Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri de Tizi-Ouzou
Faculté de génie électrique et d'informatique
Département d'informatique

# Mémoire

**de fin d'études**

**En vue de l'obtention du diplôme de Master en informatique**

**spécialité : Systèmes Informatiques**

**Sujet :**

## Contribution à l'indexation sémantique dans la plateforme de RI Terrier

**Proposé et dirigé par :**

**M me F. Amirouche**

**Réalisé par :**

**M elle : Chiout Sarah**

Année universitaire : 2010/2011.

*A mes très chers parents, à mon frère Nassim et à  ma sœur Celine,*

*Je dédie ce modeste travail*

# *Remerciements*

*Mes remerciements, les plus vifs, ma profonde gratitude et mes respects s'adressent à ma directrice Dr Amirouche Fatiha pour avoir accepté de m'encadrer, pour les conseils et orientations tant précieux qu'elle m'a prodigué durant ce Mémoire de Master. Je la remercie aussi vivement pour la démarche fructueuse qu'elle a adoptée pour m'introduire dans ce fabuleux domaine de la recherche d'information.*

*Je tiens aussi à présenter mes vifs remerciements et mes respects à Mme Fellag Samia, Mr Hamache Arezki, Mr Ramdane Mohand pour m'avoir fait l'honneur de juger ce travail.*

*Je ne remercierai jamais assez mes chers parents pour m'avoir toujours encouragée et m'avoir inculquée le goût du savoir et de l'ambition.*

*Que mes chères amies, auprès de qui j'ai toujours trouvé l'encouragement et le soutien moral, trouvent ici ma profonde gratitude.*

# Table de Matières

**Chapitre III : SemTerrier : Vers l'intégration de la sémantique dans Terrier**



**Chapitre IV : Résultats et expérimentations**

# Liste Des Figures & Tableaux

## Chapitre I :



## Chapitre II :



## Chapitre III :



## Chapitre VI :

# Introduction générale :

Le développement rapide des technologies de l'information et de la communication nous ont confrontés à une très grande masse d'informations hétérogènes.

L'introduction de l'Internet a, en effet, complètement bouleversé le monde de l'informatique documentaire en favorisant la multiplicité des ressources documentaires grand public. L'explosion du Web au milieu des années quatre-vingt-dix a propulsé la recherche d'information au premier plan et, désormais, la recherche d'information est devenu un phénomène socio-économique, politique et culturel. En dehors des applications classiques (bibliothèques, gestion électronique des documents, archives et serveurs bibliographiques, etc.…), un nombre important d'applications fait appel aujourd'hui à des solutions de recherche d'information.

Cette expansion des systèmes de recherche de l'information grand public notamment des moteurs de recherche a entraîné à la fois une multiplication et une diversification des usagers et une hétérogénéité croissante des documents. Cette double évolution n'a cependant pas modifié l'objectif fondamental de la recherche d'information : le repérage de l'information pertinente avec le maximum de précision.

En raison de cette augmentation constante du volume d'informations, nous arrivons à une situation paradoxale : jamais il n'y a eu autant d'informations disponibles, mais trouver dans cette accumulation ce que l'on recherche précisément, devient de plus en plus ardu. Le problème n'est plus tant la disponibilité de l'information mais la capacité de sélection de l'information répondant aux besoins précis d'un utilisateur, à partir des représentations qu'il perçoit.

De ce fait, des approches de recherche d'information, regroupant entre autres des techniques d'indexation ainsi que des mécanismes d'appariement ont été développées afin de mieux répondre aux besoins de l'utilisateur.

Les premières approches de recherche d'information, qualifiées de classiques, se basent sur une recherche par mots clés, les documents sont représentés comme des sacs de mots souvent pondérés, et la pertinence d'un document vis-à-vis d'une requête est souvent estimée en s'appuyant sur les fréquences d'apparition des mots de la requête dans ces mêmes documents. Mais, très vite les problèmes inhérents à la richesse des langues, se sont imposés.

L'indexation sémantique tente de pallier ces limites en s'appuyant sur les sens pour la représentation des documents. Notre travail s'inscrit principalement dans ce contexte. En effet, Notre objectif est d'implémenter une approche d'indexation sémantique de documents plats. L'approche est proposée dans le cadre d'un mémoire de magister, que ce travail de master appuie par une implémentation.

Au delà de cette implémentation, notre contribution consiste à étendre la plate forme Terrier de RI à l'indexation sémantique.

Nous présentons, dans le premier chapitre, un état de l'art des systèmes de recherche d'information en montrant la diversité des approches (booléenne, statistique, probabiliste etc...) et les différents mécanismes sous-jacents à la conception de ces systèmes (appariement, indexation, filtrage, reformulation etc...).

Nous décrivons dans le second chapitre un ensemble de travaux et de réflexions autour de l'indexation sémantique. Après une introduction des différentes approches d'indexation sémantique, nous nous focaliserons sur une l'importance de la désambiguïsation et des ressources sémantiques dans ce processus.

Dans le troisième chapitre, nous présentons la partie contribution de ce mémoire en décrivant une nouvelle approche d'indexation sémantique ainsi que son intégration dans la plateforme de recherche d'information Terrier en ajoutant principalement une base de données lexicale et un étiqueteur Syntaxique.

L'expérimentation de Semterrier est présentée d'une manière succincte dans le quatrième chapitre. Enfin, nous concluons en proposons une perspectives de recherche permettant d'améliorer notre système.

# Chapitre I

# Recherche d'information

## I.1. Introduction :

L'explosion du Web au milieu des années quatre-vingt-dix a propulsé la recherche d'information au premier plan et, désormais, la recherche d'information est devenu un phénomène socio-économique, politique et culturel. En dehors des applications classiques (bibliothèques, gestion électronique des documents, archives et serveurs bibliographiques, etc.…), un nombre important d'applications fait appel aujourd'hui à des solutions de recherche d'information. On peut citer par exemple les solutions de gestion de contenu, de veille et d'intelligence économique, de gestion des connaissances, de record management et enfin les portails d'information (extranet et intranet). Historiquement, le terme de « information retrieval » a été proposé par Calvin Moers en 1948 [Moers, 1948] pour désigner le processus d'indexation automatique et de recherche d'information. C'est aussi vers 1951 que Mortimer Taube suggère l'application de l'algèbre de Boole au repérage de l'information en proposant le concept d'uniterm ou mot-clé simple. Les premières expériences sur un ordinateur sont menées entre 1955 et 1965 aux Etats-Unis et en France. La fin des années 70 voit l'arrivée des logiciels documentaires sur mini-ordinateurs, et la décennie 80, ceux sur micro-ordinateurs [Chiaramella, 2007].

La recherche d'information (RI) fournit des techniques et des outils pour trouver les documents contenant l'information pertinente qui répond aux besoins des utilisateurs. L'objectif de ce premier chapitre est de proposer un état de l'art sur les différents SRI.

Les principales questions qui se posent lors de la conception d'un SRI sont alors :

- Qu'est-ce qu'un système de recherche d'information ?
- Comment fonctionne t-il ?
- Quels sont les modèles de RI ?
- Comment évaluer les performances d'un SRI ?

Ce chapitre tente de répondre à ces questions. A cet effet, nous présentons les concepts généraux à la base de la recherche d'information.

# I.2. Principes généraux de la recherche d'information :

## I.2.1. définition d'un SRI :

Un SRI inclut un ensemble de procédures et d'opérations qui permettent la gestion, le stockage, l'interrogation, la recherche, la sélection et la représentation de cette masse d'informations. Il est défini comme suit :

*Un **Système de Recherche d'Informations (SRI)** est un système informatique qui permet de retrouver à partir d'un ensemble de **documents**, les documents pertinents pour un besoin en information d'un utilisateur, exprimé à l'aide d'une requête.*

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

- ***Document*** :

Le document représente le conteneur élémentaire d'information, exploitable et accessible par le SRI. Un document peut être un texte, une page WEB, une image, une bande vidéo, etc…Il s'agit de toute unité qui peut constituer une réponse à un besoin en information exprimée par un utilisateur.

- ***Requête :***

Une requête constitue l'*expression* du *besoin* en informations de l'utilisateur dans un langage de requête. Plusieurs systèmes utilisent des langages différents pour décrire la requête:

- langage à base de mots clés : cas des systèmes SMART [Salton, 71] et Okapi [Robertson, 99],
- langage naturel : cas des systèmes SMART [Salton, 71] et SPIRIT [Fluhr, 85],
- langage booléen : cas du système DIALOG [Bourne, 79],
- langage graphique : cas du système NEURODOC [Lelu, 92].

- ***Pertinence :***

Pertinence est la notion centrale en RI car toutes les évaluations s'articulent autour de cette notion. Elle définit le degré de correspondance entre un document et une requête. C'est une mesure d'informativité du document à la requête.

Il existe deux formes de pertinence :

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

## I.2.2. L'architecture générale d'un SRI:

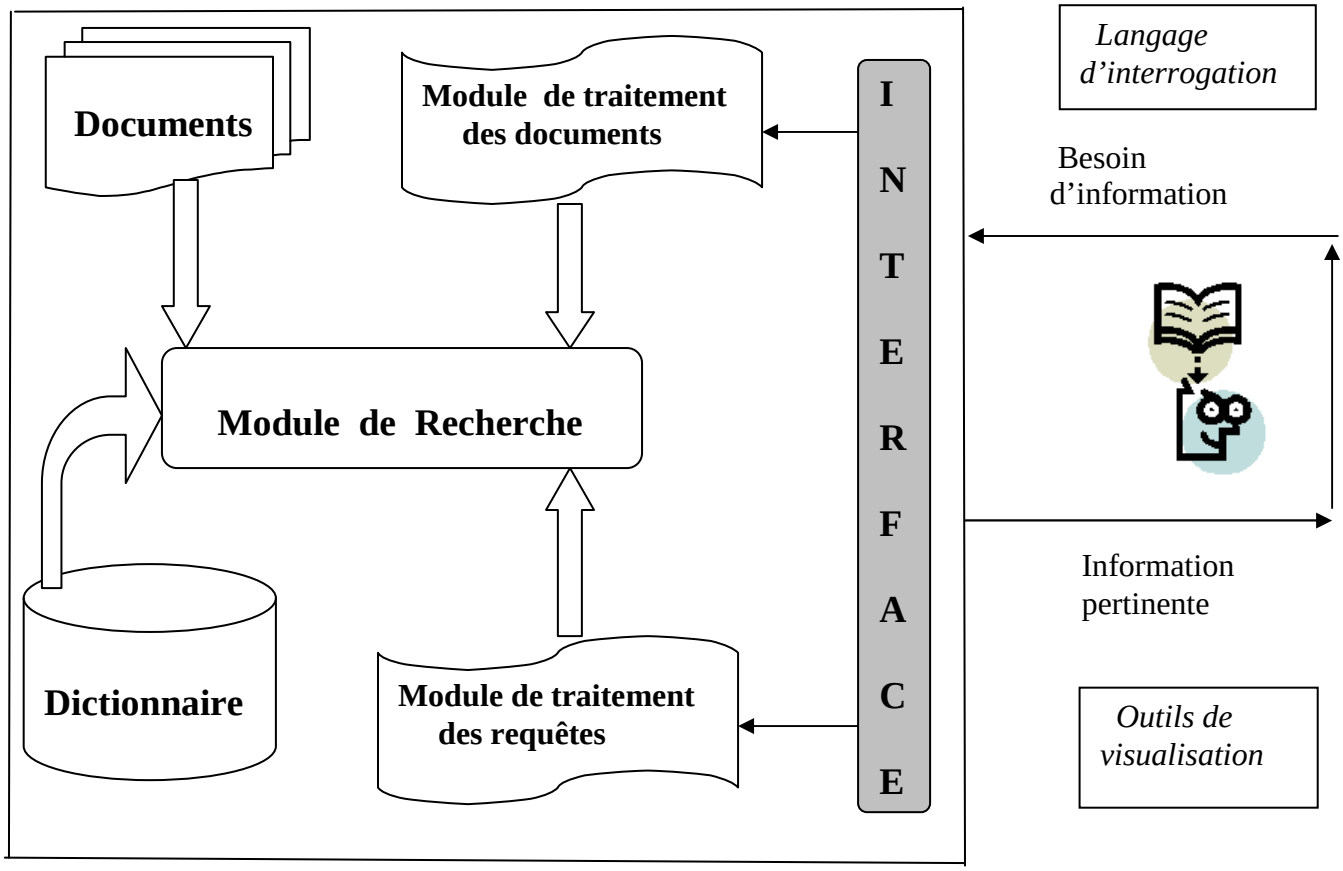


**Figure I.1 : architecture générale d'un SRI [Amirouche, 08].**

L'architecture générale d'un SRI fait ressortir les principaux éléments suivants :

**(1) L'interface** : assure la communication entre l'utilisateur et la base documentaire. L'interface doit être conviviale est ergonomique pour faciliter l'accès aux grand Public.

**(2) Module de traitement des documents** : comprend les outils de gestion de la base documentaire (représentation, organisation, stockage des données).

**(3) Module de traitement des requêtes** : ce module a pour objectif de représenter les requêtes des utilisateurs suivant le modèle choisi pour la recherche d'information.

**(4) Le langage d'interrogation** : c'est le langage dans lequel l'utilisateur formule sa requête. Il existe plusieurs types d'interrogation :

-L'interrogation en langage booléen.

-L'interrogation en langage naturel ou quasi naturel.

-L'interrogation en langage graphique.

**(5) Les outils de visualisation** : un outil de visualisation offre la possibilité de consulter l'intégralité de la base documentaire. Les documents retrouvés par le SRI seront affichés à l'utilisateur sous une forme qui lui permet de consulter aisément l'information pertinente (texte intégral, passage souligné dans le texte, …).

**(6) Module de recherche d'information** : ce module reçoit en entrée deux éléments :

La représentation des documents « ou index de documents » produit par le module de traitement des documents, et la représentation de la requête « ou requête interne » produite par le module de traitement des requêtes.

Le module de recherche calcule le degré de correspondance de ces deux représentations, et fournit en sortie les documents susceptibles d'être pertinents.

**(7) Base documentaire** : une base documentaire contient un nombre important de documents. Ce contenu diffère d'une base documentaire à une autre selon le domaine et le contexte d'application du SRI.

**(8) Dictionnaire** : un dictionnaire est une structure qui comprend les mots clés du domaine de la base documentaire sur laquelle portera la recherche.

## I.2.3. Le processus de RI :

De manière générale, la recherche dans un SRI consiste à comparer la représentation interne de la requête aux représentations internes des documents de la collection. La requête est formulée, par l'utilisateur, dans un langage de requêtes qui peut être le langage naturel, un langage à base de mots clés ou le langage booléen. Elle sera transformée en une représentation interne équivalente, lors d'un processus d'interprétation.

Un processus similaire, dit indexation, permet de construire la représentation interne des documents de la base documentaire. Le processus de recherche consiste alors à mettre en correspondance et à calculer le degré d'appariement des représentations internes des documents et de la requête. Les documents qui correspondent au mieux à la requête, ou documents dits pertinents, sont alors retournés à l'utilisateur, dans une liste ordonnée par ordre décroissant de degré de pertinence lorsque le système le permet. Afin d'améliorer les résultats de la recherche, le système peut être doté d'un mécanisme d'amélioration et de raffinement de la requête par reformulation.

Le fonctionnement général d'un SRI est donné à travers le processus de recherche communément appelé processus en U [Belkin, 92], présenté en **Figure I.2.**

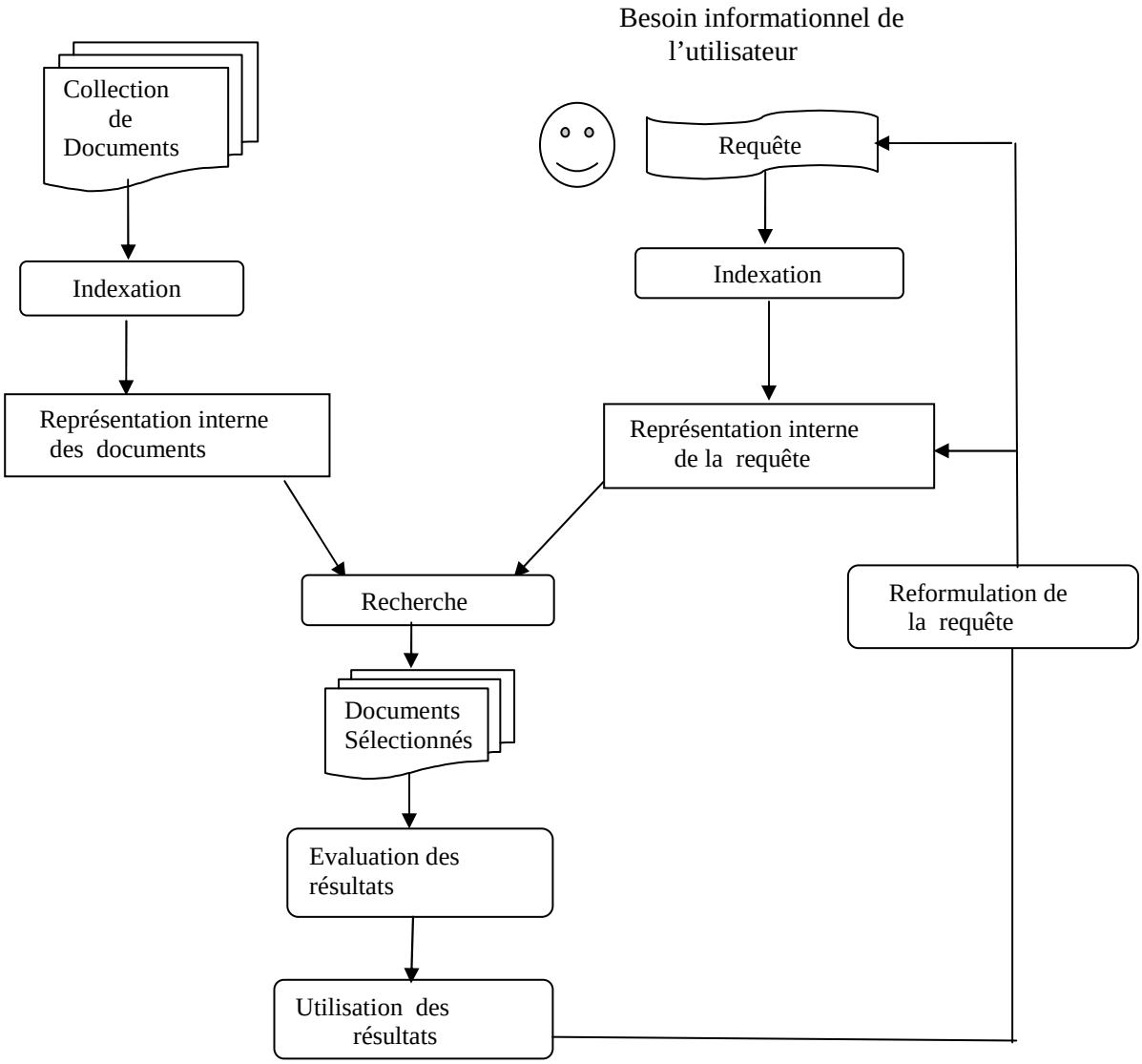


**Figure I.2 : Processus en U de la RI .**

Ce processus fait ressortir trois mécanismes de base : le processus d'indexation (quelques fois dit processus d'interprétation pour les requêtes), le processus de recherche (appariement) et le processus de reformulation des requêtes. Nous les détaillons dans les paragraphes suivants.

### I.2.3.1. L'indexation :

L'indexation est une étape très importante dans le processus de RI. Elle consiste à déterminer et à extraire les termes représentatifs du contenu d'un document ou d'une requête. Cette étape est l'objet de notre étude. Elle sera détaillée en section I.2.4.

### I.2.3.2. L'appariement requête-document :

Le processus d'appariement document-requête permet de mesurer la pertinence d'un document vis-à-vis d'une requête. De manière générale, à chaque réception d'une requête, le SRI calcule un score de pertinence (similarité vectorielle, probabiliste, etc.). Ce score de pertinence est calculé à partir d'une fonction ou d'une mesure de similarité, notée *RSV*(*Q,D*) (*Retrieval Status Value*) où *Q* est une requête et *D* un document de la collection. Le processus d'appariement est étroitement lié au processus d'indexation et de pondération des termes. Il existe deux méthodes d'appariement :

- ***Appariement exact*** (« *exact match retrieval* ») :

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

- ***Appariement approché*** (« *best match retrieval* ») :

Le résultat est une liste de documents sensés être pertinents pour la requête. Les documents retournés sont triés selon leur score de pertinence vis-à-vis de la requête.

### I.2.3.3. La reformulation de requêtes :

La reformulation de requêtes est l'une des méthodes élaborées pour l'adaptation des SRI aux besoins des utilisateurs. C'est un processus ayant pour objectif de générer une nouvelle requête plus adéquate que celle initialement formulée par l'utilisateur. L'une des stratégies de reformulation de requêtes est celle qui est dirigée par l'utilisateur. Le principe de cette stratégie est de construire une nouvelle requête à partir de la structure des documents jugés par l'utilisateur : c'est ce que l'on appelle la réinjection de pertinence « relevance feedback » [Rochio, 71] [Harman, 92] [Boughanem, 98].

La réinjection de pertinence est un processus évolutif et interactif. Son principe fondamental est d'utiliser la requête initiale pour amorcer la recherche, puis modifier celle-ci à partir des jugements de pertinence et/ou de non-pertinence de l'utilisateur. Le but est de re-pondérer les termes de la requête initiale, ou d'y ajouter (respectivement supprimer) d'autres termes contenus dans les documents pertinents (respectivement non pertinents). La nouvelle requête obtenue à chaque itération de feedback, permet de corriger la direction de la recherche dans le sens des documents pertinents.

En effet, la simple comparaison du contenu de la requête et des documents de la base ne permet pas d'avoir tous les documents correspondant à une requête donnée. Il reste toujours des documents pertinents non restitués, car ne contenant pas les termes de la requête. La reformulation de requête permet, dans une certaine mesure, de palier ce problème.

## I.2.4. Le processus d'indexation:

### I.2.4. 1. Définition de L'indexation :

*Marie Gaëlle MONTEIL* [MONTEIL, 95] définit l'indexation comme étant l'analyse documentaire qui a pour objet de produire une représentation réduite et formalisée (l'index) des documents en y retenant l'ensemble des éléments essentiels. Ces éléments ne sont autres que des mots clés du document analysé.

L'indexation consiste donc à représenter le document par un ensemble de mots clés qui résument son contenu d'une manière intelligente, permettant ainsi de le retrouver facilement et rapidement. Les mots clés choisis pour indexer les documents sont dits *termes d'indexation*.

Les termes d'indexation doivent en théorie permettre de séparer les documents pertinents des documents non pertinents pour une requête donnée.

#### *I.2.4.1.1. Définition d'un Langage d'indexation :*

L'ensemble de tous les termes d'indexation constitue le *langage d'indexation*. Le langage d'indexation exprime le contenu sémantique des documents. Il doit offrir un compromis entre la compacité de la représentation, pour qu'elle puisse être traitée efficacement par un système informatique, et l'expressivité afin d'exprimer aussi fidèlement que possible le contenu des documents.

Le langage d'indexation peut être libre ou contrôlé :

- ***Langage d'indexation Libre*** : Ce langage est construit à partir des termes extraits des documents analysés.
- ***Langage d'indexation Contrôlé*** : Ce langage est *c*onstruit à partir d'un ensemble de termes préalablement définis et organisés généralement dans un thésaurus[2] .
  Lorsqu'un document est analysé, on ne garde dans sa représentation que les mots clés appartenant à ce thésaurus.

### I.2.4.2. Types d'indexation :

Techniquement, l'indexation peut-être manuelle, automatique ou semi-automatique [Salton, 88], [Salton et al, 88].

#### *I.2.4.2.1. Indexation manuelle :*

Dans ce type d'indexation, c'est un opérateur humain, généralement expert du domaine, qui se charge de recenser selon ses connaissances propres, les concepts dont traite un document et à les représenter à l'aide d'un langage documentaire libre ou contrôlé [maniez, 02], [lefève, 00].

L'indexation manuelle a l'avantage d'assurer une meilleure correspondance entre les documents et les termes d'indexation choisis. Ceci a pour conséquence une meilleure précision dans les documents que le SRI retourne en réponses aux requêtes des utilisateurs [Nie et al., 99].
L'inconvénient majeur de cette méthode est l'effort intellectuel qu'elle exige (en temps et en nombres de personnes). De plus, un degré de subjectivité lié au facteur humain fait que pour un même document, des termes différents peuvent être sélectionnés par des indexeurs différents. Il peut même arriver qu'une même personne, à des moments différents, indexe différemment le même document

[2]C'est un vocabulaire contrôlé et dynamique de termes, obéissant à des règles terminologiques propres et reliés entre eux par des relations sémantiques.

#### *I.2.4.2.2. L'indexation automatique :*

L'indexation automatique est un ensemble de traitements automatisés sur les documents, en vue d'en extraire les termes d'index représentatifs de leurs contenus. Cette approche est sans doute celle qui a le plus été étudiée en RI étant donnée sa faculté d'automatisation du processus d'indexation. Nous la détaillons en section suivante.

#### *I.2.4.2.3. Indexation semi-automatique :*

L'indexation automatique n'existe pas véritablement. Aucun système pour le moment n'indexe de façon totalement autonome des textes numérisés; c'est pour cela que l'on parle d'indexation « semi-automatique »ou « supervisée ». Dans ce type d'indexation, le choix final est laissé au spécialiste du domaine ou au documentaliste, qui intervient souvent pour établir des relations sémantiques entre mots-clés et choisir les termes significatifs (Synonymes, etc…) à partir de thésaurus ou d'ontologie.

