, Alberta, Canada (2007) 203-217

[SUS 96]Susan Haigh. La reconnaissance optique de caractères (ROC) en tant que technologie de numérisation. Flash Réseau n°37 ISSN 1200- 5304. 1996

[TOU 00]Toutanova, K. and Manning, C.D. (2000) 'Enriching the knowledge sources used in a maximum entropy part-ofspeech tagger', *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, pp.63–70

[UNI 07] University of Colorado Digital Library, Metadata Best Practices; 2007

[VIC 13] Vicient C. , Sánchez D., Moreno A.. An automatic approach for ontology-based feature extraction from heterogeneous textual resources. *Engineering Applications of Artificial Intelligence*. 2013;26(3):1092–1106.

[WAL 08] Walid M , Kareem D, « Book search: indexing the valuable parts », Proceeding of ACM workshop on Research advances in large digital book repositories, October 30-30, 2008, Napa Valley, California, USA, pp 53-56.

[WOL 00] Wolff, J. E., Florke, H., & Cremers, A. B. Searching and browsing collections of Structural Information. *In Proceedings of IEEE Advances in Digital Libraries (ADL'2000)* , 141-150.

[WU 94] Z. Wu and M. Palmer.. Verb Semantics and Lexical Selection. In *Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics (ACL'94)*, pages 133-138, Las Cruces, New Mexico. 1994

[XUE 07] Y. Xue, Y. Hu, G. Xin, et al « Web page title extraction and its application », Information Processing & Management, September 2007, Vol 43, N° 5, Pages 1332–1347

[YAN 02]YANG, J., & CHEN, X. (2002). A Semi-Structured Document Model for Text Mining. *Computer Science and Technology , 17* (5), 603-610

[YOH 09] Yohannes Tsegay, Document Representation for Efficient Search Engines. thèse de Doctorat, université de RMIT; Melbourne; Australie .2009

[YOU 01] YOUCEF AMEROUALI, Métadonnées basées sur l'association d'éléments de description de ressources et d'éléments de profil d'utilisateur, thèse de doctorat, université Claude Bernard lyon1, 2001

.