En section suivante, nous nous intéressons particulièrement à l'approche d'indexation automatique, plus répandue, puisque c'est celle qui nous intéresse dans le cadre de notre travail.

### I.2.4. 3. Indexation automatique :

#### *I.2.4. 3.1. Les approches d'indexation automatique :*

Il existe deux grandes méthodes d'analyse en indexation automatique : la méthode linguistique et la méthode statistique.

**a. Les méthodes statistiques:**

Ces méthodes consistent à détecter, au moyen de divers indices numériques, des mots ou groupes de mots supposés plus «significatifs » que d'autres dans un corpus données (un exemple d'indice est le nombre d'occurrences d'un mot dans un document textuel). Ces indices sont généralement des poids attribués aux différents termes d'indexation (par exemple la fréquence d'apparition d'un terme dans un document) [Belhassen, 99], [Kammoun, 97].

L'avantage d'une indexation qui se base sur des méthodes statistiques est sa facilité de mise en œuvre. Ce type d'indexation peut être totalement automatique. Mais l'inconvénient est qu'on n'est pas sûr d'obtenir une « *bonne*» indexation en se basant sur ce genre d'informations

pour indexer le document. Par exemple, l'indice « nombre d'occurrences des mots » peut ne pas être significatif (un mot peut apparaître plusieurs fois sans qu'il soit pertinent par rapport aux sujets traités par le document).

**b. Les méthodes linguistiques:**

Ces méthodes se basent sur l'aspect syntaxique du texte contenu dans un document, afin d'en extraire des unités du langage **(**mots**,** groupes de mots…). Les méthodes linguistiques consistent à déterminer la nature des relations syntaxiques entre les termes du texte. Ces relations traduisent donc une certaine logique nécessaire à la construction du sens de certains textes. Ces méthodes font appel généralement à différents traitements linguistiques : l'analyse morphologique, la reconnaissance d'expressions idiomatiques, l'analyse lexical, 1'analyse syntaxique, l'analyse sémantique [Amirouche, 08] :

- ***analyse morphologique*** : on isole chaque terme par le biais d'un dictionnaire qui permet le contrôle des chaînes de caractères et le repérage des mots.
- ***la reconnaissance d'expressions idiomatiques :*** permet l'identification en tant qu'unités insécables de suites de mots contenant des séparateurs. Il s'agit d'expressions dont le sens ne peut se déduire simplement du sens des parties [Trigano, 94].
- ***analyse lexical*** : le traitement se traduit par la suppression des variantes combinatoires (flexion, dérivation, conjugaison) pour obtenir une forme canonique par réduction ou lemmatisation. Les outils nécessaires à ce procédé de réduction sont des dictionnaires de correspondances entre formes fléchies ou dérivées et formes canoniques (par exemple produira, produisent, ont produit etc…, auront la même forme canonique *produire*).
- ***l'analyse syntaxique*** : permet la levée des ambiguïtés grammaticales (un même mot peut avoir plusieurs fonctions syntaxiques) et l'établissement des relations de dépendance entre mots (complément de nom, sujet verbe**,** verbe complément d'objet direct**,** etc…) par filtrage morphosyntaxique**.**
- ***analyse sémantique*** : elle s'intéresse à reconnaître les sens des mots, les mots synonymes, les concepts représentatifs de ces mots, et plus généralement les relations sémantiques entre les mots notamment par l'usage de ressources terminologique comme WordNet [Amirouche, 08].

L'inconvénient majeur de cette approche est la difficulté de sa mise en œuvre par des systèmes informatique. De plus, ce type d'indexation ne peut pas être totalement automatique.

En pratique, l'indexation automatique est mise en œuvre à travers une combinaison des deux approches précédentes.

#### *I.2.4. 3.2. Mise en œuvre de l'indexation automatique :*

L'indexation automatique est fondée sur l'analyse des documents en vue de l'extraction des termes (mots-clés simples ou composés) représentatifs de leur contenu informationnel. Elle repose généralement sur les trois étapes suivantes :

**(1)** l'identification des termes d'indexation.
**(2)** la normalisation des termes d'indexation.
**(3)** la pondération des termes d'indexation.

### *Etape 1 : l'identification des termes d'indexation*

L'objectif est de trouver les mots qui représentent au mieux le contenu d'un document. Cette étape comprend :

- **L'extraction des termes d'index** :

C'est une étape qui peut sembler triviale au premier abord, et qui pourtant constituera la base de tout le reste du processus d'indexation. Il faut donc que cette phase soit d'une qualité maximale. Elle se base sur une analyse morphologique du texte à l'issue de laquelle, les mots du document sont identifiés.

- **L'élimination des mots vides :**

Les mots vides sont des mots peu significatifs et porteurs de peu de sens, augmentant ainsi la taille de l'index et rendant la recherche plus lente. De ce fait, leur élimination est impérative.
Les mots vides sont éliminés en se référant à une liste prédéfinie de mots « indésirables » (prépositions, pronoms, certains adverbes et adjectifs), dite *stoplist* (ou anti-dictionnaire). Lorsqu'on rencontre un mot dans un texte, on vérifie s'il appartient à la stoplist. Si c'est le cas le terme est éliminé de l'index, sinon on le considère comme terme d'index.

Par ailleurs, d'autres mots, non importants, sont éliminés. L'objectif est de garder seulement les termes dits importants, qui représentent au mieux le contenu du document. Pour déterminer l'importance d'un mot dans un document, trois approches ont été définies :

**1. Approche basée sur la fréquence d'occurrence** :

On prend la fréquence d'occurrences du mot dans un document comme un critère de sa représentativité dans le document. Les travaux de *Zipf* [zipf, 49], *Luhn* [Luhn, 58] font référence en la matière. « *La conjoncture de Luhn* » considère que les mots de fréquences quasi nulles et les mots à fréquences trop élevées peuvent être éliminés de l'index.

Cette conjoncture fixe pratiquement, deux seuils de fréquence (*seuil max* et *seuil min* dans la **Figure I.3**), pour éliminer les termes dont le contenu informatif est jugé faible. Seuls les termes entre ces deux seuils sont alors pertinents pour représenter les documents.

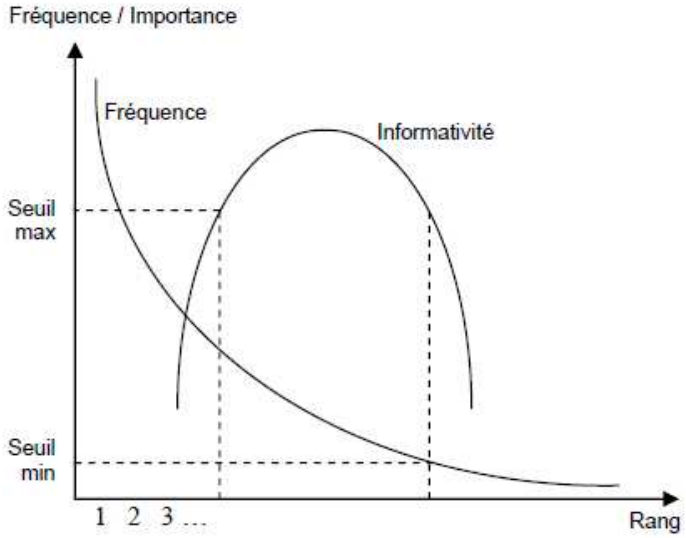


**Figure I.3 : La conjecture de Luhn.**

**2. Approche basée sur le pouvoir de discrimination d'un terme :**

On entend par « *discrimination* » le fait qu'un terme distingue bien un document des autres documents. Un terme qui a une valeur de discrimination élevée doit apparaître seulement pour un dans petit nombre de documents. Un terme qui apparaît dans tous les documents n'est pas discriminant.

L'idée de cette approche est alors de ne garder comme termes d'indexation, que les termes discriminants. Une mesure simple pour caractériser le pouvoir de discrimination d'un terme

est sa fréquence documentaire inverse, souvent notée idf *(inverted document frequency)*, et définie par :

$$idf = N/n_j$$

Où :

$n_j$ : nombre de documents contenant le terme *i*.

*N* : nombre total de documents dans la base.

**3. Approche basée sur tf*idf :**

Cette approche est généralement la plus utilisée. Elle combine les deux approches précédentes. Ainsi, l'importance d'un terme dans un document devient fonction de :

- L'importance du terme pour un document (mesurée par *tf*).
- Le pouvoir de discrimination de ce terme (mesurée par *idf*).

La représentativité d'un terme dans un document est alors calculée par la formule *tf*idf*.

Seuls les termes dont le poids tf*idf est supérieur à un seuil donné sont gardés comme termes d'indexation.

## *Etape 2 : Normalisation des termes d'indexation*

Ce traitement consiste à retrouver pour un mot sa forme normalisée (généralement le masculin pour les noms, l'infinitif pour les verbes, le masculin singulier pour les adjectifs, etc…). Ainsi, dans l'index ne sont conservées que les formes normalisées des mots, ce qui offre un gain de place appréciable, mais surtout, une recherche efficace. Le traitement associé à la normalisation repose sur deux procédures : *la lemmatisation* et *la troncature* (ou racinisation).

1. **Troncature :**

La troncature est utilisée pour éliminer les variations morphologiques ou les variantes orthographiques des mots clés de l'index. Elle consiste à couper le mot à partir d'un rang précis, afin d'obtenir son radical. Pour la langue française, la troncature à 7 caractères est adoptée.

2. **La lemmatisation** :

La lemmatisation est l'opération qui consiste à réduire les formes fléchies des mots à leur racine grammaticale. Cette technique, permet de regrouper les termes ayant des formes légèrement différentes et des sens similaires, notamment les mots conjugués.

Les approches de lemmatisation se basent sur l'élimination des terminaisons des mots, soit en examinant simplement la forme du mot, ou en se basant sur un dictionnaire.

Des expériences ont montré que la troncature et la lemmatisation améliorent significativement les performances pour les langues riches morphologiquement (ex. le français, l'italien, etc.) [Gaussier et al., 97], [Gaussier et al., 00].

## *Etape3 : Pondération des Termes :*

La pondération est l'une des fonctions fondamentales en RI. Elle est la clé de voûte de la majorité des modèles et approches de RI proposés depuis les années **1960**. L'idée derrière la pondération des termes d'indexation est d'affecter à chaque terme d'un document, un poids traduisant son importance dans le document, donc son degré d'informativité. Dans un processus d'indexation, l'élément fondamental est la technique utilisée pour pondérer les termes [Robertson et al, 97], [Singhal et al., 97], [SparkJone, 71], [SpakJones, 79]. Le poids associé à un terme peut être soit sa fréquence d'occurrence **(*tf*)** ou bien une mesure dérivant de cette fréquence (par exemple la fréquence normalisée). Cela peut être également la valeur de *tf* idf* (voir **Tableau I.1)** qui est mieux adaptée et utilisée que la première.

**Notation :**

- $tf_{ij}$ est la fréquence d'occurrences du terme $t_j$ dans le document *di*.
- $df_j$ est la fréquence documentaire du terme $t_j$**.**
- $idf_j$ est la fréquence documentaire inverse définie classiquement par :

  $Log\ (n/N_j)$ tel que $n$ est le nombre de documents de la collection et $N_j$ le nombre de documents indexés par le terme $t_j$.
- $w_{ij}$ est le poids du terme $t_j$ dans le document *di*.

- $K_1$ constante qui permet de contrôler l'influence de la fréquence du terme $t_j$ dans le document *di*. Sa valeur dépend de la longueur des documents dans la collection. Le plus souvent, sa valeur est fixée à 1,2.
- $b$ constante qui permet de contrôler l'effet de la longueur du document. Sa valeur la plus souvent utilisée est : 0,75, $dl_i$ est la longueur du document *di*.
- $\Delta l$ est la longueur moyenne des documents dans la collection entière.

| Quelques formules de pondération tf*idf | |
|---|---|
| La formule | Propriétés |
| $w_{ij} = \frac{tf_{ij}}{df_j} = tf_{ij} \times \frac{1}{df_j} = tf_{ij} \times idf_j$ [Salton et al., 73] | Cette formule est utilisée particulièrement dans des corpus de documents de tailles intermédiaires. |
| $w_{ij} = \frac{tf_{ij} * (K_1 + 1)}{K_1 \left[ (1-b) + b * \frac{dl_i}{\Delta l} \right] + tf_{ij}}$ [Robertson et al., 97] | Elle contrôle l'influence de la fréquence d'un terme dans un document. |

**Tableau I.1 : Quelques formules de pondération tf* idf.**

## I.2.5. Modèles de recherche d'information :

Le modèle joue un rôle central en RI. C'est le modèle qui détermine le comportement clé d'un système de RI. Ainsi, si c'est l'indexation qui choisit les termes pour représenter le contenu d'un document ou d'une requête, c'est au modèle de leur donner une interprétation. Étant donné un ensemble de termes pondérés issus de l'indexation, le modèle remplit les deux rôles suivants:

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

Nous présentons ici les modèles les plus couramment utilisés pour la RI, notamment le modèle Matching score, le modèle booléen, le modèle vectoriel et le modèle probabiliste.

### I.2.5.1. Modèle Matching score :

C'est peut-être le premier "modèle" utilisé dans la RI. L'idée est assez primitive et intuitive: Un document est représenté par un *ensemble* de termes pondérés par leur fréquence d'occurrences. Une requête est aussi un ensemble de termes, pondérés à 1. Le degré de correspondance est la somme des fréquences des termes de la requête dans le document:

$$R(d, q) = \sum_{i=1}^{n} f_i$$

Où $f_i$ est la fréquence d'un terme $t_i$ de $q$ dans le document $d$.

La valeur $R$ ainsi calculée est appelée la *matching score*. En réalité, cela est équivalent à parcourir le document, et à voir combien de fois les termes de la requête apparaissent dans ce document. Plus ce matching score est élevé, plus on considère que le document correspond à la requête, et donc plus il sera classé haut dans la réponse.

Ce modèle est primitif car il utilise directement le résultat de l'indexation sans aucune réorganisation ou modélisation.

### I.2.5.2. Le modèle booléen :

Dans le modèle booléen, les documents sont représentés par un ensemble de termes non pondérés. Les requêtes s'expriment à travers une expression booléenne et l'appariement ne se fait que s'il y a correspondance exacte. Le modèle booléen est largement le plus répondu et

sert de référence à tous les autres modèles. Ce modèle est basé sur la théorie des ensembles et l'algèbre de Boole [Gessler, 93].

Le modèle booléen propose la représentation d'une requête sous forme d'une équation logique. Les termes d'indexation sont reliés par des connecteurs logiques *ET*, *OU* et *NON*.

L'approche booléenne consiste à trouver les documents qui ont *exactement* les mêmes termes qu'une requête construite par mots clés. Les requêtes peuvent être affinées au moyen d'opérateurs comme *NEAR*. Ce type de recherche est à la base des moteurs de recherche comme Altavista [3]ou Google[4], ou encore le catalogue collectif RIBU [5] dont fait partie notre université.


[3] http://www.Altavista.com.

[4] http://www.Google.Com.

[5] http://60gp.ovh.net/~ribudz/

La mise en œuvre du modèle booléen est assez simple, grâce à la technique des fichiers inverses. Après avoir enregistré, pour chaque document, la liste des termes qu'il contient, on crée un fichier inversé qui dresse, pour chaque terme, la liste des documents qui le contiennent (**Figure I.4**). Cette facilité explique l'énorme succès de ce modèle.

Le modèle booléen est largement le plus répondu et sert de référence à tous les autres modèles. La majeure partie des systèmes commerciaux (catalogues en ligne, logiciels de gestion électronique de documents, serveurs bibliographiques, etc.…) utilise soit ce modèle, soit le modèle booléen hybride. Pour remédier aux inconvénients du modèle booléen, certains auteurs ont mis au point une technique dite booléenne étendue alors que d'autres proposent d'utiliser la logique floue. Salton et al. (1983) ont proposé et mis au point une technique dite booléenne étendue. Le principe est de conférer aux termes de recherche des poids et d'interpréter les opérateurs de l'équation booléenne comme des distances entre requêtes et documents.

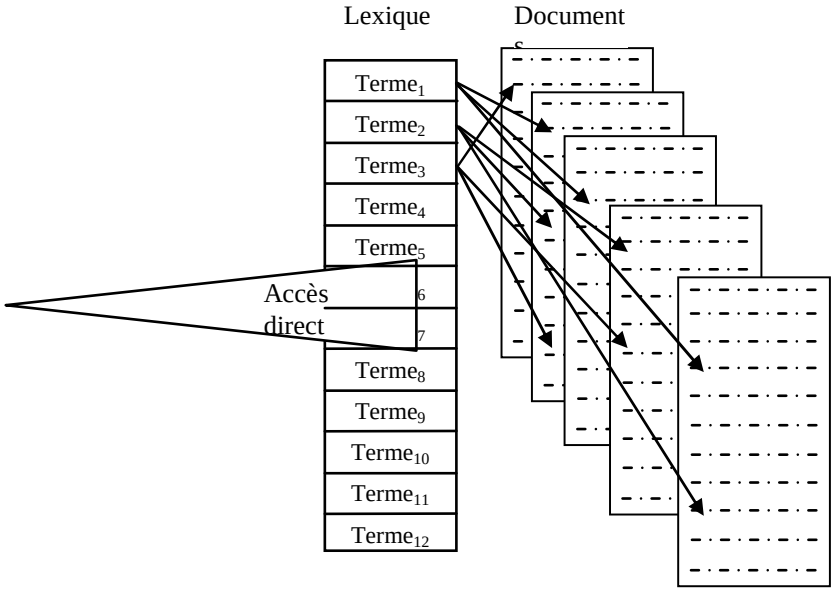


**Figure I.4: Schéma d'un fichier inverse.**

### I.2.5.3. Le modèle vectoriel :

L'idée principale du second modèle dit vectoriel est de considérer les termes d'indexation comme les dimensions d'un espace d'information multidimensionnel. Les documents et les requêtes sont alors représentés par des vecteurs dans cet espace, et la pertinence d'un document par rapport à une requête est relative aux positions respectives de ce document et de la requête dans cet espace, et est estimée par une distance définie sur cet espace. La pondération associée à un document *D* contenant un certain terme d'indexation *tj* a fait l'objet de plusieurs études, elle prend en compte en générale trois facteurs qui sont la pondération locale (correspond en général à une fonction de la fréquence d'occurrence du terme dans le document (notée *tf* pour *term frequency)*, la pondération globale (qui prend en compte l'importance de l'unité linguistique dans l'ensemble de la collection) et la normalisation en fonction de la taille de document.

Le modèle vectoriel introduit par Salton [Salton, 71], repose donc sur les bases mathématiques des espaces vectoriels. Dans ce modèle, les documents et les requêtes sont représentés dans un espace vectoriel engendré par l'ensemble des termes d'indexation $<t_1, t_2, t_3,..., t_T>$. Où *T* est le nombre total de termes issus de l'indexation de la collection des documents.

Chaque document *D* est représenté par un vecteur : $Dj = (d_{1j}, d_{2j}, \ldots, d_{ij}, \ldots, d_{Tj})$

Chaque requête est représentée par un vecteur : $Q = (q_1, q_2, \ldots, qi, \ldots, q_T)$

$d_{ij}$ **:** Poids du terme $t_i$ dans le document $D_j$

$q_i$: Poids du terme $t_i$ dans la requête *Q*.

Les termes de poids nul représentent les termes absents dans un document alors que les poids positifs représentent les termes assignés.

La fonction de calcul du coefficient de similarité entre chaque document $D_j$, représenté par le vecteur $(d_{1j}, d_{2j}, \ldots, d_{ij}, \ldots, d_{Tj})$, et la requête *Q*, représentée par le vecteur $(q_1, q_2, \ldots, q_i, \ldots, q_T)$ est appelée *Retrieval Status Value* ou *RSV*.

Ce coefficient de similarité est calculé sur la base d'une fonction qui mesure la colinéarité des vecteurs document et requête.

Différentes mesures peuvent être employées pour le calcul de la similarité: (1) le produit scalaire ; (2) la mesure de similarité de Jaccard ; (3) Le cosinus de l'angle entre les deux vecteurs :

1. **Produit scalaire :** $$RSV(Q, D_j) = \sum_{i=1}^{T} q_i * d_{ij}$$

2. **Mesure de Jaccard :** $$RSV(Q, D_j) = \frac{\sum_{i=1}^{T} q_i * d_{ij}}{\sum_{i=1}^{T} q_i^2 + \sum_{i=1}^{T} d_{ij}^2 - \sum_{i=1}^{T} q_i * d_{ij}}$$

3. **Mesure de cosinus :** $$RSV(Q, D_j) = \frac{\sum_{i=1}^{T} q_i * d_{ij}}{\left(\sum_{i=1}^{T} q_i^2\right)^{1/2} * \left(\sum_{i=1}^{T} d_{ij}^2\right)^{1/2}}$$

La similarité entre deux textes (requêtes ou documents) dépend ainsi des poids des termes coïncidant dans les deux textes. . Sur la base de cette mesure de similarité, il devient donc possible de classer les documents par ordre de pertinence décroissante

L'avantage du modèle vectoriel par rapport au modèle booléen réside particulièrement dans l'ordonnancement et le classement des documents sélectionnés selon leurs pertinences. Cependant, l'inconvénient majeur de l'approche vectorielle réside dans le fait que l'association entres les termes d'indexation n'est pas considérée. Il est impossible de représenter des phrases ou des mots multi termes. On considère en effet que les termes sont indépendants [Yates, 99]. Le modèle « Latent Semantic Indexing » pour la recherche d'information est une variante du modèle vectoriel standard et qui tente de prendre en compte, pour les représentations des documents, la structure sémantique des unités linguistiques, potentiellement implicite (i.e.Latent), représentée par leurs dépendances cachées.

### I.2.5.4. Le modèle probabiliste :

L'un des modèles qui a montré son efficacité en recherche documentaire, en particulier dans les campagnes d'évaluation TREC (Text Retrieval Conferences) est le modèle probabiliste. Le modèle probabiliste a été proposé par Robertson et Sparck Jones [Robertson, 76] en 1976. Il utilise un modèle mathématique fondé sur la théorie de la probabilité conditionnelle (appelé aussi modèle de la théorie de pertinence). Lors du processus d'indexation deux probabilités conditionnelles sont utilisées :

*P(t /pert):* La probabilité pour que le terme *t* apparaisse dans un document pertinent pour la requête.

*P(t /Nonpert):* La probabilité pour que le terme *t* apparaisse dans un document non pertinent pour la requête.

En supposant que la distribution des termes dans les documents pertinents est la même que leur distribution par rapport à la totalité des documents, et que les variables *document pertinent* et *document non pertinent* sont indépendantes, la fonction de recherche est obtenue en calculant la probabilité de pertinence d'un document *P (Pert/D).*

$$P(Pert/D) = \frac{P(D/Pert) * P(Pert)}{P(D)}$$

$$P(NonPert/D) = \frac{P(D/NonPert) * P(NonPert)}{P(D)}$$

Avec $P(D) = P(D/Pert) * P(Pert) + P(D/NonPert) * P(NonPert)$

Où :

*P (D/Pert)* (respectivement *P (D/NonPert)*) : Probabilité d'observer D sachant qu'il est pertinent (respectivement non pertinent).

*P(Pert)* (respectivement *P(NonPert))* : Probabilité à priori qu'un document soit pertinent (respectivement non pertinent).

Le coefficient de similarité requête document (*RSV*) peut être calculé par différentes formules. Robertson et Spark-Jones [Robertson, 96] proposent la formule suivante :

$$RSV = \sum \log \left( \frac{(r + 0.5) / (R - r + 0.5)}{(n - r + 0.5) / (N - n - R + r + 0.5)} \right)$$

*N*: nombre total de documents de la base,

*n*: nombre de documents contenant le terme,

*R*: nombre de documents connus comme étant pertinents,

*r*: nombre de documents connus comme étant pertinents et contenants le terme.

L'ajout de 0.5 à tous les membres s'explique par la nécessité d'écarter tous les cas limites qui entraîneraient des valeurs nulles de ces membres.

Ce modèle a donné lieu à de nombreuses extensions. Il est à l'origine du système OKAPI qui est l'un des systèmes les plus performants selon les compagnes d'évaluation **TREC**[6] [Walker, 97]. L'inconvénient majeur de ce modèle est que les calculs des probabilités sont complexes et que l'indépendance des variables n'est pas toujours vérifiée voire pas prise en compte.

Il existe d'autres modèles en RI. Un ensemble d'auteurs [Boughanem et al., 2004] tentent d'isoler au sein du processus de recherche d'information, les problèmes susceptibles d'être résolus par une approche fondée sur les réseaux de neurones ou sur les algorithmes génétiques. Le modèle offre en effet des atouts intéressants pour la représentation des relations entre termes (synonymie, voisinage, etc.), entre documents (similitude, référence, etc.) et entre termes et documents (fréquence, poids, etc.). Pour les tenants de l'approche logique en recherche d'information, un document est jugé pertinent à une requête de l'utilisateur si son contenu sémantique implique logiquement celle-ci. Ainsi [Boughanem et al, 2004] examinent en particulier les arbres sémantiques et le formalisme des graphes conceptuels en recherche d'information.

[6] http://trec.nist.gov.

## I.2.6. Évaluation de SRI :

La démarche de validation en RI se base sur l'évaluation expérimentale des performances du modèle ou du système proposé. L'évaluation des performances d'un modèle de RI, permet de paramétrer le modèle, d'estimer l'impact de chacune de ses caractéristiques et de fournir des éléments de comparaison entre modèles.

Cette évaluation peut porter sur plusieurs critères : le temps de réponse, la pertinence, la qualité et la présentation des résultats, etc. Le critère le plus important est celui qui mesure la capacité du système à satisfaire le besoin en information de l'utilisateur, c'est à dire la pertinence.

### I.2.6.1. Mesures d'évaluation :

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

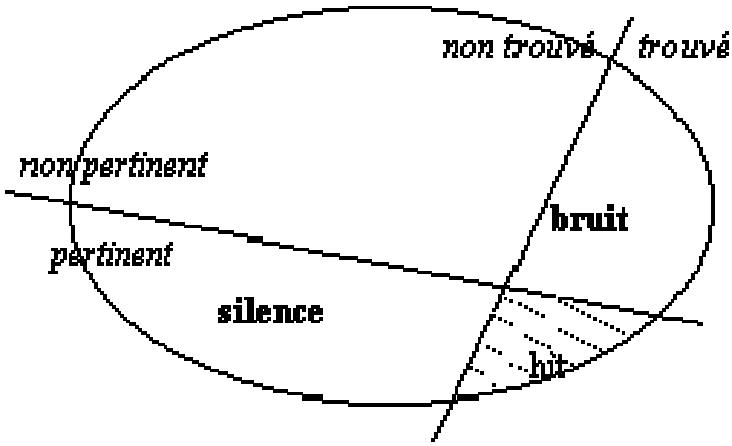


**Figure I.5 : Distribution des documents dans une collection face à une requête.**

Les taux de rappel et de précision sont les mesures les plus utilisées pour l'évaluation d'une recherche. Soient, comme illustré dans la **Figure I.5**.

$$\textbf{\textit{Taux\_rappel}} = R_d = \frac{\text{nombre de documents pertinents retrouvés}}{\text{nombre total de documents pertinents}}$$

$$\textbf{\textit{Taux\_précision}} = P_d = \frac{\text{nombre de documents pertinents retrouvés}}{\text{nombre total de documents trouvés}}$$

La précision est la proportion de documents retrouvés qui sont pertinents. Une précision égale à 1 signifie que le système n'a retrouvé que des documents pertinents.

Le rappel est la proportion de documents pertinents qui sont retrouvés. Un rappel égal à 1 signifie que tous les documents pertinents ont été retrouvés.

L'idéal serait d'avoir une précision et un rappel égaux à 1, signifiant que tous les documents pertinents sont retrouvés et qu'aucun document non pertinent n'a été retrouvé. En pratique, cet idéal n'est jamais atteint puisque ces deux quantités évoluent en sens inverse. Intuitivement, si on augmente le rappel en retrouvant plus de documents pertinents, on diminue la précision en retrouvant aussi plus de documents non pertinents. Inversement, une plus grande précision risque de rejeter des documents pertinents diminuant ainsi le rappel.

Des mesures complémentaires au rappel et à la précision, respectivement le bruit et le silence ont été définies comme suit :

$$\textbf{\textit{Silence : S}} = \frac{\text{nombre de documents pertinents non retrouvés}}{\text{nombre de documents pertinents}}$$

$$\textbf{\textit{Bruit : B}} = \frac{\text{nombre de documents non pertinents retrouvés}}{\text{nombre total de documents retrouvés}}$$

En pratique, pour mesurer le rappel et la précision d'un SRI, on s'appuie sur des collections de test. Une collection de tests est composée d'un ensemble de documents, d'un ensemble de requêtes et d'un ensemble de jugements de pertinence sur ces requêtes.

L'initiative la plus importante actuellement pour la construction de collections de tests est sans conteste TREC (*Text REtrieval Conference*) [Harman, 92].

TREC est plus qu'une collection de tests, c'est un programme d'évaluation des SRI, initié par le NIST (*National Institute of Standards and Technology*) aux USA. TREC fournit une plate-forme comportant des collections de tests, des tâches spécifiques et des protocoles d'évaluation pour chaque tâche, pour l'évaluation et la comparaison d'expérimentations sur des collections volumineuses de textes. Il faut noter que les collections TREC représentent aujourd'hui un référentiel incontournable en RI [TREC-9, 00].

#### I.2.6.2. La campagne d'évaluation TREC :

Le projet TREC consiste en une série d'évaluations annuelles des technologies pour la RI, dont l'objectif est :

- d'une part, d'offrir aux chercheurs le moyen de mesurer sur des procédures d'évaluations uniformes, l'efficacité de leurs systèmes.
- d'autre part, de leur permettre de comparer les résultats de leurs systèmes.

  Les pistes explorées par TREC sont entre autres, la recherche (ou tâche ad-hoc), le filtrage, la question-réponse, la vidéo, le web …

La tâche ad-hoc est la tâche principale dans TREC. Elle vise à évaluer les performances d'un SRI sur des ensembles statiques de documents, seuls les requêtes changent. Pour cette tâche, les participants du TREC disposent d'une collection d'environ 02 giga-octets de texte, sur un CD-ROM fourni par le NIST. Avec ces documents, le NIST procure également aux participants un ensemble de 50 requêtes en langage naturel.

Les participants testent leurs systèmes sur les documents fournis, recherchant les réponses aux requêtes données, puis classent les documents de la collection par ordre de pertinence, pour chaque requête. Les 1000 premiers documents retrouvés pour chaque requête sont soumis au NIST, chargé de l'évaluation. Le protocole d'évaluation utilisé se base sur les taux de rappel et de précision. Nous le définissons en section suivante.

#### I.2.6.3 Protocole d'évaluation TREC :

Pour chaque requête, les 1000 premiers documents restitués par le système sont examinés et des précisions sont calculées à différents points (à 5, 10, 15, 30, 100 et1000 premiers documents restitués). La précision à x (exemple précision à 5) définit le taux de documents pertinents parmi les x premiers documents retrouvés.

Une précision moyenne *MAP* est ensuite calculée pour chaque requête. Il s'agit de la moyenne des précisions de chaque document pertinent pour cette requête. La précision d'un document est la précision à *x*, tel que *x* est le rang de ce document dans l'ensemble des documents pertinents retrouvés.
Finalement, les précisions moyennes pour l'ensemble des requêtes sont calculées permettant d'obtenir une mesure de la performance globale du système.

## I.3. Conclusion :

Nous avons présenté dans ce chapitre les principaux concepts de la recherche d'information. Nous y avons développé les principales étapes d'un processus de recherche d'information, à savoir, indexation, l'appariement requête-document et la reformulation de la requête. Les principaux modèles existants dans la littérature ont été également présentés, ainsi que les différentes méthodes et cadres connus d'évaluation des performances des systèmes de recherche d'information.

La majorité des SRI présentés (basés sur les modèles vectoriels ou probabilistes) se contentent de chercher les documents qui contiennent les mêmes mots que ceux de la requête. Même si ces systèmes sont performants dans certaines situations, leurs performances se voient détériorées à cause de l'ambiguïté du langage naturel. Pour remédier à cette lacune, de nouvelles approches d'indexation basées sur les sens des mots ou « indexation sémantique » ont été développées. Dans le chapitre suivant, nous présentons un état de l'art de l'indexation sémantique.

# Chapitre II

# Indexation sémantique en RI

## II.1. Introduction :

L'indexation sémantique s'intéresse principalement à la représentation des documents et requêtes par les sens des mots qu'ils contiennent plutôt que par les mots eux-mêmes [Amirouche, 08]. L'objectif sous-jacent est d'améliorer la représentation des entités indexées et de pallier aux problèmes de l'indexation classique basée mots.

L'objectif du présent chapitre est de présenter les principales approches d'indexation sémantique. En section II.2, nous présentons la problématique de l'indexation classique basée mots_clés. Le reste du chapitre est dédié à la présentation des approches d'indexation sémantique. Ainsi, l'approche d'indexation conceptuelle est décrite en section II.3. La section II.4 est dédiée à la présentation des approches d'indexation sémantique basées sur la désambiguïsation. Tout d'abord, un aperçu des méthodes de désambiguïsation est présenté en paragraphe II.4.1, puis les approches d'indexation sémantique en paragraphe II.4.2.Ces approches sont basées soit sur la désambiguïsation basée sur les ressources externes, ou sur la désambiguïsation basée sur les corpus. Les premières sont présentées en paragraphe II.4.2.3, les secondes en paragraphe II.4.2.4.

## II.2. Problématique :

En indexation classique, les documents et requêtes sont représentées par des mots clés issus de leurs contenus. L'utilisation des mots pour représenter le contenu des documents et requêtes pose deux problèmes, *l'ambiguïté des mots* et leur *disparité* [Amirouche, 08].

*L'ambiguïté des mots* se rapporte à des mots lexicalement identiques et portants des sens différents, elle se manifeste sous trois formes [Gillon, 04] :

- l'ambiguïté lexicale polysémique,
- l'ambiguïté lexicale homonymique,
- l'ambiguïté structurale non lexicale (l'ambiguïté syntaxique).

Le problème d'ambiguïté implique que des documents non pertinents, contenant les mêmes mots de la requête sont retrouvés. Par exemple, dans une recherche sur Google[7] (voir la **Figure II.1**) a l'aide du mot clé *jaguar (animal)* nous remarquons que des références contenant le mot *jaguar* ont été retrouvées mais ne corresponde pas tout à fait à ce que l'on cherche.

[7]http:\\www.Google.com


**Figure II.1 : Présentation des résultats de recherche du mot *jaguar* sur Google.**

*La disparité des mots* (word mismatch) se réfère a des mots lexicalement différents mais portant un même sens. Ceci implique que des documents, pourtant pertinents, ne partagent pas de mots avec la requête, ne sont pas retrouvés.

Plusieurs solutions spécifiques on été proposer pour pallier aux problèmes cité auparavant à travers différents travaux, tel que l'utilisation *d'expression ou de mot composé* pour résoudre le problème lier à l'ambiguïté des mots mais aussi *l'expansion de requêtes* pour réduire la disparité des mots [Salton et al., 83].

L'utilisation d'une solution globale permettant de répondre au problème de l'ambiguïté et la disparité des mots été nécessaire d'où *l'indexation sémantique*.

L'indexation sémantique s'intéresse à deux principaux points : d'abord retrouver le sens correct de chaque mot dans le document (respectivement de la requête), ensuite représenter ce document (respectivement cette requête) [Amirouche, 08].

Comme solution au premier point portant sur l'identification du sens des mots, l'indexation sémantique s'appuie sur des techniques dites de désambiguïsation des mots ou WSD (Word Sense Disambiguation) que nous présentons en section II.4.

En réponse au second point portant sur la représentation sémantique des documents et requêtes, deux principales approches de représentation existent:

- ***La représentation basée sur les sens*** : Un terme d'indexation est alors représenté uniquement par son sens.
- ***la représentation combinée mots-clés/sens*** (la représentation Mixte): Un terme d'indexation est alors représenté par le couple (mot-clé, sens associé).

Notons enfin, qu'il existe une approche autre que l'indexation sémantique, exploitant la sémantique des textes dans la représentation de l'information est c'est *l'indexation conceptuelle.*

## II.3. L'indexation conceptuelle :

Les concepts véhiculant également un sens, l'indexation conceptuelle peut être considérée comme une extension de l'indexation sémantique. Les techniques employées diffèrent entre ces deux modes d'indexation. L'indexation conceptuelle utilise *systématiquement* des ressources sémantiques externes afin de déterminer le sens des mots à traiter.

### II.3.1. Présentation des ressources sémantiques externes :

#### II.3.1.1. Les concepts en RI :

Selon le dictionnaire de l'académie française, *Le concept regroupe les objets qu'il définit en une même catégorie appelée « classe »*. De façon générale, le terme concept est souvent utilisé comme se référant à toute notion, de l'idée au lexème, en passant par l'entité et la catégorie.

En RI, les concepts sont le plus souvent prédéfinis et préordonnés dans des structures conceptuelles telles les hiérarchies de concepts ou les ontologies. Un concept, qui correspond à un nœud d'une structure conceptuelle peut alors différer d'une structure à une autre.

La notion de concept est assez difficile à cerner vu l'insuffisance de ressources conceptuelles générales, alors dans certains cas, on ne se limite pas au nœud de l'ontologie utilisée (exemple : WordNet) mais on considère la région entourant ce nœud (nœuds et arcs).

**II.3.1.2 Les ontologies :**

Gruber [Gruber, 93] définit une ontologie comme suit «*une ontologie est une spécification explicite d'une conceptualisation*».

Le mot *ontologie* est généralement utilisé pour renvoyer à des structures lexicales et sémantiques variées. Par exemple : les lexiques comme WordNet.

***II.3.1.2 .1 Wordnet :***

WordNet[8] est un réseau lexical électronique qui couvre la majorité des noms, verbes, adjectifs et adverbes de la langue Anglaise et qu'elle structure en un réseau de nœuds et de liens.

- Les nœuds sont constitués par des ensembles de termes synonymes appelés *synsets*.
  - Un synset représente un concept.
  - Un concept est une entité sémantique, lexicalement représentée par un terme.
  - Un terme peut être un mot simple ou une collocation (mot composé).
- Les liens représentent des relations sémantiques entre concepts.

Un exemple de hiérarchie de synsets correspondant au nom « human» est donné dans la **Figure. II.2.**

[8] http://wordnet.princeton.edu/

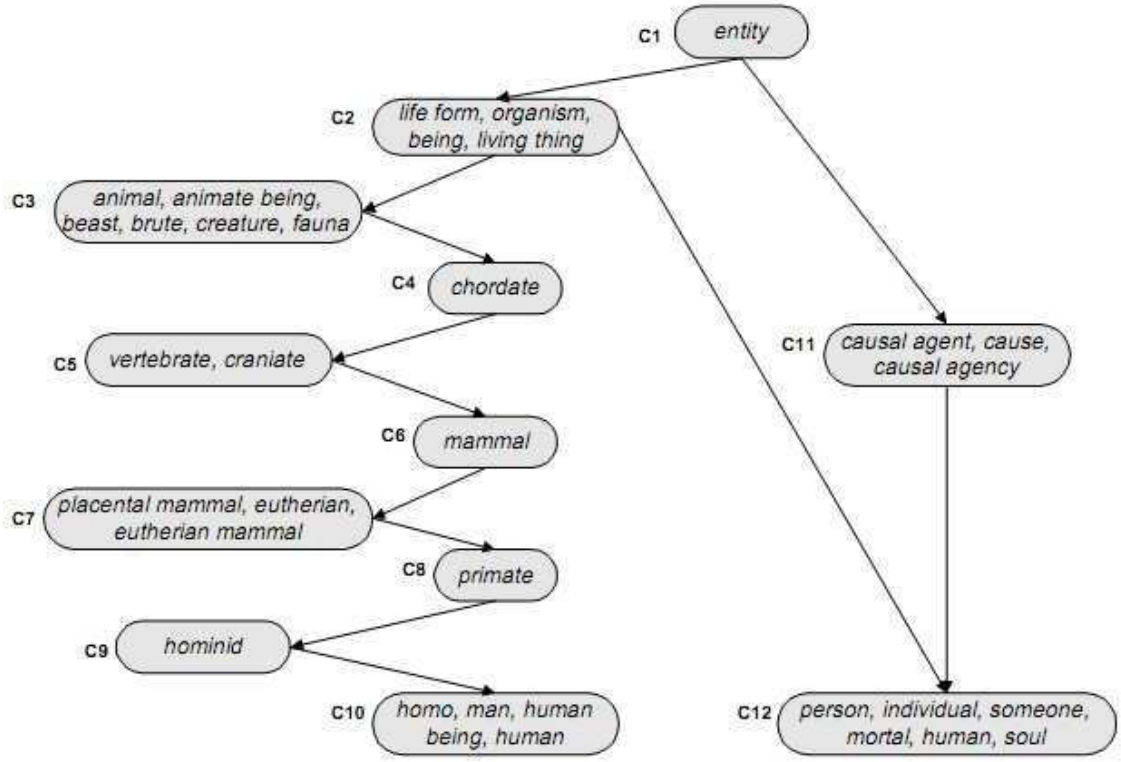


**Figure.II.2 : Hiérarchie WordNet pour le terme human.**

La liste qui suit énumère les relations sémantiques disponibles dans Wordnet.

- ***Synonymie*** : relation liant deux concepts équivalents ou voisins. Il s'agit d'une relation symétrique.
- ***Antonymie*** : relation liant deux concepts opposés. Cette relation est symétrique.
- ***Hyperonymie*** : relation liant un concept-1 à un concept-2 plus général.

- ***Hyponymie*** : relation liant un concept-1 à un concept-2 plus spécifique. C'est la réciproque de l'hyperonymie. Cette relation peut-être utile en RI. En effet, si l'on cherche tous les textes traitant de véhicules, il peut être intéressant de retrouver ceux qui parlent de voitures ou de motos.
- ***Méronymie*** : relation liant un concept-1 à un concept-2 qui est une de ses parties, un de ses membres ou une substance le constituant.
- ***Métonymie*** : relation liant un concept-1 à un concept-2 dont il est une des parties. C'est la relation inverse de la méronymie.

- ***Implication*** : relation liant un concept-1 à un concept-2 qui en découle.
- ***Causalité*** : relation liant un concept-1 à son effet.
- ***Valeur*** : relation liant un concept-1 (adjectif) qui est un état possible pour un concept-2.
- ***A pour valeur*** : relation liant un concept-1 à ses valeurs (adjectifs) possibles. C'est la relation inverse de Valeur.
- ***Voir aussi*** : relation entre des concepts ayant une certaine affinité.
- ***Similaire à*** : certains concepts adjectifs dont le sens est proche sont regroupés. Un synset est alors désigné comme étant central au regroupement. La relation Similaire à lie un synset périphérique au synset central.
- ***Dérivé de*** : indique une dérivation morphologique entre le concept cible (adjectif) et le concept origine.

### II.3.1.3 Les réseaux sémantiques en RI :

Désigne un ensemble de mots représentant des objets ou concepts, reliés entre eux en fonction de critères sémantiques particuliers :

Par exemple, pour exprimer qu'un *chat est un espèce de la classe des animaux*, on crée un réseau sémantique formé des mots *chat* et *animal* qui seront liés par une relation nommée *sorte de* (le réseau pourrait alors avoir cette forme : ('chat' **--***sorte de* **-->** 'animal').

Ainsi, dans sa plus simple expression, un réseau sémantique est constitué de deux mots (dits *nœuds*) qui sont liés par une relation orientée (dit *arc*) dont la signification relève de la connaissance du monde. Autrement dit, il s'agit d'un graphe orienté, formé de nœuds reliés par des arcs qui expriment les relations sémantiques entre les nœuds.

## II.3.2. Définition de l'indexation conceptuelle selon Wood :

Wood [Wood, 97] propose une autre définition de l'indexation conceptuelle. Elle combine les techniques de représentation de connaissances et de traitement du langage naturel avec les techniques classiques pour indexer des documents telles que l'utilisation de lemmatisation ou de mesures extraites de l'approche statistique, afin de permettre au système d'établir des connections entre la terminologie d'une requête et celle de l'information recherchée.

L'indexation conceptuelle permet également au système d'indexation de bénéficier de la structure conceptuelle des phrases dans les résultats de l'indexation. Ceci étant réalisé en superposant des structures conceptuelles sur les phrases, afin de discerner comment ses éléments sont agencés pour générer un sens. Ainsi, il est possible d'organiser automatiquement les mots et phrases en une *taxonomie conceptuelle* qui lie chacun de ces concepts à son concept générique le plus proche, ainsi qu'aux concepts de sens approchés.

La hiérarchie ainsi générée comporte des relations de *most general subsumees* (les noeuds fils de la taxonomie) et de *most specific subsumers* (leurs parents), formant ainsi une organisation basée sur la généricité des termes. Deux étapes peuvent être distinguées: l'extraction de concept, et ensuite, leur structuration en hiérarchie. Ici, la méthode décrite par Wood [Wood, 97] se base sur WordNet afin de déterminer les concepts.

## II.4. L'indexation sémantique basée sur la désambiguïsation :

Même si la logique voudrait que la désambigüisation des mots (WSD) permette d'accroître les performances de certains systèmes de recherche d'information (RI), il n'en demeure pas moins qu'il a été nécessaire d'étudier l'impact de l'ambiguïté sur la RI et l'opportunité d'introduire les techniques de désambiguïsation en indexation des documents. L'un des premiers travaux fondamentaux quant à l'évaluation de l'apport de connaissances sémantiques dans un système de recherche d'information est rapporté par Krovetz et Croft [Krovetz et Croft, 92].

Leurs expériences sont effectuées à l'aide des collections de test CACM [9]et TIME.[10] L'étude manuelle des réponses fournies par le système aux requêtes leur a permis de considérer l'impact de l'ambiguïté sémantique sur un système de RI en comparant les différences entre des textes pertinents et des textes non pertinents. Ils constatent que, pour les premiers les différences entre le sens des termes communs entre la requête et le texte sont plus nombreuses que pour les deuxièmes. Ce résultat est logique puisque si le sens avec lequel un mot est utilisé dans un document n'est pas le même que celui de la requête, ce document risque de ne pas être pertinent par rapport à ce mot. Cette constatation est en faveur de l'utilité d'une désambiguïsation du sens.

Ils concluent que les différences entre les mots-sens de la requête et les mots-sens du texte rapporté sont plus fréquentes lorsqu'il y a moins de termes en commun entre la requête et le texte. La désambiguïsation sémantique apporterait donc une amélioration aux systèmes de recherche d'information pour récupérer des textes ayant peu de mots en commun avec la requête ou lorsque la requête est de petite taille (1 ou 2 mots). Mais, d'après d'autres expériences, l'impact d'une telle désambiguïsation est relativement faible puisque l'amélioration apportée par une désambiguïsation parfaite ne serait que de 2 %. Cela s'explique par le fait que la présence d'autres termes de la requête dans le texte permet d'effectuer une sorte de désambiguïsation implicite du sens des mots.

---

[9] http:\\www.dcs.gla.ac.uk/idom/ir_resources/test_collections/cacm/

[10] https://isserver11.princeton.edu/

Suite aux travaux de Krovetz et Croft, Sanderson [Sanderson, 94a, 96] a effectué des expériences afin d'évaluer l'impact de la polysémie sur la RI et donc l'utilité d'un système de désambiguïsation sémantique. Il utilise les textes de la collection Reuters [Sanderson, 1994b]. Pour le premier aspect, ses expériences semblent montrer que la polysémie ne pose pas réellement de problème à un système de RI. Si des termes d'une requête sont polysémiques, la cooccurrence de chacun des termes dans un document fait que l'ambiguïté est, le plus souvent, levée, ce qui rejoint les conclusions de [Krovetz & Croft, 1992]. Enfin, il considère que, pour qu'un système de désambiguïsation sémantique puisse apporter une amélioration à un système de RI, il faut que ses performances dépassent les 90 % de réussite. En effet, son système de RI est très sensible aux erreurs de désambiguïsation. Les travaux de Sanderson de sont très souvent cités pour montrer que l'utilisation d'une désambiguïsation sémantique dégrade les résultats

Par la suite, Gonzalo et ses collègues [Gonzalo & al., 98] on proposé une nouvelle méthode qui consiste à indexer directement les synsets Plutôt que d'indexer des lemmes. Le but de cette méthode est d'augmenter à la fois la précision et le rappel. En effet, lors d'un enrichissement de la requête, les termes utilisés pour l'enrichissement sont bien souvent eux-mêmes polysémiques. Si les termes de la requête et ceux des textes sont remplacés par un concept (sous la forme d'un synset), ce problème de la polysémie des termes d'enrichissement est écarté. L'expérience montre une augmentation des performances de 14 % (en passant de 48 % à 62 %). Ce résultat est très encourageant mais il convient de le relativiser par la nature même de l'expérience.

A partir de ses différentes expériences, l'indexation par le sens des mots(ou indexation sémantique) a été pressentie comme un moyen qui permettrait d'améliorer les performances de la recherche. Pour retrouver les sens corrects des mots dans un document, l'indexation sémantique a recours aux techniques de désambiguïsation des sens des mots.

Avant de décrire les approches d'indexation par les sens des mots, nous présentons d'abord les principes fondateurs des approches de désambiguïsation puis les travaux les plus significatifs dans le domaine.

## II.4.1. Les approches de désambiguïsation des sens des mots (WSD) :

Les approches cherchant à résoudre le problème de l'ambiguïté du sens des mots sont nombreuses et variées. Elles peuvent être classées en plusieurs groupes de la manière qui suit :

- ***Les approches d'intelligence artificielle*** *:* ce type d'approche vise à modéliser la compréhension du langage humain ; nous y trouvons indifféremment des *approches symboliques* et des *approches connexionnistes*.

- ***Les Approches utilisant des bases de connaissances informatisées (approches exogènes)*** *:* ces approches s'appuient généralement sur des bases de connaissances existantes, comme des lexiques tels que ldoce[11] ou WordNet, des thésaurus ou d'autres bases de connaissances.

- ***Les Approches basées sur le corpus (approches endogènes)*** : ces méthodes sont généralement de type statistique et utilisent de gros corpus de texte ; deux types d'approches sont à distinguer, celles utilisant des corpus étiquetés dans la phase d'apprentissage et celles s'affranchissant de cette limitation.

- ***Les Approches Mixtes*** : il existe aussi des approches faisant intervenir simultanément plusieurs de ces techniques.

### II.4.1.1. Premiers pas en désambiguïsation lexicale :

C'est dans le domaine de la traduction automatique que nous trouvons les premières recherches pour résoudre de manière automatique le problème de l'ambiguïté lexicale des mots [Weaver, 49], [Kaplan, 55], [Reifler, 1955], [Richens, 58], [Masterman, 61], [Quillian, 61], [Quillian, 69]. Ils ont constaté très tôt que l'étude du contexte de la cible constituait la principale information qui permet d'en sélectionner le sens adéquat tel qu'introduit dans le Memorandum de Weaver [Weaver, 49] mais aussi dans les expériences de Kaplan [Kaplan, 55] ou il montre l'importance du choix de la taille du contexte.

[11] http://www.ldoceonline.com/

En 1955, Reifler [Reifler, 55] montre l'importance prépondérante des éléments du contexte syntaxiquement liés à la cible pour effectuer ce choix. Il propose alors la notion de *coïncidences sémantiques* entre un mot et son contexte, comme facteur principal en désambiguïsation.

Pour résoudre les problèmes d'ambiguïtés sémantiques, les besoins en représentation des connaissances se sont fait ressentir très tôt. Ainsi, de nombreuses recherches tentent de créer un langage intermédiaire basé sur des principes logiques et mathématiques. Parmi ses recherches, celles de Richens et Masterman [Richens, 58] [Masterman, 61] conduisent à la notion de *réseau sémantique*.

En 1949, Weaver précise déjà que des études statistiques sont nécessaires comme première étape pour la désambiguïsation. Suite à ces considérations, certains auteurs cherchent à établir une approche basée sur l'analyse statistique du langage [Richards, 53], [Pimsleur, 57].

D'une manière surprenante, les premiers travaux menés en désambiguïsation sémantique ont rapidement soulevé les problèmes fondamentaux et proposé une grande variété de solutions tout à fait représentatives de ce qui se fait actuellement dans le domaine.

### II.4.1.2. Les approches d'intelligence artificielle :

Les approches d'intelligence artificielle des années 60-80 plaçaient la désambiguïsation sémantique dans un contexte plus large de la compréhension du langage humain. Le fonctionnement des systèmes dédiés à cette tâche était basé sur une modélisation des connaissances de nature sémantique et syntaxique. [Ide et Veronis, 98] particularisent deux types de méthodes d'intelligence artificielle caractérisant cette période: les méthodes *symboliques* et les méthodes *connexionnistes.*

Les méthodes dites *symboliques* s'appuient sur la *représentation symbolique du sens des mots* par l'intermédiaire de réseaux sémantiques. Certains chercheurs ont construit des systèmes permettant de choisir le sens correct d'un mot en calculant le plus court chemin entre les nœuds d'un réseau de concepts. D'autres approches ont tenté de résoudre le même problème en utilisant des réseaux sémantiques enrichis par des informations sur les rôles, les relations et les contraintes gouvernant la combinaison des mots dans une phrase. On proposait aussi des

modules de raisonnement permettant de trouver dans une ontologie les ancêtres communs des mots co-occurrent dans le même contexte, idée qui anticipait le concept de *"similarité sémantique"* développé dans les années 90 par exemple par Resnik [Resnik, 95].

Les méthodes *connexionnistes* regroupaient les approches basées sur le modèle des réseaux d'activation *(spreading activation network),* terme utilisé pour désigner un réseau dont les nœuds, activés par un certain contexte, produisent l'activation des nœuds connexes. On retrouve ici également le schéma des réseaux neuronaux capables d'apprendre à partir d'une collection d'exemples préalablement désambiguïsés.

### II.4.1.3. Approches utilisant des bases de connaissances informatisées (approches exogènes) :

Les approches exogènes se basent sur l'exploitation du contexte et des définitions issues de ressources linguistiques externes telles que les dictionnaires informatisés ou MRD (*Machine Readable Dictionnary*), [Lesk,86][Veronis et al.,90] [Ide et al.,90] [Wilks et al., 90] [Guthrie et al.,91],les thésaurus [Yarowsky,92],les ontologies[Sussna,93][Resnik,93a] [Resnik, 93b] [Resnik,95] ou une combinaison d'entre elles[Agirre et al.,01].

Le principe général de ces approches consiste à utiliser le dictionnaire, ou l'équivalent, comme référence. La désambiguïsation d'un mot polysémique dans un contexte donnée, par exemple une phrase, consiste dans un premier temps à extraire les mots cooccurrents présents dans cette phrase. Parallèlement, pour chaque sens du mot polysémique, on extrait du dictionnaire la liste des mots présents dans chacune des définitions correspondantes. Ensuite, la désambiguïsation consiste à choisir parmi les sens possibles, celui dont la définition possède la liste de mots la plus proche de la liste extraite de la phrase. Nous allons voir que les modèles ont depuis été largement optimisés, mais le principe reste le même.

#### *II.4.1.3.1. Les approches basées sur les dictionnaires informatisés :*

La première tentative d'utilisation d'un dictionnaire informatisé ou MDR pour la désambiguïsation était par Lesk [Lesk, 86]. Le principe de base de cette méthode est de mesurer le chevauchement entre les différentes définitions, dans le dictionnaire électronique

*Oxford Advanced Learner's Dictionary of Current English (OALD)*, d'un mot ambigu et les définitions de ses voisins immédiats. Son utilisation du dictionnaire est illustrée dans la **Figure II.3**, où on cherche à résoudre le sens du mot *Ash* dans la phrase *There was ash from the coal fire*. La désambiguïsation du mot *Ach* prend le sens *Ach'* grâce à l'apparition du mot *burnt* dans les définitions des sens *Ach'* et *coal'*.

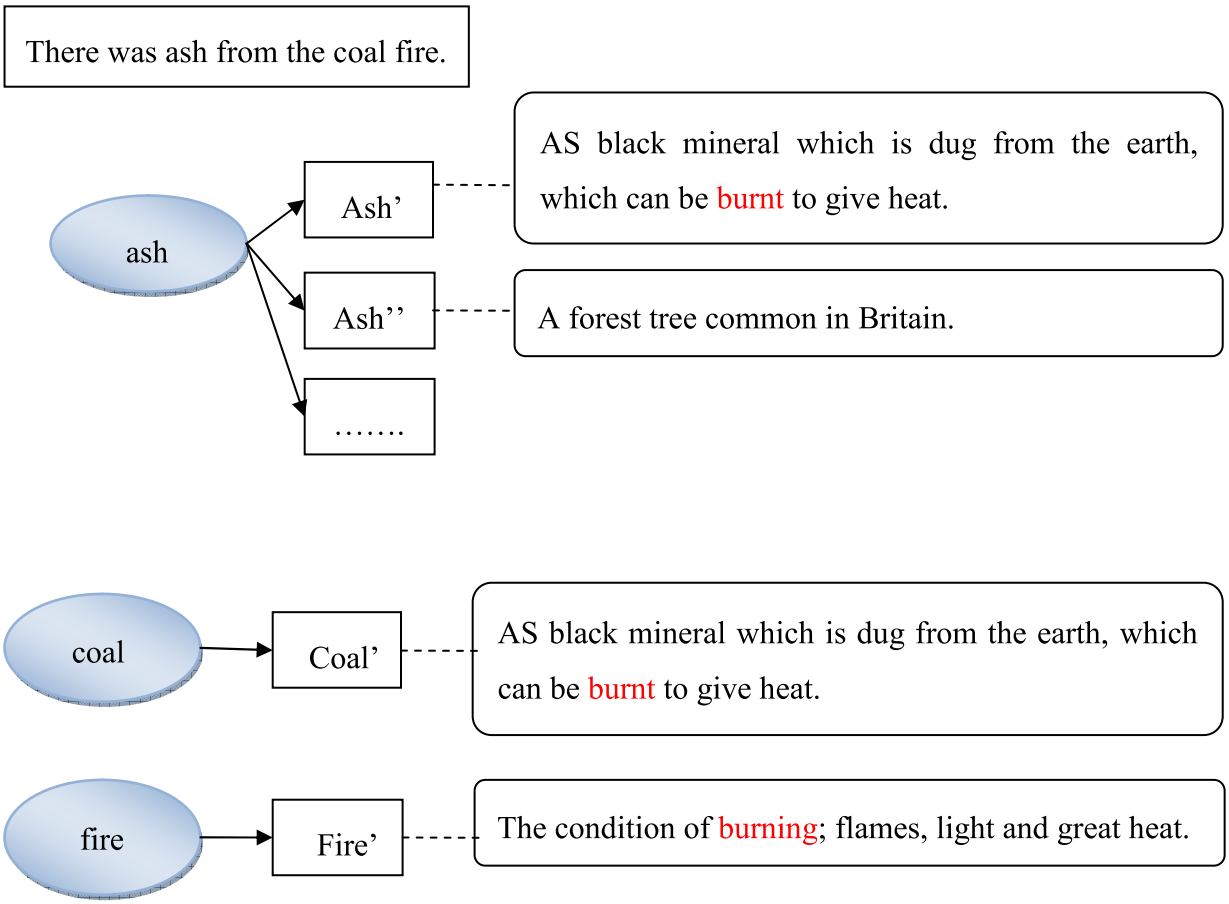


**Figure II.3: exemple d'utilisation des définitions des dictionnaires.**

Cette méthode permet de trouver le bon sens dans 50% à 70% des cas, en utilisant une palette de sens assez fine. Cependant, elle présente l'inconvénient (cité dans [Sanderson, 97], [Amirouche ,08]) d'être très sensible aux mots qui se trouvent dans chaque définition: la présence ou l'absence d'un mot donné peut radicalement changer le résultat. La méthode de Lesk sert tout de même de base pour la plupart des travaux subséquents, en désambiguïsation, utilisant des dictionnaires informatisés.

Pour résoudre le problème des définitions courtes posées par Lesk, Wilks et al [Wilks et al., 90] ont utilisé l'approche de désambiguïsation de Lesk avec le dictionnaire LDOCE (*Longman Dictionary of Contemporary English*) [Longman, 88], dont les définitions ont été manuellement étendues. La technique d'expansion utilisée consiste à enrichir toutes les définitions du dictionnaire LDOCE avec les mots qui co-occurrent généralement avec les textes de ces définitions. Cette information de co-occurrence a été dérivée de toutes les définitions du dictionnaire.

Wilks a examiné la précision de son désambiguïseur sur le mot *bank* qui apparaît dans environs 200 phrases du dictionnaire LDOCE. Pour évaluer son désambiguïseur, Wilks a d'abord désambiguïsé manuellement ces phrases. Puis les résultats de la désambiguïsation automatique sont comparés à ceux de la désambiguïsation manuelle. Wilks a rapporté que son système retrouvait le sens correct du mot *bank* dans plus de 50% des cas.

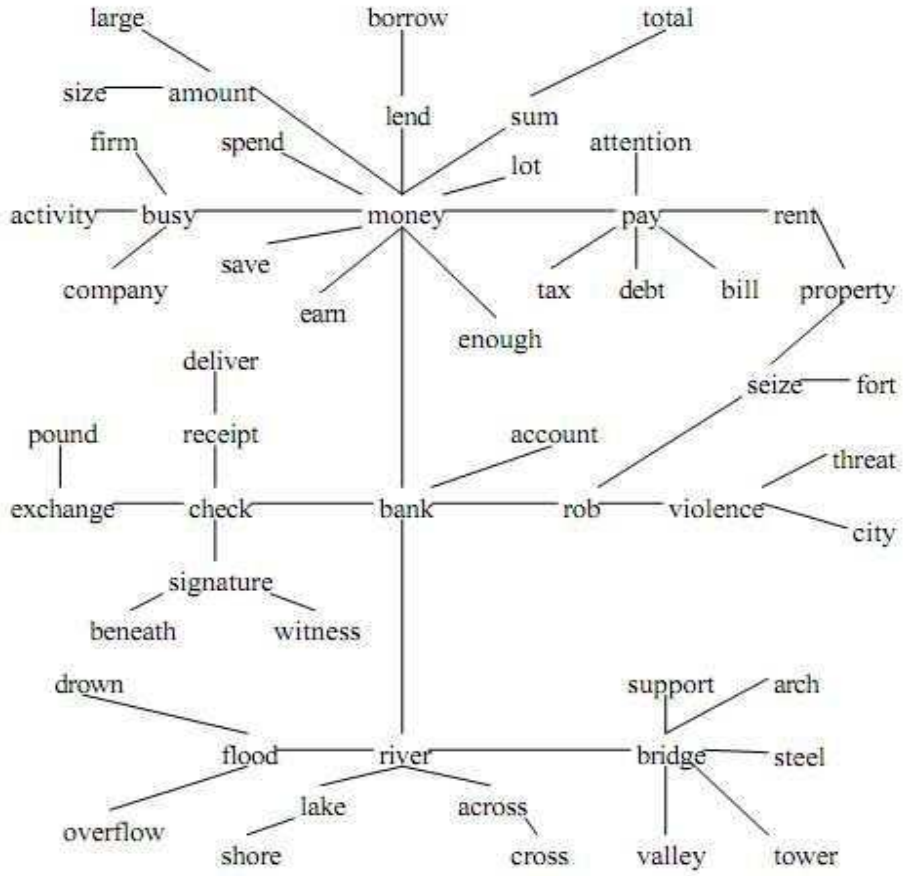


**Figure II.4 : La co-occurrence des mots dans LDOCE**

Cowie et al. [Cowie, Guthrie et Guthrie 92] On tenté d'amélioré l'approche de Lesk on essayant de déterminer la combinaison optimale des sens des mots dans une phrase, en comptant le nombre de mots communs entre les définitions de tous les sens d'une combinaison, à un moment donné. La méthode d'optimisation qu'ils appliquent est connue sous le nom de *simulated annealing*. Pedersen et Banerjee [Banerjee, Pedersen ,2002] on eux aussi proposé une autre variante de la méthode de Lesk, on utilisant les définitions de sens *(glosses)* de *WordNet*.

En raison du fait que les dictionnaires sont créés pour l'usage humain, et non pour les ordinateurs, il ya quelques inconsistances [Veronis and Ide, 1991] [Ide and Veronis ,1993a] [Ide and Veronis ,1993b]. En effet, si les dictionnaires contiennent des informations détaillées au niveau lexical, ils manquent cruellement d'informations pragmatiques utilisées pour la détermination du sens. Par exemple, les liens entre *ash* et *tobacco*, *cigarette* ou *tray*, sont très indirectes dans le dictionnaire alors que ces trois mots sont fréquemment en cooccurrence avec le mot *ash* dans les corpus.

#### *II.4.1.3.2. Les approches basées sur un thésaurus :*

Les thésaurus (*thesauri* en anglais) fournissent des informations sur les relations entre les mots et plus particulièrement les synonymes. Le thésaurus international Roget (Roget's International Thesaurus) est informatisé dans les années 1950, il est le plus fréquemment utilisé dans WSD.

L'approche de Yarowsky [Yarowsky, 1992], se base sur l'encyclopédie Grolier multimédia [Grolier] ainsi que 1042 catégories sémantiques[12] dans lesquelles tous les mots du thésaurus Roget [Kirkpatrick, 88] sont placés, elle consiste en deux étapes : la première consiste à assigner une catégorie (parmi les 1024 citées ci-dessus) au mot à désambiguïser, la seconde consiste à assigner le sens correct à l'occurrence de ce mot dans la catégorie ainsi déterminée.

[12] Il s'agit de larges catégories couvrant des domaines comme, les machines/outils ou les insectes/animaux

Pour décider à quelle catégorie sémantique une occurrence de mot ambigu doit être assignée, un ensemble de mots indices (ou mots déterminants selon la terminologie de [Ricart, 06]), est construit pour chaque catégorie sémantique, en utilisant l'encyclopédie Grolier.

Pour désambiguïser un mot dans une catégorie donnée, on examine son contexte. Si un mot déterminant apparaît dans ce contexte, le mot ambigu appartient probablement à la catégorie du mot déterminant.

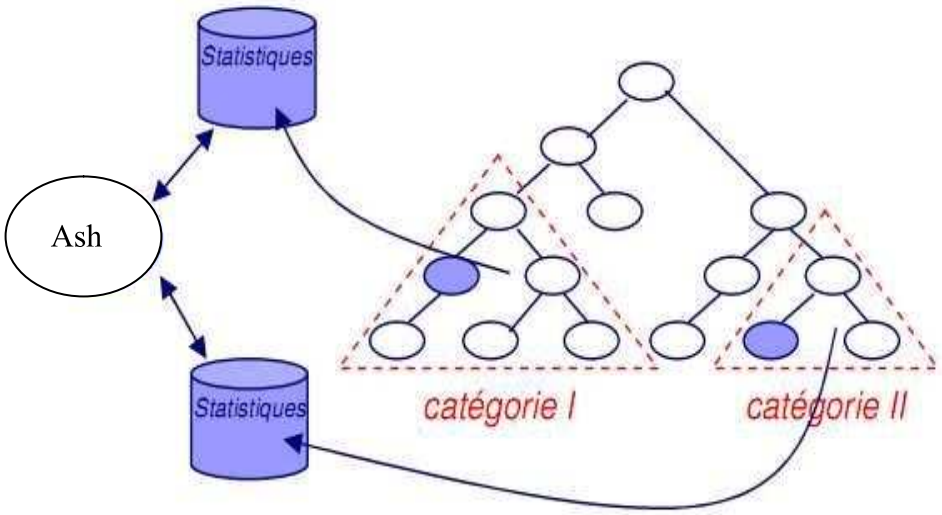


**Figure II.5 : schéma de l'algorithme proposé par Yarowsky.**

Dans ses expériences Yarowsky a appliqué son désambiguïseur sur 12 mots ambigus, la précision atteinte est de 92% avec une moyenne de trois sens par mot. Yarowsky observe que cette méthode permet essentiellement d'extraire les informations sur le thème et qu'elle est donc essentiellement efficace pour les noms.

Le problème des thésaurus est semblable à celui des dictionnaires informatisés (MDR) c'est-à-dire qu'ils manquent de cohérence et sont, avant tout, destinés à être utilisés par des humains.

#### *II.4.1.3.3. Les approches basées sur un lexique :*

Au milieu des années 1980, de nombreux efforts se portent sur la réalisation manuelle de grandes bases de connaissances informatiques appelées *lexiques informatiques* (Computational lexicons en anglais), par exemple WordNet [Miller, Beckwith, Fellbaum, Gross & Miller, 90] [Fellbaum, 98], cyc [Lenat & Guha, 89], acquilex [Briscoe, 91], comlex [Grishman, MacLeod & Meyers, 94, 99], etc. Cependant, Wordnet reste le lexique informatique le plus populaire utilisé dans plusieurs travaux de recherche [Sussna, 1993], [Resnik, 1995], [Voorhees, 93].

Voorhees [Voorhees, 93] ayant observé que la polysémie des mots influe négativement sur la précision, et que la synonymie altère le rappel, a proposé une procédure d'indexation automatique en utilisant WordNet. Elle est partie du principe qu'un groupe de mots utilisés dans un certain contexte a un sens plus précis (non ambigu) même si les mots le constituant sont isolément ambigus. Ainsi, elle a proposé une technique d'expansion des requêtes avec les sens. Pour ce faire, elle a utilisé les synsets correspondant aux noms dans WordNet et les relations hiérarchiques (is-a).

Pour désambiguïser une occurrence d'un mot ambigu, les synsets de ce mot sont classés selon une valeur de cooccurrence calculée entre le contexte de ce mot et un voisinage contenant les mots du synset dans la hiérarchie de WordNet.

Voorhees pour définir le voisinage du sens d'un mot contenu dans un synset introduise un nouveau concept appelé *hood*. En considérant l'ensemble des synsets et les relations d'Hyperonymie et d'Hyponymie dans WordNet comme les sommets et les arcs dirigés d'un graphe, le hood d'un synset S est défini comme le plus large sous-graphe connecté connexe :

1. qui contient S,
2. qui contient seulement les descendants d'un ancêtre de S et
3. qui ne contient aucun synset qui ait un descendant qui inclut une autre instance (une autre forme) d'un membre de S

Par exemple, à partir du fragment de la structure de WordNet donnée par la **Figure II.6**, le voisinage du premier sens de *house* inclurait les termes: *housing*, *lodging, apartment, flat, cabin, gatehouse, bungalow, cottage.*

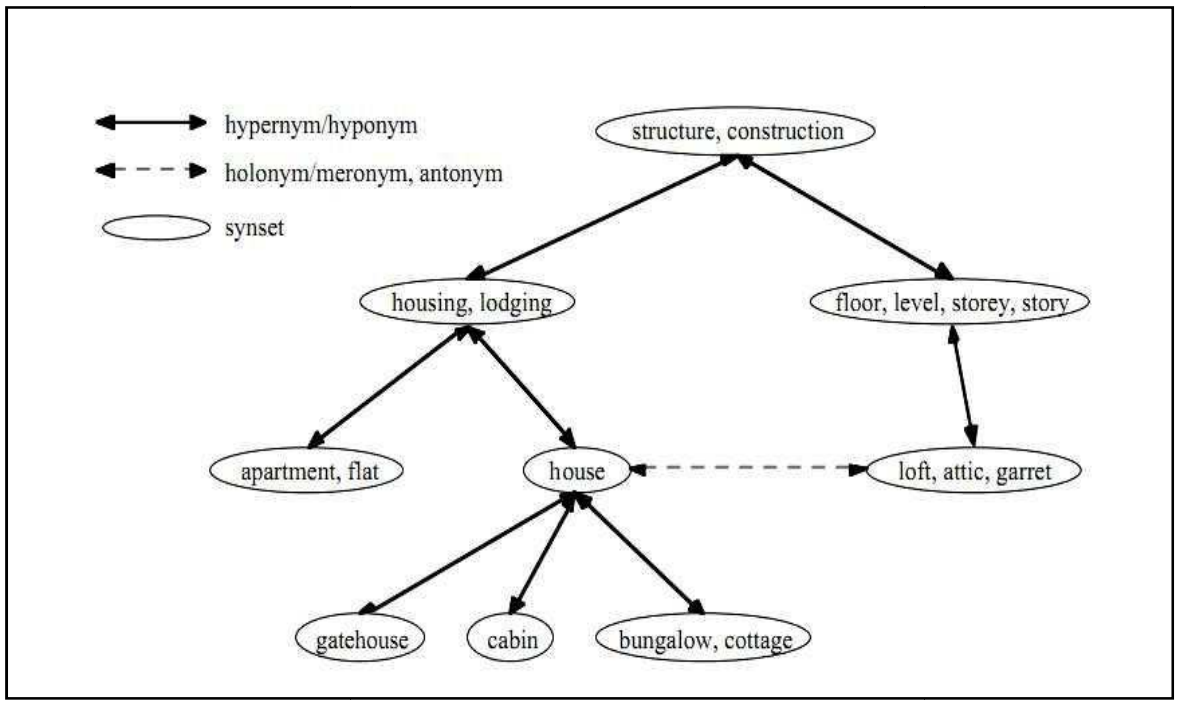


**Figure II.6 : Exemple de hood (voisinage de mot) selon Voorhees**

Les termes structure et *construction* (situés en haut de la hiérarchie), ne seraient pas inclus puisque un des descendants de leur synset contient un autre sens du terme *house*.

Voorhees a soumis cette approche pour expérimentation sur une collection de test désambiguïsée manuellement (les requêtes et les documents) et l'a comparée aux performances du même système et la même collection non désambiguïsée. Elle a constaté une dégradation dans la performance de son système pour la plupart des requêtes. Elle suppose que l'exploitation des deux relations de WordNet n'est pas suffisante pour retrouver les sens corrects des noms. Elle suggère que l'amélioration de la méthode d'indexation requiert une amélioration de la méthode d'identification des concepts importants dans le texte et une désambiguïsation correcte.

Les travaux de Sussna [Sussna, 93] viennent élargir les travaux de Voorhees [Voorhees, 93], basé sur Wordnet et se distingue par l'utilisation de relations dans WordNet différentes à la hiérarchie is-a. Il utilise Wordnet dont le but est de calculer la distance sémantique entre deux mots. Des poids pour les différentes relations de chaque nom sont définis dans le réseau sémantique. La force du poids assigné reflète la *similarité sémantique* exprimée par la relation. Par exemple, pour les relations de Synonymie le poids le plus fort leurs a été assigné, alors que pour les relations d'antonymie les poids les plus faire leurs a été affecté. La distance sémantique entre deux synsets a été calculée en additionnant les poids attachée aux relations qui constituent le plus court chemin entre ces deux synsets.

La méthode de désambiguïsation de Sussna [sussna, 93] est la même que la méthode utilisé par Voorhees [Voorhees, 93]: étant donné un mot ambigu apparaissant dans un certain contexte, tous les synsets (sens) contenant ce mot ont été recherchés dans WordNet.

Chaque synset a donné un score calculé comme la somme des distances sémantiques entre les mots du contexte et le synset. Le score a été utilisé pour classer les synsets, avec celle du haut étant choisie comme le sens du mot ambigu.

Sussna essayé sa technique désambiguïsation dans un certain nombre de configurations. Les principaux paramètres qui variés étaient la taille du contexte utilisé et le nombre de mot a désambiguïsé simultanément.

Les tests ont été effectués sur dix documents tirés de la collection Time, Dans ces documents 319 mot ambigu été sélectionnés et manuellement désambiguïsé par Sussna. Cette méthode a obtenu d'assez bons résultats (56% de désambiguïsations correctes relativement à 78% atteint par des humains).

En 1995, Resnik propose un système [Resnik, 95] qui réalise la WSD de noms avec l'aide de WordNet. A partir d'une liste de noms, il construit toutes les paires possibles et pour chacune d'elles, l'algorithme de WSD calcule la similarité entre les deux, basée sur la hiérarchie de WordNet.

La similarité entre deux mots est fonction du niveau du premier concept partagé dans la hiérarchie. Plus, ce concept est bas dans l'arbre des sens, plus la similarité est importante. A

partir de ces similarités, la désambiguïsation se fait en essayant de maximiser la similarité entre les noms d'entrée.

Par exemple, dans la **Figure II.7** nous pouvons voir que les formes A et B, avec les sens A', A'' et B', B''.

*Supposons les formes :* *et les sens comme :*

A = doctor et B = nurse, A' = « la profession dans la santé »,

A'' = « personne avec un Ph.D »,

B' = «la profession dans la santé »

B'' = « nourrice »

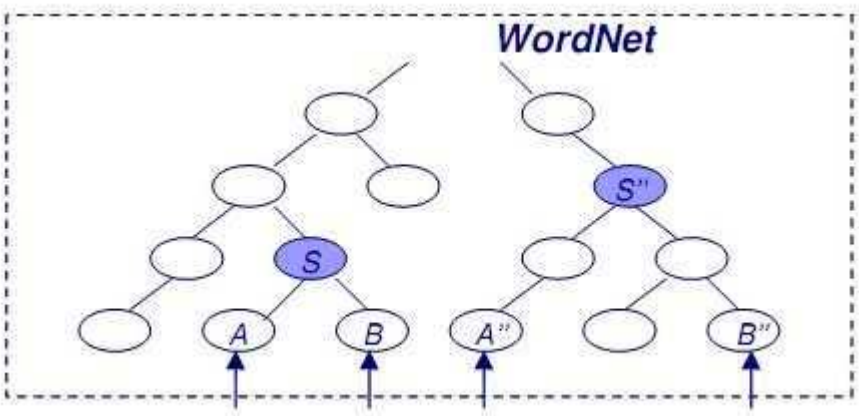


**Figure II.7 : schéma de la similarité en utilisant WordNet.**

Dans la hiérarchie de WordNet, le sens partagé de « *health professional* » est plus concret que celui de « *person, individual* ». A' et B' seront donc les sens choisis.

Resnik a utilisé dans ses expérimentations une collection de textes tirés du *Brown Corpus of American English* [Francis and Kučera, 82]. Les résultats finaux ont été prometteurs, proches des jugements humains.

### II.4.1.4. Approches basées sur le corpus (approches endogènes) :

A côté du développement des dictionnaires, thésaurus et lexiques, l'évolution des systèmes informatiques des années 80 a encouragé la création et le stockage des corpus de textes de grande taille et le retour de l'étude des mots aux méthodes *empiriques (statistiques) basées sur le corpus,* avancées dès les années 30-40. A partir de ces prémisses, le domaine de la désambiguïsation sémantique a connu deux orientations principales.

L'une regroupe les *approches supervisées* [Weiss, 73;Kelly et al., 75] qui utilisent des corpus d'entraînement annotés (étiquetés), comportant des étiquettes de sens, pour désambiguïser les nouvelles occurrences des mots polysémiques, en faisant appel à des hypothèses de type théorie de l'information.

L'autre, regroupant les *approches non-supervisées* [Schütze, 92;98], essaye de dériver les informations nécessaires à la désambiguïsation à partir des corpus non-annotés (non étiquetés), par des méthodes de classification des sens ou *clustering*.

#### *II.4.1.4.1 Approches basées sur les corpus étiquetés :*

Weiss [Weiss, 1973] est l'un des premiers à s'intéresser à l'apprentissage automatique de règles de désambiguïsation à partir de corpus étiquetés. En examinant 5 occurrences d'un mot ambigu dans un corpus d'une vingtaine de phrases, Weiss a manuellement construit un ensemble de règles permettant la désambiguïsation. Ces règles étaient de deux types :

- règles générales de contexte
- règles de modèle.

Une règle générale de contexte déduisait qu'un mot ambigu avait un certain sens si un mot particulier apparaissait près de ce mot ambigu. Par exemple, si le mot *type* était près du mot *print*, le sens du premier mot était très probablement celui d'un petit bloc de métal avec un caractère sur une extrémité.

Une règle de modèle déduisait qu'une occurrence d'un mot avait un certain sens si un mot particulier apparaissait dans un endroit spécifique relatif à l'occurrence (un terme appelé collocation en anglais et coïncidence sémantique dans ce document). Par exemple, si *of*

apparaissait juste après le mot type, le sens de cette occurrence était susceptible de signifier une subdivision d'un genre particulier de chose.

Weiss a constaté que les règles de modèle étaient meilleures à déterminer le sens que les règles de contexte. L'exactitude de son désambiguïseur était de l'ordre de 90% mais le nombre de tests était très réduit.

Les chercheurs Kelly et Stone [Kelly&Stone, 75] créèrent un désambiguïseur basé sur un ensemble de règles construites manuellement pour six mille mots. En outre, quelques-unes de ces règles vérifiaient certains aspects grammaticaux des occurrences des mots. La catégorie grammaticale d'un mot peut en effet être un fort indicateur de son sens, comme dans le cas de *the train* et *to train*.

Ces règles de grammaire et de contexte ont été groupées de sorte que seulement certaines règles soient appliquées dans certaines situations. Ce système a été conçu pour traiter une phrase entière en même temps, et il était capable de changer l'ordre dans lequel les mots d'une phrase se désambiguïsaient en arrêtant la désambiguïsation d'un mot, en essayant de traiter des autres mots, et puis en revenant au mot original pour vérifier si la désambiguïsation pouvait alors être accomplie.

Cependant, cette stratégie n'a pas eu de bons résultats et les auteurs ont constaté qu'à une plus grande échelle le système échouait.

Une autre approche de désambiguïsation a été tentée par Black [Black, 88], il a extrait des arbres de décision sémantique d'un corpus de 22 millions de mots dont il avait étiqueté environ 2 000 occurrences de cinq lexèmes.

Cependant, cette approche se trouve confrontée à deux problèmes majeurs. Le premier problème est, comme nous l'avons déjà mentionné, le manque de corpus lexicalement étiquetés. En raison de ce manque, la plupart des études sont menées sur un nombre limité de mots, voire sur un seul. Une méthode permettant de contourner ce problème est présenté dans la section qui suit. Le second problème ne concerne pas que les corpus étiquetés et est probablement le plus préoccupant. Il réside dans la dispersion des données.

#### *II.4.1.4.2 Approches basées sur les corpus non étiquetés :*

Pour pallier le problème de la rareté des corpus lexicalement étiquetés, des recherches sont menées dans le but de s'affranchir de cet étiquetage [Pereira, Tishby & Lee, 93]; [Schütze, 92, 98]. Dans ce type d'approche, la notion de sens est généralement directement induite du corpus.

Ainsi, Schütze [Schütze, 92,98] propose une méthode basée sur le modèle des espaces de vecteurs utilisé en recherche d'informations [Salton, Wong & Yang, 75]. Dans cette approche, chaque mot est représenté par un vecteur dans un espace de grande dimension. Ces vecteurs sont automatiquement regroupés en paquets en fonction de leur degré de similitude. Chaque paquet est supposé être représentatif d'un sens. Ces méthodes possèdent certains avantages mais véhiculent également un inconvénient majeur : les sens ne correspondent à aucun ensemble de sens bien défini.

Pour pallier ce problème, Pedersen et Bruce [Pedersen et Bruce, 97a] proposent une technique permettant de faire correspondre ces groupes de sens aux sens d'un lexique donné. Cependant, les résultats obtenus ne sont pas encourageants et moins bons qu'en affectant le sens le plus fréquent à chaque occurrence.

### II.4.1.5. L'approche Mixte :

Une autre tendance actuelle dans le domaine de la désambiguïsation automatique, signalée par Audibert [Audibert, 06], est la conception des *systèmes Mixtes* qui combinent plusieurs sources d'informations (fréquence des mots, informations d'ordre morphologique, sémantique, contextuel) et types de méthodes (extracteur de collocations et de définitions, étiqueteur syntaxique, analyseur des traits sémantiques etc.). Des expériences récentes ont montré la validité de ce type d'approche pour la désambiguïsation automatique

## II.4.2 Les approches d'indexation sémantique :

L'indexation sémantique s'intéresse à la représentation des documents et requêtes par les sens des mots qu'ils contiennent. Les sens des mots sont retrouvés par application d'une méthode de désambiguïsation. Selon la méthode de désambiguïsation utilisé (présentées dans les sections précédentes) nous pouvons distinguées deux types d'approche d'indexation sémantique [Amirouche, 08] :

- ***Les approches d'indexation sémantique basée sur la désambiguïsation endogène :***

Dans ce cas, des corpus d'apprentissage sont d'abord utilisés pour construire la connaissance nécessaire à la désambiguïsation. Les mots d'index sont ensuite identifiés dans la collection à indexer, puis désambiguïsés. Finalement, les textes de la collection son indexés en utilisant les sens ainsi retrouvés.

- ***Les approches d'indexation sémantique basée sur la désambiguïsation exogène :***

Dans ses approches la connaissance nécessaire à la désambiguïsation n'est plus apprise à partir d'un corpus, mais est extraite de la ressource linguistique externe utilisée. Formellement, cette connaissance se traduit par des scores associés aux différents sens d'un mot, sur la base de [Amirouche, 08] :

- La distance sémantique de ce sens aux différents sens associés aux autres termes dans le document (contexte global).

- Degré de recouvrement entre d'une part, le contexte local de ce mot et d'autre part le voisinage [Voorhees, 93] de ce sens ou la définition de ce sens (ensemble de synonymes) [Katz et al., 98] dans la ressource linguistique utilisée.

La plupart des approches d'indexation sémantique basées sur la désambiguïsation exogène, s'appuient en général sur des ontologies pour déterminer les différents sens du mot mais aussi pour désambiguïser les sens des mots. Le principe de base de l'indexation consiste alors à extraire dans un premier temps, l'ensemble des termes descripteurs du document. Il s'agit ici d'une indexation classique. Ces termes sont ensuite désambiguïsés. Pour ce faire, les sens de chaque terme d'indexation sont d'abord retrouvés à partir de la ressource externe. Puis, des

scores sont associés aux différents sens ainsi retrouvés. Le sens qui maximise le score est alors retenu comme sens adéquat du terme d'indexation correspondant. Une fois les termes d'indexation désambiguïsés, la représentation des textes indexés se fait soit à partir des seuls sens (ou concepts) identifiés lors de l'étape de désambiguïsation, soit à partir d'une combinaison des mots-clés et sens corrects associés.

## II.5. Conclusion :

Dans ce chapitre, nous avons présenté l'état de l'art de l'indexation sémantique en RI. Nous avons mis l'accent sur son intérêt à résoudre le problème d'ambigüité et de disparité des mots en RI. Comme dans toute discipline informatique, toute méthode novatrice est évaluée empiriquement. Ce qui nous a amené à consacrer une grande partie de ce chapitre pour la présentation de quelques travaux connus dédiés à l'indexation sémantique en RI.

De cette étude, il en découle que beaucoup d'efforts ont été investis en exploitant les ressources disponibles afin d'atteindre une recherche sémantique d'information à proprement parler. Cette dernière a été initialement basée sur la désambiguïsation. Elle est connue comme une tâche difficile pour un être humain, en particulier, quand la collection de document est de grande taille. Avec l'insuffisance de ressources sémantiques, la désambiguïsation est aussi difficile pour une machine. Ce qui justifie en partie l'hétérogénéité des résultats des travaux présentés dans ce chapitre.

Bien que les résultats des travaux étudiés divergent sur certains points, ils convergent tous vers une même conclusion, à savoir : la nécessité d'une désambiguïsation. Quand elle est bien réalisée, elle contribue de façon significative à l'amélioration des performances des SRI. Elle dépend aussi de la richesse de la ressource sémantique utilisée ainsi que de la taille de la collection des documents ciblée.

Pour notre part, étant très convaincus de l'intérêt de l'indexation sémantique en RI, il nous est venu à l'esprit de définir une nouvelle approche d'indexation sémantique et de l'intégrer dans une plateforme de RI qui est *Terrier*. Le chapitre suivante est dédiée à la présentation de notre contribution à la définition d'une nouvelle approche d'indexation sémantique et de son intégration dans Terrier.

# Chapitre III

# SemTerrier : *Vers l'intégration de la sémantique dans Terrier*

## III.1. Introduction :

Nous avons présenté dans le **chapitre. II** les différents travaux liés à l'indexation sémantique. Notre travail s'inscrit dans ce contexte et vise deux objectifs : (1) implémenter un module d'indexation sémantique (l'approche sous jacente a été proposée dans le cadre d'un travail de magister en cours que nous avons étendu par quelques propositions dont l'étiquetage syntaxique des textes) et (2) intégrer ce module à la plate forme de RI Terrier. Dans ce chapitre, nous décrirons d'une part l'approche d'indexation sémantique implémentée et d'autre part, l'architecture et le fonctionnement du nouveau système *SemTerrier* résultant de l'intégration du module d'indexation sémantique dans Terrier.

Le système terrier est complexe et son fonctionnement interne difficile à appréhender. Cette difficulté tient du nombre de classes interconnectées utilisées dans son implémentation. Et bien que ce soit un bon candidat pour d'éventuelles extensions, l'intégration de nouveaux modules dans Terrier n'est pas chose évidente. Les principales questions auxquelles nous nous sommes confrontées sont :

1) Comment Terrier indexe-t-il les documents ?

2) Comment Terrier effectue-t-il la recherche ?

3) Comment intégrer un nouveau module d'indexation dans Terrier ?

Le chapitre est organisé de manière à réponde graduellement à ces questions : ainsi, avant de décrire l'architecture de S*emTerrier*, nous aborderons le fonctionnement du système d'indexation de Terrier, puis nous décrirons les différentes étapes vers l'intégration du module d'indexation sémantique dans Terrier. Ces éléments sont abordés en section III.3. La section II.4 présente l'approche d'indexation sémantique que nous allons implémenter.

Mais avant d'aborder la description détaillée de notre contribution, nous présentons ci-après différentes terminologies et notations qui seront utilisées dans la suite de ce chapitre.

## III.2. Terminologie et notations :

- Un concept réfère à un sens particulier d'un mot donné.
- On appelle mot orphelin [Amirouche, 11] un mot qui n'a pas d'entrée dans Wordnet.
- On appelle mot simple [Amirouche, 11] un mot qui a une entrée dans Wordnet.
- Les collocations sont généralement considérées comme une combinaison récurrente de mots qui se trouvent ensemble plus souvent que par le simple fait du hasard et qui correspondent à une utilisation arbitraire
- Nous appelons termes complexes les différentes collocations de WordNet.
- L'ensemble $L_{complexe}$ correspond à un ensemble qui contient tous les termes complexes (collocation) d'un document donné.
- L'ensemble $L_{simples}$ correspond à un ensemble qui contient tous les mots simples d'un document.
- L'ensemble $L_{orphelins}$ contient tous les mots orphelins d'un document.

# III.3. Présentation de Terrier :

Terrier[13], *TeRabyte RetrIEveR* a été développé par le département informatique de l'université de *Glasgow*. C'est un logiciel open source entièrement écrit en java. Il est utilisé avec succès pour la recherche Ad-hoc, la recherche web et la recherche inter-langages dans des environnements centralisés et distribués. C'est une plate forme qui est destinée à l'indexation de volumes importants de documents en plein-texte: jusqu'à 25 millions de documents. Selon l'étude comparative sur les SRI en open sources effectuée par Ricardo Baeza-Yates[14], Terrier fait partie des trois meilleurs systèmes dans un environnement Java. Les deux autres sont MG4J et Lucene.

Terrier offre plusieurs modèles de pondération de documents et d'expansion de requêtes basé sur le Framework DFR (*Divergence From Randomness*)[15].Comme tous les moteurs de recherche Terrier possède les principales facettes suivantes :

***Indexation*** : Permet l'extraction des termes des différents documents du corpus (basic indexed unit).

***Recherche*** : Permet de générer des résultats aux requêtes formulées par les utilisateurs.

## III.3.1. Le processus d'indexation de Terrier :

Pour l'indexation d'une collection de documents, Terrier se base sur l'indexation classique à 4 étapes (voir **Figure III.1**) :

- ***Splitter la collection de document*** : consiste à parcourir l'ensemble du corpus et d'envoyer chaque document à l'étape suivante.

- ***L'extraction des termes*** (Tokenize Document) : qui consiste à parser chaque document reçu et en extraire les différents termes.

13 http://www.terrier.org

14 http://wrg.upf.edu/WRG/dctos/Middleton-Baeza.pdf

15 http://terrier.org/docs/v2.2.1/dfr_description.html

- ***Traitements des termes extraits*** : et qui consiste en l'élimination des mots vides et la lemmatisation des termes.

- ***Construction de l'index***

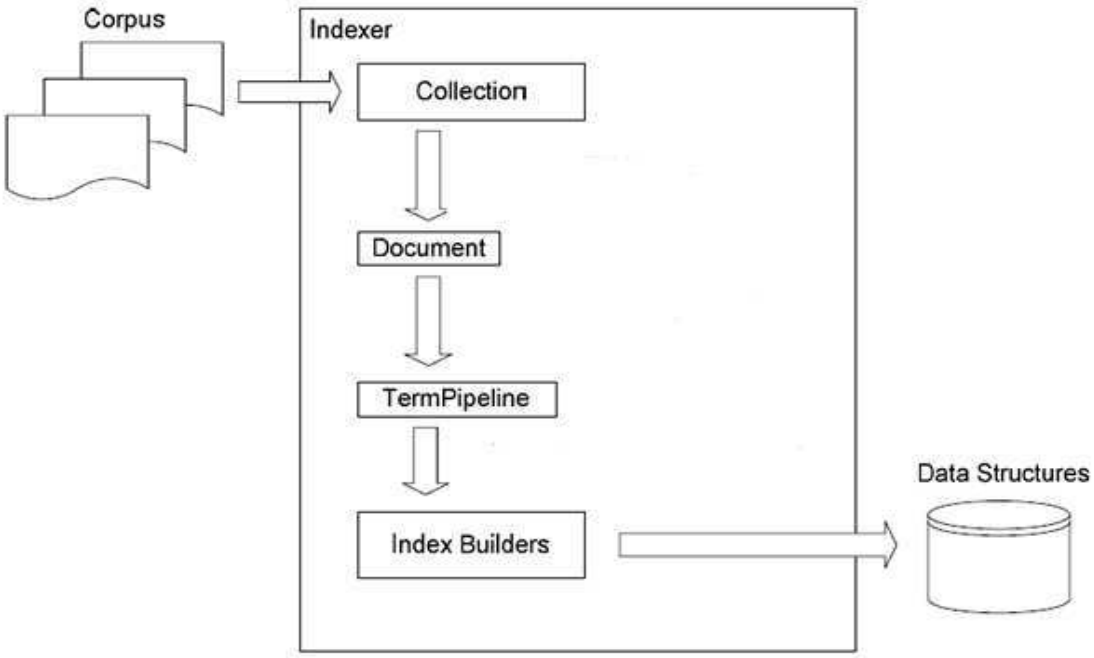


**Figure III.1 : Présentation du processus d'indexation de Terrier**.

Le résultat de l'indexation consiste en un index constitué des structures de données suivantes : Inverted File, lexicon File, Direct Index, Document Index (présentés en détail en Annexe 2).

Les Modules Collection, Document, TermPipeline, Index Builders font partie de L'API d'indexation de Terrier. Ils sont présentés en détail dans l'annexe 2.

Dans notre proposition SemTerrier, nous avons utilisé cette même architecture que nous avons étendue par l'adjonction d'un module d'indexation sémantique.

### III.3.2. Le processus de recherche de Terrier :

Un des principaux objectifs de Terrier est de faciliter la recherche d'information. Terrier implémente pour cela un certain nombre de fonctionnalités de recherche qui offre un large choix pour le développement de nouvelles applications et pour les tests en RI. En effet, Terrier offre un grand choix de modèles de pondération[16], il propose aussi un langage de requête avancée[17]. Une autre fonction de recherche très importante intégrée dans Terrier est l'automatisation de l'expansion de requête. La **Figure.III.2** présente un aperçu du processus de recherche dans Terrier.

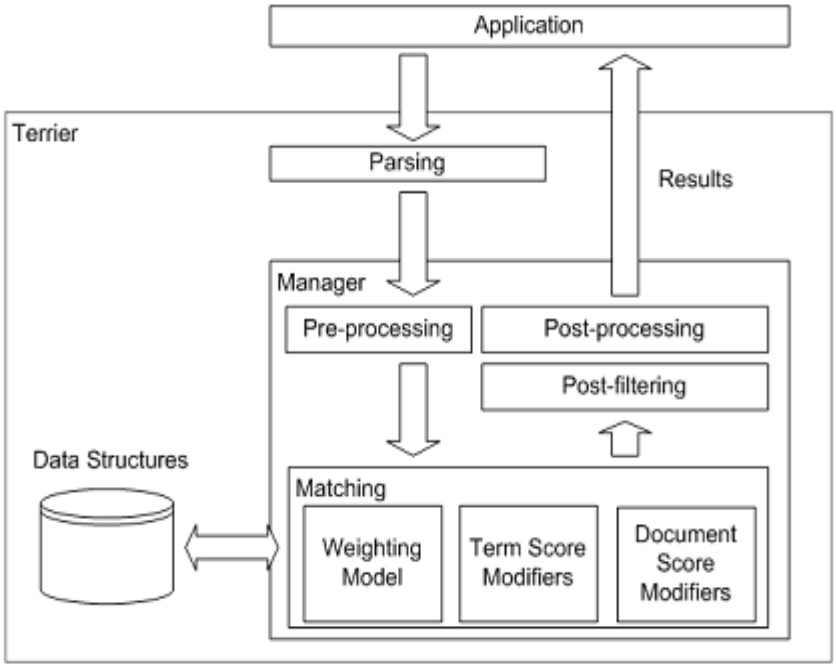


**Figure III.2: Présentation du processus de recherche de Terrier [Ounis, 10].**

Le processus de recherche de Terrier est constitué d'un ensemble de modules : Query, Manager, Matching (présentés en détail dans l'annexe 2).

Pour notre par le module qui nous intéresse est l'objet *Weighting Model* qui se trouve dans le module matching. Il nous permet de définir notre modèle de pondération.

[16] http://terrier.org/docs/v2.2.1/configure_retrieval.html
[17] http://terrier.org/docs/v2.2.1/querylanguage.html

## III.4. Présentation de l'approche d'indexation sémantique implémentée dans SemTerrier :

L'approche d'indexation sémantique implémentée dans SemTerrier est résumée à travers la figure III.3.L'approche d'indexation s'articule sur trois étapes principales qui sont: (1) l'Identification des termes d'index, (2) La désambiguïsation des termes et (3) la pondération des concepts.

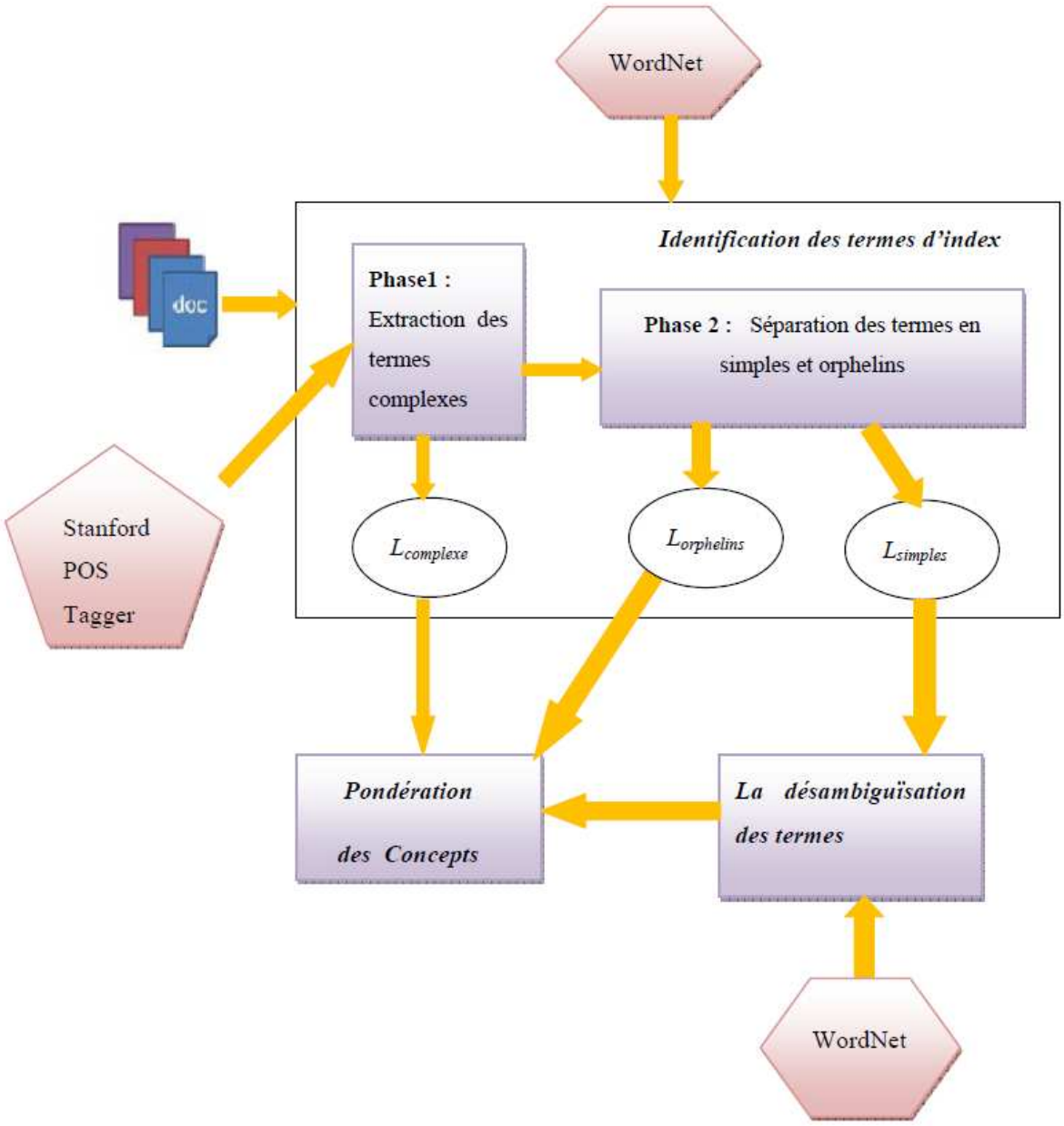


**Figure III.3 : Vue d'ensemble de l'approche d'indexation sémantique de SemTerrier.**

## III.4.1. Identification des termes d'index :

Cette étape a pour objectif d'identifier les termes (simples ou composés) sensés représenter au mieux le contenu d'un document. Trois types de termes sont successivement identifiés à l'issue de cette étape: les termes simples, les termes orphelins et les termes complexes. Comme appui pour l'extraction des termes complexes et la séparation des termes en mots simples et orphelins, la base lexicale *WordNet* est utilisée.

❖ ***Phase 1 : L'extraction des termes complexes :***

Pour chaque occurrence d'un terme $t_i$ à analyser et avant tout autre traitement, en particulier avant d'élaguer les mots vides du document, on commence par extraire l'ensemble des collocations $C_i$ de WordNet qui commencent par $t_i$ . La sélection des collocations est faite à partir d'une liste qui comporte toutes les collocations de WordNet2.1 préalablement construite. Par la suite on ordonne les éléments de $C_i$ selon leurs longueurs dans un ordre décroisant et on obtient une nouvelle liste $C_i$', puis on projette chaque élément de $C_i$' sur des *concepts candidats* formé en combinant par des « _ » les mots adjacents de $t_i$ dans le texte. Si un concept candidat s'apparie avec une collocation, elle est retenue comme terme complexe et insérée dans l'ensemble $L_{complexe}$. Si aucune collocation de $C_i$' ne s'apparie avec un concept candidat de $t_i$, alors $t_i$ est passé à la **phase 2** « séparation des termes en simples et orphelins ».

❖ ***Phase 2 : Séparation des termes en simples et orphelins :***

Cette phase reçoit un terme $t_i$ de la phase précédente et avant tout autre traitement elle vérifie d'abord si $t_i$ est un mot vide ou pas. Si $t_i$ est vide, il est ignoré. Sinon, si $t_i$ possède une entrée dan Wordnet alors il est inséré dans l'ensemble $L_{simples}$, sinon dans l'ensemble $L_{orphelins}$.

## III.4.2.Désambiguïsations des termes :

L'objectif de la désambiguïsation consiste à déterminer, parmi les différents sens possibles d'un terme, celui qui convient dans un contexte particulier. La désambigüisation concernera uniquement les mots simples (les collocations étant généralement désambiguïsées par nature). Les mots sont désambiguïsés dans leur contexte. Deux types de contextes sont considérés:

(1) *Le contexte local* : il désigne l'ensemble des mots simples et des collocations de la phrase qui contient l'occurrence du mot $m_i$ examiné. On note le contexte local d'un mot simple $m_i$ par : $CL_i$,

(2) *Le contexte global* : le contexte global $CG_i$ d'un mot simple $m_i$ dans un document $D_i$, est l'union de tout ses contextes locaux $CG_i = LC_k \cup LC_{k+5} \cup \ldots\ldots \cup LC_l$ dans le document.

Soit donc l'ensemble des termes de $L_{simples}$ et $t_i$ un terme dans $L_{simple}$ Cette approche est basée sur le calcul d'un score (Score) pour chaque concept (sens) associé à chaque $t_i \in Contexte$

$$Score(C_j^i) = \sum_{\substack{l \in [1..m] \\ l \neq i}} \sum_{k \in [1..n_l]} Dist(C_j^i, C_k^l)$$

Où :

$m$ est le nombre de termes dans *Contexte*,

$n_l$ représente le nombre de sens de WordNet propres à chaque terme $t_l$ ;

$Dist(C_j^i, C_k^l)$ est une mesure de proximité sémantique entre les concepts.

Selon que l'on considère le contexte local ou le contexte global, nous distinguons deux approches de désambigüisation :

1. Une approche de désambiguïsation locale.
2. Une approche de désambigüisation globale.

### III.4.2.1.La mesure de similarité :

Diverses mesures de proximité sémantique ont été proposées dans la littérature qui sont soit basées sur le chemin (path based measures) entre les deux concepts à comparer telles que définies par exemple dans [Rada et al. ,89] [Leacock et al., 94] [Jiang et al., 97], sur la notion de contenu d'information (Information Content ou IC) telle que définie par Wu et Palmer [Wu et al., 94] , Resnik [Resnik, 99], sur une combinaison du chemin et du contenu d'information [Lin, 98] , ou sur l'algorithme de Lesk que Patwardhan, Banerjee et Pederson [Patwardhan et al., 03] ont adapté à WordNet. Pour notre par nous avons opté pour la mesure de similarité Lin.

- **La mesure de Lin :**

Selon le Théorème de Similarité de Lin [Lin, 98], la similarité entre deux concepts est mesurée par le ratio du contenu d'information nécessaire pour mesurer la "communalité" des deux concepts, sur le montant du contenu d'information nécessaire pour décrire chacun des deux concepts.

La « communalité » entre deux concepts dépend du contenu d'information (IC) de leur plus spécifique subsumer (lcs) et du contenu d'information des deux concepts eux même. Cette mesure est proche de celle de Jiang et Conrath même si elles sont développées séparément :

$$Sim_{\_lin}(c_1, c_2) = \frac{2 \cdot IC(lcs(c_1, c_2))}{IC(c_1) + IC(c_2)}$$

Ceci peut être vu comme le contenu d'information de l'intersection des deux concepts (multiplié par deux) qui est divisé sur leur somme.

### III.4.2.2. Algorithme de désambigüisation utilisé dans SemTerrier :

Les algorithmes de désambiguïsation locale et globale que nous avons implémentés s'appuient sur un modèle extrêmement simple décrit dans la **Table III.1**. Ce modèle prend en entrée un mot à désambiguïser ainsi qu'une liste de ses sens candidats et produit en sortie le sens sélectionné.

---

Algorithme de désambigüisation :

Entrée:

t, un mot à désambiguïser

S= {$S_1$, $S_2$,…, $S_n$}, les sens candidats ordonnés en ordre décroissant de fréquence

Sortie :

Sens, l'indice dans S du sens retenu

```
score ←0 ;
sens ← 1                                   //choix par défaut du numéro d'ordre du sens le plus fréquent

 pour chaque mot à désambiguïser t
        déterminer C(t), le contexte de t        //contexte du mot cible qui peut être soit CL ou bien CG

        pour chaque sens candidat s_i de t
                    sup ← 0
            pour chaque mot w du contexte C(t)

                        pour chaque sens s'_k de w

                            sup ← sup + lin (s_i, s'_k)          //cumul des superpositions
                          fait ;
              fait ;
                            si sup > score alors
                            score ← sup
                           sens ← i
         fait ;

fait ;
```

**Table III.1: Algorithme de désambiguïsation des termes.**

### III.4.3.La Pondération des concepts :

Une fois les termes (concepts) extraits du document, il faut leur affecter un poids pour indiquer leur importance dans le document. L'approche de pondération proposée par les auteurs prend on compte la fréquence du terme dans le document ainsi que sa proximité sémantique avec d'autres concepts du document. C'est cette approche que nous avons implémenté dans SemTerrier.

La formule de pondération est donnée par :

$$W\left(C^{i}\right) = \alpha * tf\left(C^{i}\right) + \left(1-\alpha\right)\sum_{i \neq l} Dist\left(C^{i}, C^{l}\right)$$

Où :

$C^{i}$ est le concept

*tf* la fréquence tu terme dans le document.

$\alpha$ est un facteur de pondération qui permet de balancer la fréquence par rapport à la pertinence.

Ce schéma de pondération permet aussi la pondération des termes complexes et des termes orphelins. Dans ce dernier cas, seule la fréquence est considérée, les proximités sémantiques inexistantes, sont initialisées à zéro.

### III.4.4. Etiquetage syntaxique des textes :

Les mots simples peuvent avoir plusieurs sens associés dans WordNet. Dans ce cas, ils sont ambigus. Il faut les désambiguïser. La désambiguïsation des sens des mots doit tenir compte de la partie de discours (part of speech) [Krovetz, 97]. Cette dernière peut contribuer à déterminer le bon sens du mot ambigu. Nous avons donc estimé nécessaire d'étendre l'approche étudiée par un module d'étiquetage syntaxique. Nous avons en particulier utilisé l'étiqueteur syntaxique *Stanford POS Tagger*[18] que nous présentons en Annexe 3. Stanford

[18] http://nlp.stanford.edu/software/tagger.shtml

POS Tagger possède aussi un module de lemmatisation que nous avons expérimenté pour la lemmatisation des termes.

## III.5.Architecture globale de SemTerrier:

SemTerrier est basé sur la plateforme Open Source Terrier auquel nous avons ajouté l'approche d'indexation sémantique présenté ci-dessus.

### III.5.1. Présentation générale :

Nous avons intégré dans Terrier quatre éléments :

- Un module d'indexation sémantique
- Un nouveau modèle de pondération
- Une base de données lexicale, *WordNet* en l'occurrence
- Un étiqueteur Syntaxique, le *Stanford POS Tagger*.

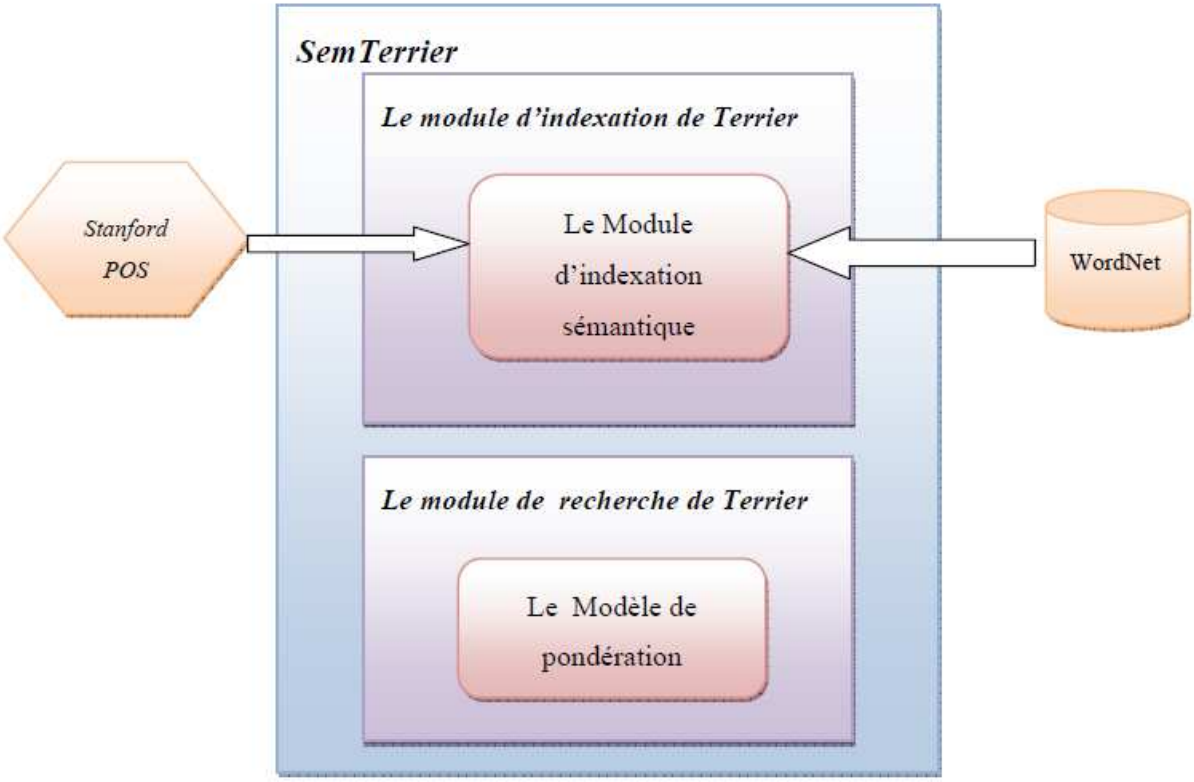


**Figure .III.4: Présentation générale de SemTerrier.**

Comme le montre le schéma ci-dessus, SemTerrier contient un module d'indexation sémantique qui consiste en un ensemble de plug-ins. Ce module utilise Wordnet et Stanford POS Tagger.

Nous avons aussi développé un nouveau modèle de pondération sémantique de concepts, que nous avons intégré dans Terrier, et dont la particularité et de prendre en compte la similarité entre les concepts dans le calcul des poids.

## III.5.2. Présentation détaillée :

Il existe deux API importante dans Terrier qui sont « Indexing API » et « Querying API », que nous avons, comme le montre la figue suivante, modifié par intégration des plug-ins de l'indexeur sémantique.

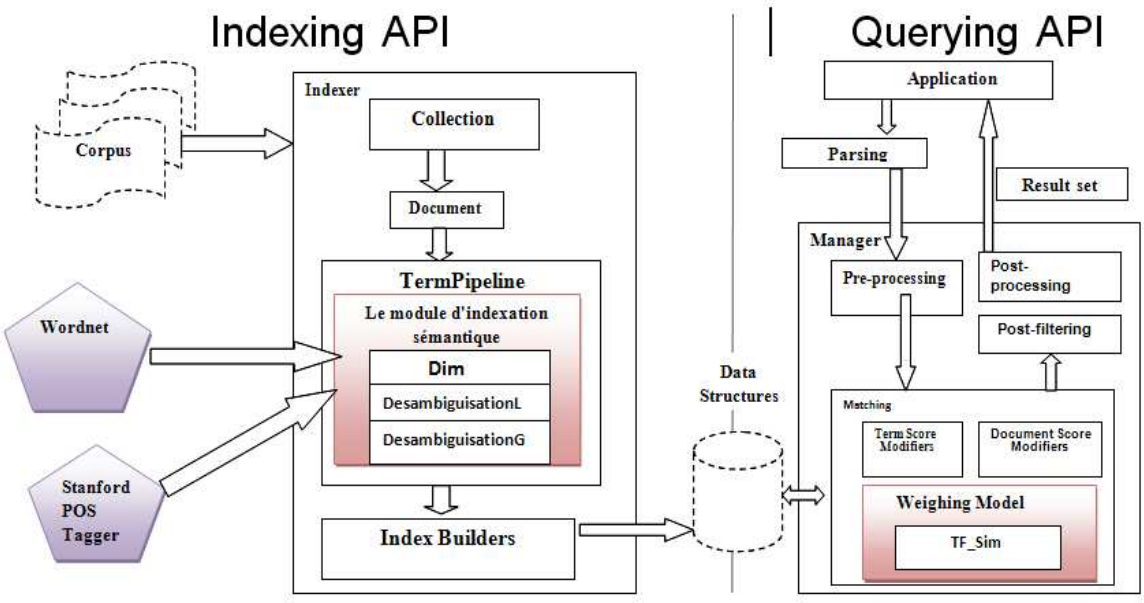


**Figure .III.5 : Présentation détaillé de SemTerrier.**

L’une des caractéristiques principales de Terrier est son Open Source complet et extensible dont nous avons utilisé une partie pour développer SemTerrier.

Le modèle de classes présenté en **Figure .III.8** représente les différentes contributions apporté par SemTerrier.

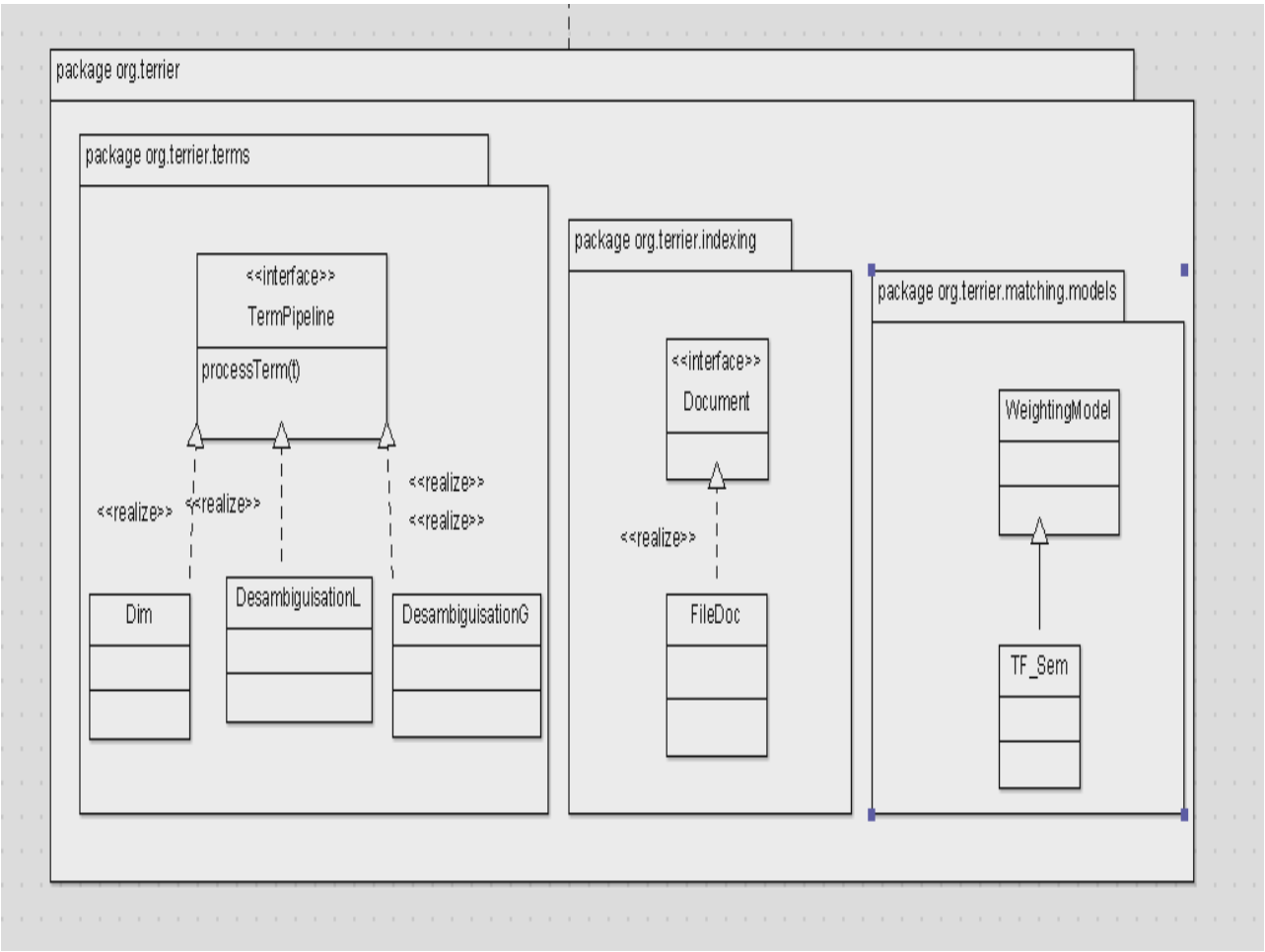


**Figure .III.6 : diagramme de classes.**

Nous décrivons les classes modélisées dans le **Table III.2**:

| Classe | Descriptif |
|---|---|
| ***FileDoc*** | C'est un nouveau parseur qui permet de parcourir les documents ligne par ligne |
| ***Dim*** | Cette classe permet la détection des termes d'index : la détection des colocations, des mots simples et des mots orphelins |
| ***DesambiguisationL*** | Cette classe implémente l'approche de désambiguïsation locale présentée dans le chapitre.III |
| ***DesambiguisationG*** | Cette classe implémente l'approche de désambiguïsation globale présentée dans le chapitre.III |
| ***TF_Sem*** | C'est la classe qui implémente un nouveau modèle de pondération qui correspond à l'approche de pondération des concepts présentés dans le chapitre.III |

**Table III.2: Descriptif des classes proposées.**

# III.6. Conclusion :

Nous avons présenté dans ce chapitre un double aspect de notre contribution : (1), la description détaillée d'une nouvelle approche d'indexation sémantique basée sur l'utilisation conjointe de Wordnet et de Stanford POS Tagger. Cette dernière permet la détection des termes d'index, leur désambiguïsation et la pondération des concepts. (2) notre seconde contribution n'est pas des moindres, et consiste en une extension de Terrier à l'indexation sémantique. Nous avons ainsi décrit notre contribution à l'intégration de cette nouvelle approche dans la plateforme Terrier.

# Chapitre IV

# Résultats et expérimentations

## IV.1. Introduction :

Nous avons présenté dans le chapitre précédant, nos contributions pour un système d'indexation et de recherche fondé sur la sémantique : SemTerrier. Nous présentons dans ce chapitre des expérimentations préliminaires concernant l'approche d'indexation sémantique proposée par [Amirouche et Azzoug, 11] et que nous avant implémenté dans Terrier.

L'Approche d'indexation sémantique trouve son application sur des textes plats. Nous présentons brièvement le corpus utilisé pour l'expérimentation. Puis par la suite, nous présentons les résultats de l'évaluation de l'approche d'indexation sémantique utilisée.

## IV.2. Environnement technologique :

SemTerrier est développé entièrement en Java (1.6). Le choix de Java c'est imposé vu que l'Open Source de Terrier est entièrement écrit en java. Java offre aussi un certain nombre d'APIs qui ont servi à la réalisation de SemTerrier, en voici la liste.

- **JAWS [19] (*Java API for WordNet Searching*):** C'est une API qui permet d'accéder à la base de données de Wordnet et qui ne peut être utilisé qu'avec la version 2.1. Ce choix est de plus motivé par la simplicité de ses déclarations et de son utilisation.

  **Exemple de code:**

```
System.setProperty ("wordnet.database.dir", "C:\\Program Files\\WordNet\\2.1\\dict\\");
WordNetDatabase database = WordNetDatabase.getFileInstance();
        Synset[] synsets = database.getSynsets(word);
```

- **Java WordNet Similarity.beta.11.01[20]** : C'est l'API qui nous permet de calculer la similarité entre deux concepts. Elle propose différentes mesure de similarité tel que : Lin, Resnik, Lesk, ect. Elle utilise la JWS comme API d'accès à WordNet.

  **Exemple de code:**

```
JWS ws = new JWS (dir, "2.1");
        Lin lin = ws.getLin();
            scores = lin.lin (mot, j, word, pos);
```

- **Eclipse :** Pour le développement de nos applications nous avons utilisé l'environnement de développement (IDE) Eclipse.

# IV.3. Évaluation expérimentale :

L'objectif de notre évaluation expérimentale est d'étudier l'impact de l'approche d'indexation sémantique sur la performance de la recherche d'information. Nous décrivons dans ce qui suit le cadre d'évaluation et présentons, les résultats obtenus.

[19] http://lyle.smu.edu/~tspell/jaws/index.html
[20] http://www.cogs.susx.ac.uk/users/drh21/

## IV .3.1. Cadre d'évaluation :

### IV .3.1. 1. Point sur la collection de test utilisé:

Pour mener cette expérimentation nous avons opté pour l'utilisation de la collection Muchmore[21]. Vu la complexité temporelles induits par les méthodes d'identification de termes d'index, de pondération et de désambiguïsation nous avons été contraint de réduire la collection à 35 documents et d'utilisé que 5 requêtes.

- Voici un extrait d'un document Muchmore (extrait de Arthroskopie.00130041.eng) :

*In a prospective study, 35 consecutive patients with an isolated posterior knee instability were stabilized arthroscopically in a new all-inside, double bundle technique using the autologous quadrupled semitendinosus and the doubled gracilis tendon.*

- Voici un exemple de requête Muchmore :

*Arthroscopic treatment of cruciate ligament injuries*

- Exemples de jugements de pertinence :

*1 0 DerUnfallchirurg/01030864 1*
*1 0 DerUnfallchirurg/60990869 1*
*1 0 DerUnfallchirurg/81010491 1*

### IV.3.1.2. Protocole d'évaluation :

L'approche est évaluée en utilisant le système de RI Terrier .L'évaluation est effectuée selon le protocole TREC. Plus précisément, chaque requête est soumise au système de RI avec les paramètres fixés. Terrier utilise toutes les mesures de précision comme P@5, P@10 qui sont respectivement la précision moyenne aux 5, 10 premiers documents retournés sur l'ensemble de 35 requêtes. Pour chaque requête, les 1000 premiers documents sont renvoyés par le système et les précisions moyennes (P@5, P@10) sont calculées pour mesurer la performance de la RI.

[21] http://muchmore.dfki.de/about.html

## IV.3.2. Résultats expérimentaux :

Pour évaluer la performance de l'approche d'indexation sémantique définit dans le chapitre précédant, nous avons réalisé une série d'expérimentation dont le but est de comparer l'indexation sémantique par rapport à l'indexation classique en utilisant la configuration standard sous la plateforme Terrier avec le schéma de pondération de référence OKAPI BM25 (Robertson et al, 1998). Cette configuration est utilisée comme la base d'évaluation comparative (baseline), notée BM25.

### IV.3.2.1. Evaluation de l'approche d'indexation sémantique:

Le tableau. IV .1 présente les résultats obtenus pour l'ensemble des requêtes tests. Les résultats montrent que notre approche d'indexation sémantique qui utilise la pondération une pondération basé sur les concepts présenté dans le chapitre précédant offre une précision moyenne plus importante que l'indexation classique basée sur le modèle de pondération BM25.

| Information | BM25 | Indexation sémantique (TF_Sem ) |
|---|---|---|
| Number of queries<br>Retrieved<br>Relevant<br>Relevant retrieved | 5<br>64<br>30<br>29 | 5<br>64<br>30<br>29 |
| Average Precision<br>R Precision | 0.7835<br>0.8083 | 0.8118<br>0.8083 |
| Precision at 1 :<br>Precision at 2 :<br>Precision at 3 :<br>Precision at 4 :<br>Precision at 5 :<br>Precision at 10 :<br>Precision at 15 :<br>Precision at 20 : | 0.8000<br>0.8000<br>0.8667<br>0.9000<br>0.8000<br>0.4600<br>0.3733<br>0.2800 | 0.8000<br>0.9000<br>0.9333<br>0.9500<br>0.8000<br>0.4600<br>0.3733<br>0.2800 |

| | | |
|---|---|---|
| Precision at 30 : | 0.1933 | 0.1933 |
| Precision at 0%: | 1.6333 | 1.8333 |
| Precision at 10%: | 1.6333 | 1.8333 |
| Precision at 20%: | 1.7167 | 1.8833 |
| Precision at 30%: | 1.7933 | 1.4833 |
| Precision at 40%: | 1.4433 | 1.4500 |
| Precision at 50%: | 1.5017 | 1.2250 |
| Precision at 60%: | 1.2083 | 1.3977 |
| Precision at 70%: | 1.3797 | 0.8310 |
| Precision at 80%: | 0.9363 | 0.5961 |
| Precision at 90%: | 0.5961 | 0.5961 |
| Precision at 100%: | 0.5961 | |
| Average Precision: | 0.7835 | 0.8118 |

**Tableau IV.1: Résultat de l'évaluation de l'approche d'indexation sémantique avec La plateforme de RI Terrier.**

# IV.4 Conclusion :

Dans ce chapitre nous avons présenté notre expérimentation et qui porte sur l'approche d'indexation sémantique décrite dans ce mémoire. Selon les données statistiques il en résulte une amélioration légère de la précision, toute fois en ne peut pas généraliser a partir de nos données expérimentales puisque l'expérimentation à été réalisé sur un corpus de 35 documents. Il faudra alors refaire une évaluation sur un corpus plus importante.

# Conclusion générale & perspectives

De nouveaux modèles de production, de traitement et réception de l'information émergent. Si l'arrivée du Web a eu un impact important dans le champ scientifique, économique ou social, l'avènement du Web 2.0 (notamment les réseaux sociaux) élargit cet impact à notre espace privé. Dans les deux cas, les sources d'informations s'accroissent rendant le besoin de disposer de systèmes permettant un accès intelligent à l'information de plus en plus indispensable.

La première partie du mémoire consacrée à la présentation des principaux modèles de recherches et à la description des processus d'indexation a permis de dégager les choix réalisés par les concepteurs de ces systèmes, mais aussi leurs limites.

L'analyse des travaux de recherche nous a permis de constater qu'un des obstacles à toute recherche d'information est de trouver, à partir d'une requête donnée, une formulation dans les termes de la base documentaire. L'efficacité d'une recherche d'information nécessite la connaissance de la façon dont la base documentaire a été indexée. La recherche documentaire est donc une tâche assez complexe qui nécessite la mise en relation d'un besoin d'information imprécis et le contenu des bases de données documentaires. Pour mener sa recherche, l'usager doit en être en mesure de maîtriser plusieurs savoir-faire dont le plus important est sans doute d'ordre sémantique : comment définir et formuler ce qu'on l'on cherche d'une façon compréhensible par le SRI ? Comment élaborer une stratégie de recherche pour affiner et/ou élargir les résultats obtenus [Borgman, 1996].

Les conséquences de cet écart de vocabulaire sont une des causes majeures du taux d'échec et de surcharge d'information. En effet, le langage ne permet pas toujours aux usagers d'extérioriser et de bien exprimer leurs besoins d'informations, du fait de l'économie de

langage qu'ils pratiquent. Par ailleurs, les travaux de [Spink, 2004] sur l'usage des moteurs de recherches et des bibliothèques numériques ont montré que les ressources des SRI sont souvent sous-utilisées et que les outils mis à la disposition de l'usager final pour explorer le nombre élevé de réponses sont insuffisantes et inadaptées. Il s'agit donc à la suite de ces observations de proposer un système permettant d'augmenter la précision des réponses obtenues par le SRI.

Comme nous l'avons vu tout au longs de ce mémoire, les SRI sont naturellement associés à du langage naturel qui contient un très grand nombre d'ambiguïtés et de difficultés qu'un être humain arrive généralement à dominer sans problème. L'utilisation de connaissances linguistiques devrait donc augmenter les performances des SRI [De Loupy, 2004]. C'est la raison pour laquelle nous nous sommes orientés vers une approche basée sur la sémantique en contribuant à son intégration à la plateforme de RI Terrier : *SemTerrier*. Nous avons au préalable présenté les travaux de l'indexation sémantique dans la deuxième partie de notre mémoire. Afin d'augmenter la précision de notre nouveau système nous avons utilisé un module de désambiguïsation sémantique. Or selon Voorhees, cité par [De loupy, 2004], les performances d'une ressource comme WordNet sont importantes notamment pour les requêtes courtes. Le principal apport de ce mémoire est donc l'intégration d'un module d'indexation sémantique (un détecteur de collocation et un désambiguisateur) dans la plateforme de RI Terrier mais aussi WordNet et Stanford POS Tagger. Nous avons aussi ajouté un nouveau modèle de pondération permettant d'augmenter la précision de la recherche en intégrant la similarité dans le calcule des scores.

L'utilisation conjointe de WordNet et de Stanford POS Tagger offre certainement de nouvelles perspectives à la recherche d'information qui nécessitent d'êtres évaluées et enrichies. Les résultats de notre évaluation ont confirmé, malgré la faiblesse du corpus (35 documents), cette hypothèse de recherche. Il serait judicieux d'effectuer de nouvelles évaluations sur un corpus plus important. Par ailleurs, la généralisation de plate forme de SRI en open source, nous permettra sans doute de tester notre approche (Wordnet & POS) sur des outils de type LUCENE, GATE, X-IOTA. En fin, dans notre nouveau système, il n'y a pas d'information concernant le domaine d'application des concepts issus de WordNet (WordNetDomain). Or, cette information serait très utile pour la désambiguïsation du sens.

Ces perspectives devraient contribuer à la validation de notre modèle en comparant ses performances avec des systèmes comme OKAPI dans des campagnes d'évaluation de type TREC.

# Références bibliographiques

[Agirre et al., 01] E. Agirre and D. Martinez. Knowledge sources for Word Sense Disambiguation. In Proceedings of the Fourth International Conference TSD 2001, Plzen (Pilsen), Czech Republic. Published in the Springer Verlag Lecture Notes in Computer Science series. Vaclav Matousek, Pavel Mautner, Roman Moucek, Karel Tauser (eds.) Copyright Springer-Verlag.

[Amirouche, 08] Fatiha Boubekeur Amirouche. Contribution à la définition de modèles de recherche d'information flexibles basés sur les CP-Nets. Thèse de Doctorat en Informatique de l'Université Paul Sabatier de Toulouse (Sciences). Juillet 2008.

[Banerjee, Pedersen ,2002] Banerjee, Satanjeev and Ted Pedersen. ``An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet" In: Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLING-02), Mexico City, February, 2002.

[Baziz, 05] Baziz M. Indexation Conceptuelle Guidée Par Ontologie Pour La Recherche d'Information. Thèse de Doctorat en Informatique de l'Université Paul Sabatier de Toulouse (Sciences). Décembre 2005.

[Belhassen, 99] Amina Sayeb Belhassen. Prise en compte de 1 'aspect utilisateur au niveau de la recherche documentaire sur Internet. Article, Laboratoire PGL, ENSI, Tunis, 1999.

[Boughanem, 98] M .Boughanem, C. Chrisment, C. SOULE-DUPUY, Query modification based on Relevance Back-propagation in ad-hoc environment. IPM: Information Process and Mangement 1998.

[Boughanem et al., 2004] Mohand Boughanem, Lynda Tamine. *Connexionisme et génétique pour la recherche d'information.* Dans : *Les systèmes de recherche d'informations*. M. Ihadjadène (Eds.), Hermès, p. 77-99, 2004.
Accès : ftp://ftp.irit.fr/IRIT/SIG/Connexionisme-génétique-2004.pdf

[Bourne, 79] C.Bourne & B.Anderson, DIALOG LabWorkbook. Second edition, Looked Information Systems, PaloAlto, Californie, USA. 1979.

[Borgman, C.L. 1996]. *Why are online catalogs still hard to use?* Journal of the American society for information science, 47(1996):7, 493-503.

[Briscoe, 91] Briscoe, T. "Lexical Issues in NLP", en E. Klein & F. Veltman (eds.) Natural Language and Speech. The Netherlands : Springer-Verlag. (1991).

[C. de Loupy, 2004], "Traitement automatique des langues et systèmes de recherche d'information" in *Systèmes de recherche d'information*. Éditions Hermès. pp. 139-158: 2004

[Cowie, Guthrie et Guthrie 92] Lexical disambiguation using simulated annealing. *14th International Conference on Computational Linguistics (COLING- 1992)*, 359–365.

[Briscoe, 91] : Briscoe, T. "Lexical Issues in NLP", en E. Klein & F. Veltman (eds.) Natural Language and Speech. The Netherlands : Springer-Verlag. (1991).

[Cowie, Guthrie et Guthrie 92] Lexical disambiguation using simulated annealing.*14th International Conference on Computational Linguistics (COLING- 1992)*, 359–365.

[Fellbaum, 98] *Wordnet : An electronic lexical database*. Cambridge, Massachusetts: MIT Press. (ISBN 0-262-06197-X)

[Fluhr, 85] C.Fluhr, SPIRIT « A linguistic and probabilistic information storage and retrieval system.

[Gaussier et al., 97] E. Gaussier, G. Grefenstette, et M. Schulze. Traitement du langage naturel et recherche d'informations : quelques expériences sur le français. Premières Journées

Scientifiques et Techniques du Réseau Francophone de l'Ingénierie de la Langue de l'AUPELF-UREF, 1997.

[Gaussier et al., 2000] E. Gaussier, G. Grefenstette, D. Hull, et C. Roux. Recherche d'information en français et traitement automatique des langues. Revue Traitement Automatique des Langues (TAL), 41(2):473–494, 2000.

[Gessler, 93] N. Gessler, George Boole et l'algèbre de la logique. Etudes logiques, Neuchâtel, 1993.

[Golder, 06] Scott A. and Huberman, Bernard A, *The structure of collaborative tagging systems*, 2006 (http://www.hpl.hp.com/research/idl/papers/tags/tags.pdf

[Gonzalo & al., 98] J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarran. 1998. Indexing with WordNet synsets can improve text retrieval. In Proceedings of COLING-ACL '98 Workshop on Usage of Word.Net in Natural Language Processing Systems, Montreal,Canada, August.

[Grishman, MacLeod & Meyers, 94, 99] : comlex syntax : Building a computational lexicon. *15th International Conference on Computational Linguistics (COLING-1994)*, 268–272. A large syntactic dictionary for natural language processing. *Computers and the Humanities*

[Grolier] Grolier Multimedia Encyclopedia CD-ROM. Grolier interactive Inc., 90 Sherman Turnpike, Danbury, CT 06816, USA.

[Grossman, 93] D. A. Grossman, O. Frieder, Information retriveval, Algorithms and heuristics. Kluwer Academic Publishers 1993.

[Grossman et Frieder, 1998] D. Grossman, O. *Frieder: Information Retrieval. Algorithms and Heuristics*. Kluwer Academic Publishers, 1998.

[Guthrie et al.,91 ] : Subject-dependent cooccurrence and word sense disambiguation. *29thAnnual Meeting of the Association for Computational Linguistics (ACL-1991)*, 146–152.

[Harman, 92] D. Harman, *Relevance feedback revisited*. Proceedings of ACM SIGIR1992.

[Ide and al ,90] : Ide, N., & Veronis, J. Mapping Dictionaries : A Spreading Activation Approach,Sixth Annual Conference of the UW Centre for the New Oxford English Dictionary (pp.52-64). Waterloo (Canada). (1990).

[Ide and Veronis ,1993a] : Knowledge extraction from machine-readable dictionaries: An evaluation. *3rd International European Association for Machine Translation (EAMT) Workchop* « *Machine Translation and the Lexicon* »

[Ide and Veronis ,1993b] : Refining taxonomies extracted from machine-readable dictionaries. *Research in Humanities Computing II*, 145–159.

[Ihadjadene, 01] IHADJADENE M., ROLE F., CHAUDIRON S., « Exploitation de l'indexation auteur pour le filtrage de documents structurés complexes », in Filtrage et résumé automatique de l'information sur les réseaux - Actes du 3ème Colloque du Chapitre français de l'ISKO, Chaudiron S. et Fluhr C. (sous la dir. de), Nanterre 5 et 6 juillet 2001, Université de Paris X, p. 39-47.

[Kammoun, 97] Hager Kammoun. *Classification automatique des textes dans unfond documentaire.* Mémoire de *DEA*, Faculté des sciences de Tunis, 1997.

[Kaplan, 55] An experimental study of ambiguity and context. *Mechanisme Translation, 2*, 39–46. (Première publication : Mimeographed, November 1950)

[Lefèvre, 00] : *La recherche d'information, du texte intégral au thésaurus*, Paris, Hermès, 253 p.

[Leloup, 98] Catherine Leloup. *Moteurs d'indexation et de recherche: Environnement Client-Serveur Internet et Intranet.* Eyrolle, *1998*

[Lelu, 92] A Lelu, C. François: *Automatic generation of hypertext links in information retrieval systems*. Communication of colloque ECHT'92, ACM Press, New York, 1992.

[Luhn, 58] Luhn, H. The automatic creation of literature abstracts. IBM Journal of Research and Developpment 24, 2 (1958), 159–165.

[Maniez, 02] Maniez (J.). *Actualité des langages documentaires ; Fondements théoriques de la recherche d'information*, ABDS, Paris, 2002

[MONTEIL, 95] MONTEIL Marie-Gaëlle, *Indexation manuelle et automatique :comparaison et perspectives*, IDT 95, 12e Congrès, Paris, 12-15 juin 1995, pp. 214-215.

[Nie et al., 99] : Fuji Ren, Lixin Fan, Jian-Yun Nie, SAAK Approach: How to Acquire Knowledge in an Actual Application System, IASTED International Conference on Artificial Intelligence and Soft Computing, Honolulu, 1999, pp.136-140.

[Salton71] Salton (Gerald). – *The SMART retrieval system: experiments in automatic document processing*. – Prentice Hall, 1971.

[Salton et al., 73] Salton, G., and Yang, C. On the specification of term values in automatic indexing. In Journal of Documentation, 29 (1973), 351–372.

[Salton, 88] Salton, G. Syntactic approaches to automatic book indexing. In Proc. of the annual meeting on Association for Computational Linguistics (ACL) (1988), Department of Computer Science, Cornell University, Ithaca, New York, pp. 204–210.

[Salton et al., 88] Salton, G., and Buckley, C. Term-weighting approaches in automatic textretrieval. Information Processing & Management (IPM) 24, 5 (1988), 513–523.

[Salton, 94] G. Salton & J. Allan, *Automatic Text Decomposition and Structuring*. Actes du Congrés RIAO'94, Intelligent Multimedia Information on Retrieval Systems and Management, New York. 1994.

[Search, 02] Search Engine Showdown Size statistics. http://www.searchengineshowdown.com/stats/sizeest.shtml.

[Singhal et al., 97] Singhal, A., Mitra, M., and Buckley, C. 1997. Learning routing queries in a query zone. In Proceedings of the 20th Annual international ACM SIGIR Conference on Research and Development in information Retrieval (Philadelphia, Pennsylvania, United States, July 27 - 31, 1997). N. J. Belkin, A. D. Narasimhalu, P. Willett, and W. Hersh, Eds. SIGIR '97. ACM Press, New York, NY, 25-32.

[SparkJones, 71]: K. Sparck Jones. Automatic keywords classification for information retrieval.

[Sparck Jones, 79] Karen Sparck Jones: Experiments in relevance weighting of search terms. Inf. Process. Manage. 15(3): 133-144, 1979.

[Spink A, 2004] Web Search: Public Searching of the Web. Kluwer Academic Publishers.

[Robertson, 76] S.Robertson & K.Sparck Jones, *Relevance Weighting for Search Terms.* Journal of the American Society for Information Science, Vol 27, N°3, 1976

[Robertson et al., 97]: S. E. Robertson and S. Walker. On relevance weights with little relevance information. In Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, pages 16–24. ACM Press, 1997

[Robertson, 99] S.E.Robertson, S.Walker, M.Beaulieu, *Automatic Adhoc Filtring, VLC and Interactive Track*. In Poceeding of the 7th Text Retrieval Conference TREC7, 1999.

[Rocchio, 71] J. Rocchio : *Relevance feedback information retrieval*. In Gerald Salton (editor), The SMART retrieval system- experiments in automated document processing. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[Rocchio, 71] J. Rocchio: *Relevance feedback information retrieval*. In Gerald Salton (editor), The SMART retrieval system- experiments in automated document processing. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[TREC9, 2000] proceedings of the Ninth Text REtrieval Conference (TREC-9) held in Gaithersburg, Maryland, November13-16, 2000.
URL: ***http://trec.nist.gov/pubs/trec9/t9_proceedings.html***.

[Trigano, 94] Philippe Trigano. *Indexation automatique et sauvegarde des connaissances de l'entreprise.* Article : Projet ISMICK 94, Université de Compiègne, France, 1994.

[Veronis et al., 90] Veronis, J. and Ide, N. Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. 13th International Conference on Computational Linguistics (COLING-1990), 2, 389–394. 1990.

[Veronis and Ide, 1991] Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. *13th International Conference on Computational Linguistics (COLING-1990), 2*, 389–394.

[Voorhees, 93] Using WordNet to disambiguate word senses for text retrieval. *Association for Computing Machinery Special Interest Group on Information Retrieval (ACM-SIGIR-1993): 16thAnnual International Conference on Research and Development in Information Retrieval*, 171–180.

[Walker, 97] S. Waller, S. E. Robertson, M. Boughanem, G. J. F. Jones, K. Sparck Jones. *Okapi at TREC-6 automatic and ad hoc*, VLC routing, filtring and QSDR. Proceeding of TREC-6, 1997.

[Weaver, 49] Translation. In N. William & A. D. Booth (Eds.), *Machine translation of languages* (pp. 15–23). New-York: J. Wiley and Sons. (Reprinted from *Mimeographed*, 1949, 12 pp.)

[Weiss, 73; Kelly et al., 75] Weiss, S. F. (1973). Learning to disambiguate. Information Storage and Retrieval, 9,33_41.
*Computer recognition of english word senses.* North-Holland Publishing. North-Holland, Amsterdam.

[Wilks et al., 90] Y. Wilks, D. Fass, C. Guo, J.E. Mcdonald, T. Plate, B.M. Slator. Providing Machine Tractable Dictionary Tools. In Machine Translation, 5 99-154. (1990).

[Yarowsky, 92] Word sense disambiguation using statistical models of Roget's categories trained on large corpora. *14th International Conference on Computational Linguistics (COLING-1992)*, 454–460.

[Yates, 99] R. B.Yates, R. Neto, *Modern Information Retrival*. ACM Press, Addison Wesley, 1999.

[Zipf, 49] Zipf, H. Human behaviour and the principle of least effort. Addison- Wesley, Cambridge, Massachuset

# Annexe 1:

# WordNet

## 1.1. Origine de WordNet :

WordNet est un réseau sémantique de la langue générale. Plus précisément, c'est une base de données lexicale construite par un groupe de psychologues et de linguistes du laboratoire de sciences cognitives de l'université de Princeton, dirigé par le professeur Georges A. Miller. Elle a été initialement conçue dans le cadre d'un projet lancé en 1985 et gracieusement financé par l'agence de renseignements américaine (CIA), avec l'objectif de tester les déficits lexicaux dans des expériences de psychologie cognitive. A l'origine, ces concepteurs ne prétendaient construire ni une structure conceptuelle, ni une ontologie, mais bien une ressource lexicale rendant compte de l'usage des mots et de leur mise en relation dans la langue. Ce n'est qu'ensuite que le réseau lexical de WordNet a été perçu comme une représentation conceptuelle (Lexical Conceptual Graph ou LGC) [Guarino & al., 99] qui pourrait tenir lieu d'ontologie.

## 1.2. Contenu de WordNet

WordNet couvre la majorité des noms, verbes, adjectifs et adverbes de la langue Anglaise. Sa dimension ainsi que le domaine de la langue générale qu'il traite lui permettent souvent de couvrir les sujets traités dans les collections de test conventionnelles de la RI (TREC, CLEF). Ces dernières sont le plus souvent de type presse.

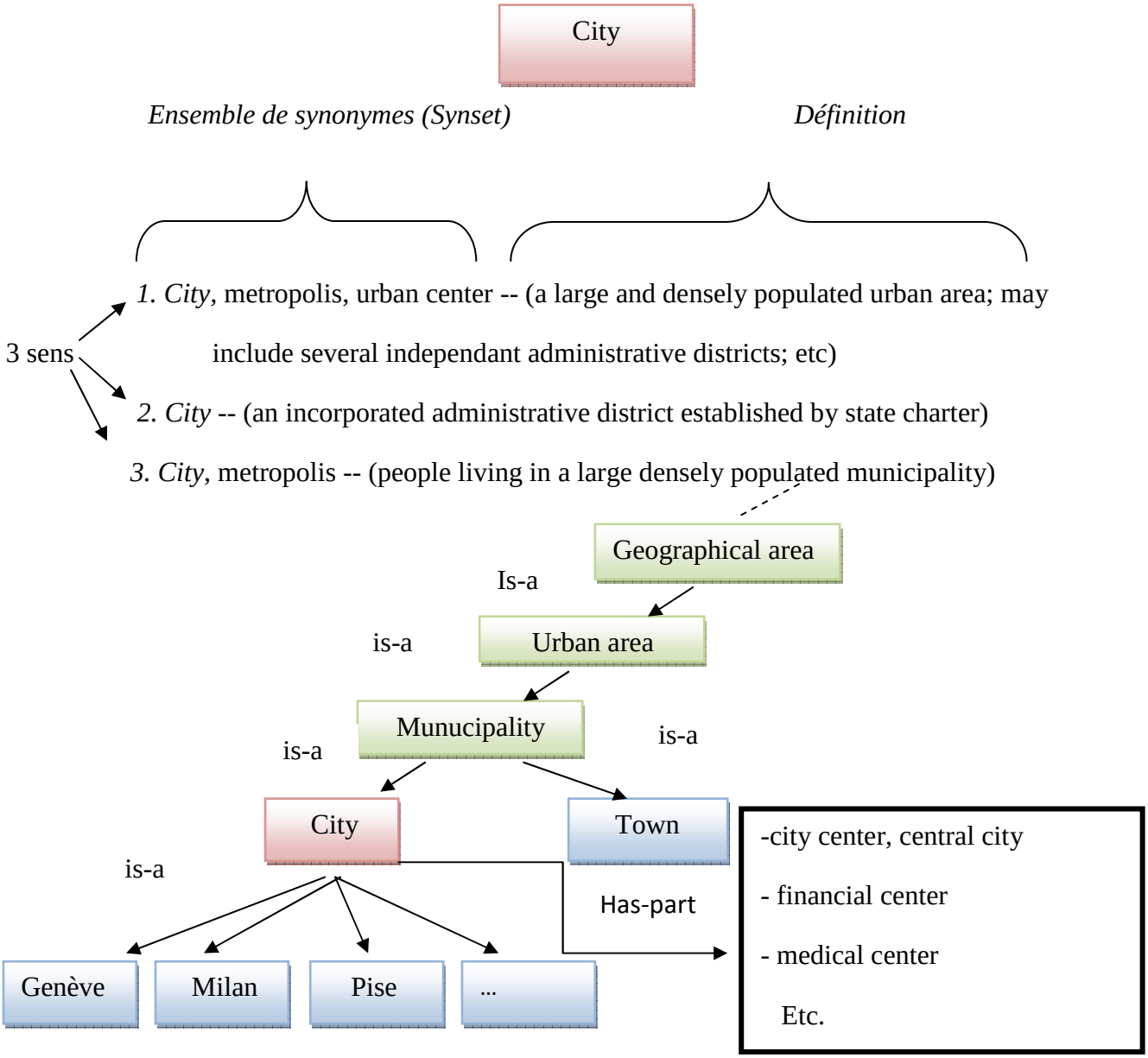


**Figure.1.1 : Exemple de contenu et de structure de WordNet.**

| Partie de discours | Nombre de mots | Nombre de synsets | Nombre de sens |
|---|---|---|---|
| *Noms* | 117097 | 81426 | 145104 |
| *Verbes* | 11488 | 13650 | 24890 |
| *Adjectifs* | 22141 | 18877 | 31302 |
| *Adverbes* | 4601 | 3644 | 5720 |
| *Total* | 155327 | 117597 | 207016 |

**Tableau 1.1 : Nombre de mots, de synsets, et de sens de mots dans Wordnet 2.1.**

Ce tableau montre bien la largeur de couverture de WordNet.

## 1.3. Répartition de la polysémie dans WordNet :

Les statistiques sur la répartition des mots polysémiques et monosémiques sont données dans le tableau suivant.

| Partie de discours | Nombre de mots et de sens monosémiques | Nombre de mots polysémiques | Nombre de sens polysémiques |
|---|---|---|---|
| *Noms* | 101321 | 15776 | 43783 |
| *Verbes* | 6261 | 5227 | 18629 |
| *Adjectifs* | 16889 | 5252 | 14413 |
| *Adverbes* | 3850 | 751 | 1870 |
| *Total* | 128321 | 27006 | 78695 |

**Tableau 2.2 : Répartition de la polysémie et de la monosémie dans WordNet 2.1.**

## 1 .4. Les concepts dans WordNet :

Les concepts de ce réseau n'ont cependant pas vocation à sous-tendre un système de représentation des connaissances. Ils ont été appliqués en TALN dans de nombreux contextes, en particulier pour l'indexation sémantique de textes et à des fins de recherche documentaire.

## 1.5. Critiques de WordNet:

WordNet a été très utilisée en recherche d'information, cependant elle a été critiquée sur de nombreux points notamment pas [Baziz, 03]. Ces critiques peuvent être résumées en ces points :

- Elle est trop fine, donc pour un concept donné, elle peut présenter une multitude de sens différents, qui ne sont nécessairement pas tous courants. Ce qui complique la désambiguïsation.
- WordNet ne couvre pas la totalité des mots de la langue anglaise. En effet, certains mots pourtant courants ne sont pas reconnus, à l'exemple du mot « EU » contrairement à « USA » ou « UN »,
- l'ordre des sens retournés par WordNet n'est pas toujours celui attendu. Comme c'est le cas pour le mot « whale » : WordNet retourne le premier sens correspondant à une personne de corps volumineux, pour ensuite donner en deuxième position le vrai sens de « whale » (correspondant à une baleine).

# Annexe 2 :

# La plateforme de RI *Terrier 3.0*

Cette annexe est principalement consacrée à la présentation de Terrier (Version 3.0), une plateforme de RI de haute performance et évolutive qui permet le développement rapide et à grande échelle de nouvelle applications de recherche d'information.

## 2.1 Introduction :

Terrier[18], *TeRabyte RetrIEveR* a été développé par le département informatique de l'université de *Glasgow*. C'est un logiciel open source entièrement écrit en java. Il est utilisé avec succès pour la recherche Ad hoc, la recherche web et la recherche inter-langage dans des environnements centralisées et distribuées.

Terrier offre plusieurs modèles de pondération de documents et d'expansion de requêtes basé sur le Framework DFR (Divergence From Randomness)[19].Comme tout les moteurs de recherche Terrier possède les principales facettes suivantes :

***Indexation*** : Permet l'extraction des termes des différents documents du corpus (basic indexed unit).

***Recherche*** : Permet de générer des résultats aux requêtes formulées par les utilisateurs.

[18] http://www.terrier.org

[19] http://terrier.org/docs/v2.2.1/dfr_description.html

## 2.2. Installation de Terrier :

### 2.1.1 Préalablement :

Pour pouvoir utiliser Terrier il est nécessaire d'installer une JRE (1.5.0 ou plus). La JRE ou la JDK peuvent être téléchargé sur le site de Java[20].

### 2.2.2 Installation:

Après avoir de téléchargé une copie de Terrier version 3.0 sur la page d'accueil du projet Terrier[21], créer un nouveau répertoire et dézippé Terrier dans ce dernier.

## 2.3. La structure des répertoires de Terrier :

Terrier contient un ensemble de répertoires et ils sont structurés comme suit:

- bin\ : Contient les scripts nécessaires pour démarrer Terrier.
- doc\ : Contient la documentation relative à Terrier.
- ect\ : Contient les fichiers de configurations de Terrier.
- lib\ : Contient les classes compilées de Terrier et les différentes librairies externes utilisées par Terrier.
- share\ : contient la liste des mots vides (stop word list).
- scr \ : Contient le code source de Terrier.
- var/index : Contient les structures de données : fichier inverse, fichier lexicon, index direct, Document Index.
- var/results : Contient les résultats de la recherche.
- licenses/ : contient les informations sur la licence des différent composants inclus dans Terrier.

[20]http://www.oracle.com/technetwork/java/javase/downloads/index.html

[21] http://www.terrier.org

## 2.4. Les Applications de Terrier :

Terrier offre trois applications :

***Batch(TREC) Terrier***: permet l'indexation, la recherche et l'évaluation des résultats d'une collections TREC[22].

***Interactive Terrier:*** permet une recherche interactive en exécutants le script interactive_terrier[23]. C'est un moyen facile de tester Terrier.

***Desktop Terrier***: c'est une interface graphique pour la plateforme de RI Terrier.

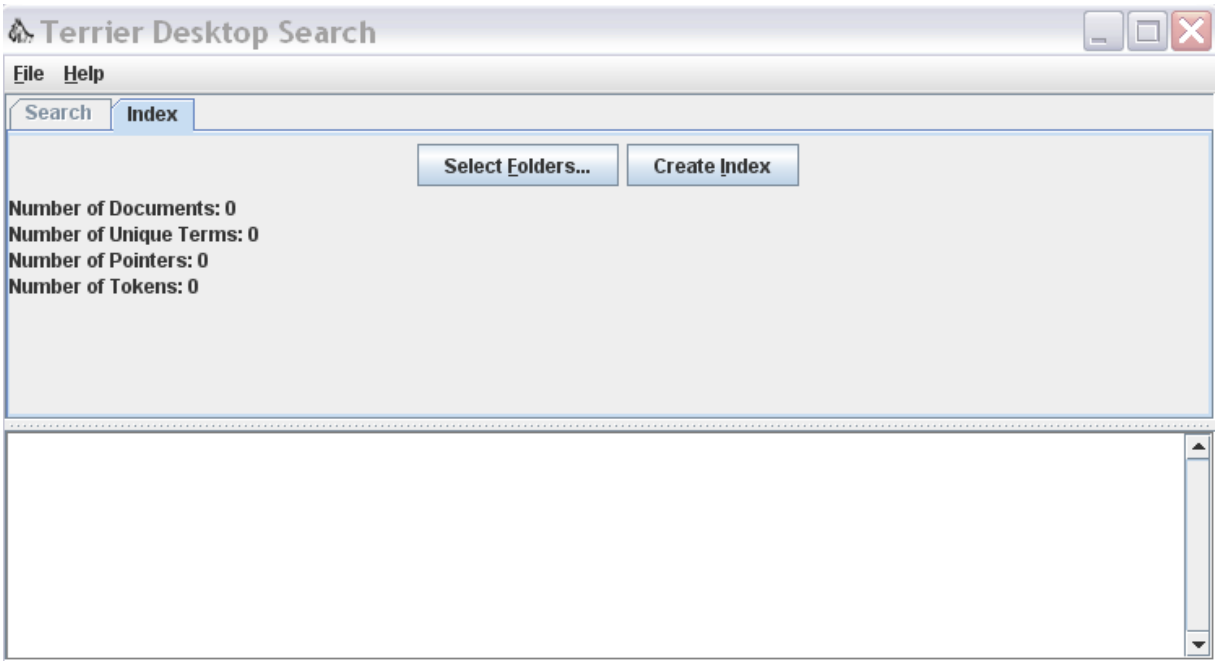


**Figure.1 : Présentation de l'interface Desktop Terrier.**

Autre que l'interface Desktop Terrier, Terrier propose une *interface Web* (Web-based interface) pour réaliser l'interactivité.

[22] http://trec.nist.gov/

[23] http://terrier.org/docs/v2.2.1/javadoc/uk/ac/gla/terrier/applications/InteractiveQuerying.html

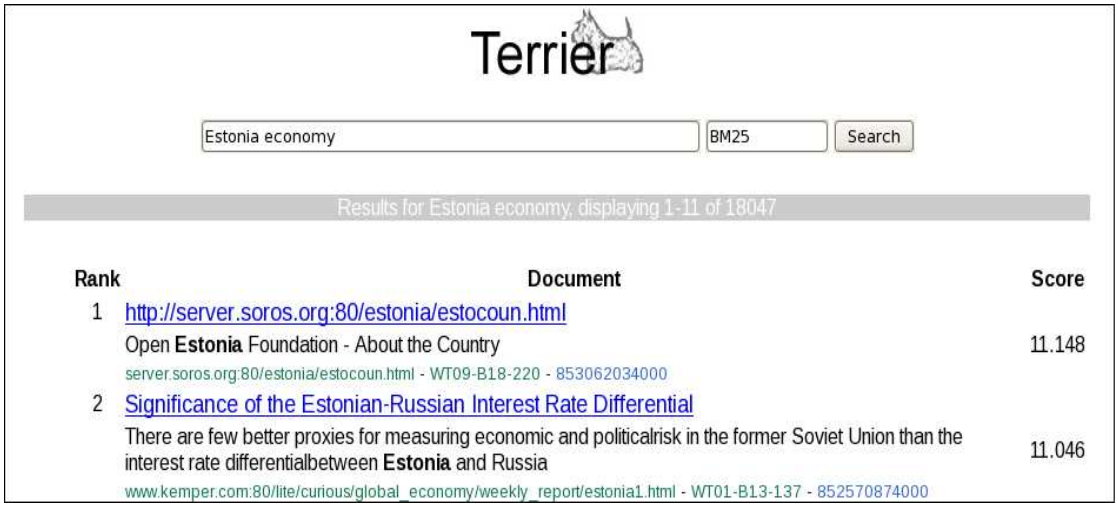


**Figure.2 : Présentation de l'interface Web de Terrier.**

## 2.5. L'API d'indexation de Terrier :

L'indexation dans Terrier est divisée en quatre procédures (étages) et à chaque étage des plug-ins (des classes java) peuvent être ajouté pour la personnalisation du Système. Les quatre étages sont :

1. Extraction de l'objet *Document* de la *Collection* qui est générer par l'ensemble des *Corpus* reçu en entrée par Terrier.
2. Parcourir chaque document de la collection pour en extraire les termes ainsi que les informations relatives.
3. Traitement des Termes extrait on utilisant *TermPipelines.*
4. La construction de l'index.

**La Figure.3** ci-dessous présente les différents étages du processus d'indexation dans Terrier.

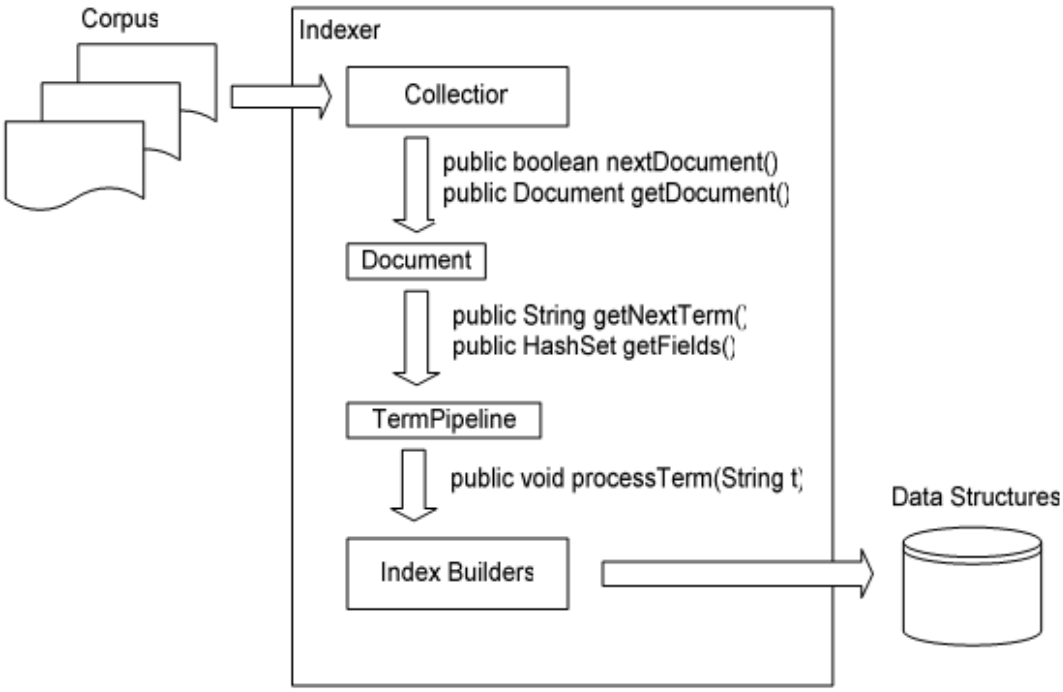


**Figure.3 : Présentation du processus d'indexation dans Terrier [Ounis, 10].**

Les composantes principales du processus d'indexation sont présentées dans les paragraphes suivant.

## 2.5. 1.Collection :

Cette composante englobe le concept le plus fondamental de l'indexation avec Terrier qui est la « *Collection* ». C'est un objet qui représente le corpus, c'est à dire un ensemble de documents. Collection[24] est une interface qui se trouve dans le package org.terrier.indexing. Utilisée généralement par Terrier pour intégrer de nouvelles sources de données (nouveaux corpus) et cela en ajoutant une nouvelle classe java qui implémente cette interface. Elle permet de parcourir un ensemble de document et de renvoyer un objet document en faisant appel à sa méthode nextDocument ( ).

[24] Pour plus d'information voir la Javadoc de Terrier

Terrier offre plusieurs classes qui implémentent cette interface tel que SimpleFileCollection, SimpleMedlineXMLCollection, SimpleXMLCollection, TRECCollection, TRECUTFCollection, WARC018Collection et WARC09Collection.

### 2.5. 2.Document :

C'est une interface qui se trouve dans le package org.terrier.indexing. Cette composante englobe le concept de Document. Les classes qui implémentent cette interface s'occupe de parcourir les documents et dont extraire les différents Termes. Terrier possède plusieurs parseurs de documents, par exemple : HTMLDocument, FileDocument, MSExcelDocument, ect.

### 2.5. 3. Term Pipeline :

C'est une interface qui se trouve dans le package org.terrier.terms. Cette composante reçoit l'ensemble des termes extrait des documents. Elle est considérer comme étant un pipeline qui traite et transforme chaque termes.

### 2.5. 4. Indexer :

Cette composante est chargée de la gestion du processus d'indexation. Entre autre la construction de l'index et de son écriture dans la structure de données approprié.

Terrier offre deux types d'indexeur : BasicIndexer, BlockIndexer.

### 2.5. 5. Les structures Index :

L'Index de Terrier consiste en quatre structures de données, en plus de quelques fichiers auxiliaires :

- Vocabulary/Lexicon (data.lex)
- Inverted Index (data.if)
- Document Index (data .docid)
- Direct Index (data.df)

**Le Tableau.1** ci-dessous permet de présenter le contenu de ces différentes structures index.

| Index Structure | Contents |
|---|---|
| Lexicon | Term<br>Term id<br>Document Frequency<br>Term Frequency<br>Byte offset in inverted file<br>Bit offset in inverted file |
| Inverted Index | Document id<br>Term Frequency<br>Fields (# of fields bits)<br>Block Frequency<br>[Block id ] |
| Document Index | Document id<br>Document Length<br>Document Number<br>Byte offset in direct file<br>Bit offset in direct le file |
| Direct Index | Term id<br>Term frequency<br>Fields (# of fields bits)<br>Block frequency<br>[Block id ] |

**Tableau.1 : Présentation des différentes structures Index.**

L'indexation en Terrier peut être configurée on changeant les propriétés approprié dans le fichier *etc\terrier.properties*. Chaque étage cité auparavant possède ces propres propriétés.

## 2.6. L'API de recherche de Terrier :

Terrier utilise trois composantes principales pour la recherche : *Query*, *Manager*, *Matching*.

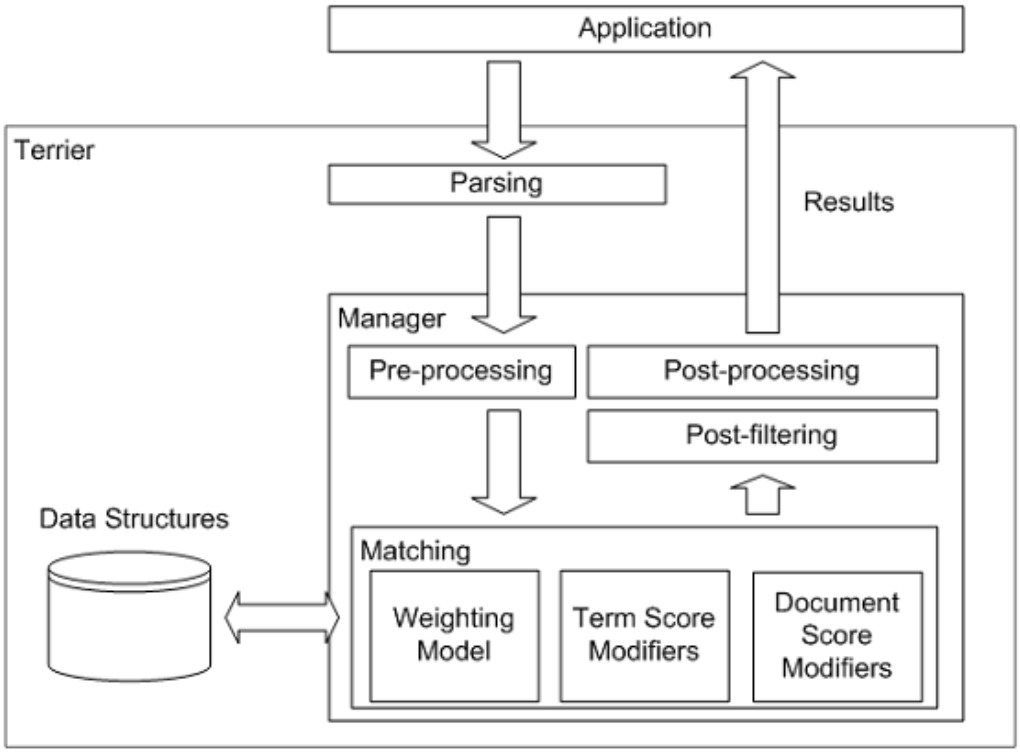


**Figure.4 : Présentation du processus de recherche de Terrier [Ounis, 10].**

### 2.6.1. Query :

C'est l'entrée que l'application offre à Terrier (voir la **Figure.4**). Le Module Maching est responsable de déterminer quel document correspond à une requête spécifique et attribut un score au document qui respect la requête.

Query est une classe abstraite, qui se trouve dans le *pakage org.terrier.querying.parser* et qui modélise les requêtes. Elle consiste soit en un s*ub-queries* ou bien un *query terms*. Un objet Query est créé pour chaque requête.

Terrier offre un support pour différents types de requête :

- *SingleTermQuery* : c'est le modèle de requête avec un seul terme.
- *MultiTermQuery*: consiste en un modèle de requêtes qui contiennent plus d'un terme.
- *FieldQuery* : c'est le modèle de requête qualifié par un champ (field)

### 2.6.2. Manager:

C'est le module qui est chargé de la gestion de la recherche. Dans un premier niveau il lit chaque requête ont utilisant le parseur approprié. Puis il créait un second niveau de gestion associé à chaque requête on récupérant l'ensemble des paramètres nécessaire et en spécifiant un modèle de pondération. Puis il créer un fichier résultat (result file) ou il associe à chaque requête les documents qui lui sont pertinents. Le résultat de chaque requête est donné par le second niveau de gestion.

Le second niveau de gestion est responsable de l'ordonnancement et de la coordination des opérations principales de haut niveau sur une seule requête. Ces opérations sont: Pre-processing, Matching, Post-processing, Post-filtering.

### 2.6.3. Matching :

Le composant Matching est responsable de déterminer qui des documents correspondent à la requête spécifié et donne un score au document qui vérifie la requête. Il utilise le *Weighting Models* pour assigner un score à chaque mot de la requête dans le document. Terrier contient un nombre important de modèle de pondération qui offre des performances robustes.

- *Document Score Modifiers* : Modifie le score des documents en fonction de la requête.
- *Term Score Modifiers* : Modifie le score des documents en fonction de la position des termes.

## 2.7. Modification Terrier :

L'une des caractéristiques principal de Terrier est sont extensibilité, il permet la modification du code source pour intégrer tout changement nécessaire. Pour que toutes modification soit prise en compte il est nécessaire de *recompilé* tout le code source de Terrier.

## 2.8. Utilisation de Batch(TREC) Terrier :

Dans cette section nous décrivant l'utilisation de Batch(TREC) Terrier pour l'indexation, la recherche et l'évaluation sur le système d'exploitation Windows XP.

### 2.8.1. Indexation:

1. Allez au répertoire où Terrier est installé on utilisant la commande *cd* :

    > > cd terrier-3.0

2. Initialisation de Terrier pour l'indexation d'une nouvelle collection TREC.

>> ***.\bin\trec_setup*** <le chemin absolu du répertoire contenant les documents à indexer>

3. Maintenant vous pouvez indexer la collection TREC.

**>>*.\bin\trec_terrier –i***

***Remaque :***

Pour indexer une Collection autre qu'une Collection TREC il est nécessaire d'apporter quelques modifications au fichier *terrier.properties.*

### 2.8.2. Recherche et Evaluation :

Avant toute recherche ou évaluation il est nécessaire de réaliser les trois étapes suivantes :

1. Spécification du fichier de jugement de pertinence dans le fichier etc\trec.qrels.

```
#add the qrels files to use for evaluation
/home/user/collection/Info/en.qrels
```

**Figure.5 : Exemple d'un fichier trec.qrels.**

2. Spécification des fichiers contenant les requêtes (topics file) dans le fichier etc\ trec.topics.list.

```
#add the topic files to use for querying
/home/user/collection/Info/en.topics.list
```

**Figure.6: Exemple d'un fichier trec.topics.list.**

3. Spécification du modèle de pondération (ex. TF_IDF) à utiliser dans le fichier etc/trec.models.

Après que ses étapes soient faites la recherche ou l'évaluation peuvent être lancé sur Terrier.

4. Pour lancer la recherche dans Terrier il suffit d'exécuter la commande :

***.\bin\trec_terrier -r***

5. Les résultats de la recherche peuvent être évolués en exécutant la commande :

***.\bin\trec_terrier –e***

## 2.9. Comparaison des plateformes de RI :

Dans le tableau qui suit, nous présentons une comparaison entre les plateformes de RI: Lemmur, Lucence, Terrier.

| | Lemur | Lucene | Terrier |
|---|---|---|---|
| Indexing | ▪ Claims can index up to terabytes of data<br>▪ **Incremental indexing** | ▪ Can **index** over **20MB/minute** on a home machine<br>▪ **small RAM requirements** -- only 1MB heap<br>▪ **index size about 20% -30%** the size of text indexed **(400GB➔ 80GB)**<br>▪ **Nutch** supports distributed indexing<br>▪ **Incremental indexing** | ▪ Some numbers:<br>**size of files to index:**<br>400 GB<br>**resulting size of index files:**<br>17 GB ➔ 4% of actual text<br>**time to build :**<br>3 days (2 processors)<br>**time to retrieve:**<br>4 sec/query (8 processors)<br>▪ Supports distributed indexing<br>▪ Does **not support incremental indexing** |
| Retrieval Models | ▪ KL-divergence<br>▪ Vector space<br>▪ Okapi BM25<br>▪ Language Model<br>▪ TF-IDF | ▪ VSM<br>▪ Boolean retrieval model | ▪ 126 Divergence From Randomness (DFR) models<br>▪ Okapi BM25<br>▪ Language modeling<br>▪ TF-IDF |
| Prog. Lang | C++ | Java | Java |

**Tableau.2 : Comparaison des plateformes de RI : Lemur, Lucence, Terrier.**

# Annexe 3 :

# Stanford POS Tagger

*Stanford Part-of-Speech Tagger* est un étiqueteur morpho-syntaxique et un lemmatiseur développé par *The Stanford Natural Language Processing Group*, il est distribué librement à des fins d'évaluation, de recherche ou d'enseignement. Pour l'étiquetage, il implémente une méthode probabiliste (arbres de décision) nécessitant une phase d'entraînement ; il est donc possible de développer une version spécifique selon la langue pour laquelle on souhaite l'utiliser. Une version dédiée à l'anglais est ainsi disponible sur la page d'accueil du Stanford *POS Tagger*; seul le fichier de paramètres varie : le moteur probabiliste reste inchangé. La Liste des catégories grammaticales utilisées par Stanford POS Tagger pour l'anglais est présentée dans le **tableau 3.1** ci-dessous. Cette Liste correspond aux jeux d'étiquetage *Treebank Tag set*.

| **Catégorie** | **Signification** |
|---|---|
| CC | conjonction |
| CD | nombre |
| DT | déterminant (*the, a, all, and, both, etc.)* |
| EX | *there* |
| FW | mot ou expression étrangère |
| IN | préposition (*across, after, as, for, in, etc.*) |
| JJ | adjectif |
| LS | référence |
| MD | auxiliaires (*can, should, may, would, will, might*) |
| NN | Nom |
| NNP | Nom propre |
| NNPG* | nom de groupe (société, association, etc.) |
| NNPL* | Nom de lieu |

| | |
|---|---|
| NNP’ | Nom de personne |
| NNS | Pluriels |
| PDT | *all, both, that, this* |
| POS | *'s* possessif |
| PRP | Pronom personnel |
| PRP$ | Pronom possessif |
| RB | adverbe |
| RBR | adverbe de comparaison (*better, etc*.) |
| SPUNC’ | ponctuation forte |
| TO | infinitif to |
| UH | interjection |
| UNK’ | mot inconnu |
| VB | Verbe |
| VBD | Verbe au passé |
| VBG | participe présent |
| VBN | participe passé |
| VBP | auxiliaires (*be, do, have, is, does, has*) |
| WDT | *that, whatever, which* |
| WP | *whadya, what, who, whom, adjectifs interrogatifs relatifs* |
| WP$ | *whose* |
| WPUNC’ | ponctuation faible |
| WRB | *how, when, whence, whenever, where, whereby, wherever, why* |
| ZTRM’ | fin de phrase |

**Tableau 3.1 : Liste des catégories grammaticales.**