

**Faculté des Sciences Exactes et de l'Informatique**  
**Département de Mathématiques et d'Informatique**  
**Filière : Informatique**

MEMOIRE DE FIN D'ETUDES  
Pour l'Obtention du Diplôme de Master en Informatique  
Option : Ingénierie des Systèmes d'Information

THEME :

**Classification des documents textuels par  
la détection des entités nommées**

**Etudiant:**

Gouaich Hassane

**Devant le jury :**

Mme HAMAMI Dalila

Président.

Mme MAGHNI Sandid Zoulikha

Encadreur.

Mr MIDOUN.M

Examineur.

## **Résumé :**

Dans ce document, je présente les travaux de recherche que j'ai menés, particulièrement les données textuelles, le développement d'outils d'analyse et de traitement automatique des textes, notamment la classification automatique de textes, est devenu indispensable, pour assister les utilisateurs, de ces collections de documents, à explorer et à répertorier toutes ces immenses banques de données textuelles.

Le traitement des entités nommées s'oriente désormais vers de nouvelles perspectives avec, entre autres, la désambiguïsation et une annotation enrichie de ces unités. La tâche de reconnaissance et de catégorisation des noms de personnes, de lieux, d'organisations, etc. apparaît en effet comme fondamentale pour diverses applications participant de l'analyse de contenu et nombreux sont les travaux se consacrant à sa mise en œuvre, obtenant des résultats plus qu'honorables.

Ces travaux, combinant approche reconnaissance d'entités nommées, rendent compte de la possibilité d'une double annotation de corpus (catégories sémantique et catégories syntaxique) et d'une désambiguïsation des entités nommées et d'améliorer les performances et l'efficacité du modèle de classification.

Outre, dans ce travail nous avons adapté l'algorithme K-NN sur les entités nommées et les corpus complets pour améliorer la classification des documents textuels, évaluer ces performances et comparer entre les résultats obtenus afin de mieux choisir la meilleure classification.

**Mots clés :** Recherche d'information, Le processus de RI, détection des entités nommées, Identification, Classification automatique et manuelle, Indexation des documents.

# *Dédicaces*

**J**e dédis ce modeste travail, qui est le fruit récolté après tant d'années d'efforts :

A mes très chers parents qui m'ont soutenu & encouragés durant mes études, Eux qui m'ont toujours apporté leur soutien moral et matériel depuis mon premier jour à l'université.

A mes très chers frères Aucune dédicace ne serait exprimer assez, je vous dirais tout simplement un grand merci.

A mes très chers amis En témoignage de l'amitié sincère qui nous a liées et des bons moments passés ensemble.

# *Remerciement*

**A**vant tout, Je remercie mon dieu qui m'a donné la patience et la volonté pour terminer mon travail.

Je souhaiterais remercier le département informatique de m'avoir appris à aimer le monde numérique et digital.

Je remercie également mes enseignants pour la qualité de l'enseignement qu'ils m'ont prodigué au cours de ces 5 années passées à l'université ABD ELHAMID BEN BADIS de Mostaganem. Je remercie tout particulièrement mon encadreur Mme Maghni Sandid Zoulikha qui m'a laissé une large part d'autonomie dans ce travail tout en m'aiguillant sur des pistes de réflexions riches et porteuses, Je souhaiterais aussi remercier tout le corps administratif.

Je remercie enfin l'ensemble des mes proches qui m'ont aidé et motivé durant ce cursus, je les remercie pour l'aide qu'ils m'ont apporté dans la réalisation de ce travail. Je veux remercier les personnes optimistes que j'ai pu croiser.

Merci à vous tous !

# *Table de matière*

Introduction générale .....	1
-----------------------------	---

## **Chapitre I : Recherche d'information (RI)**

1-Introduction .....	3
2- La recherche d'information .....	3
2-1- Définitions .....	3
2-2- Concepts de base de la RI .....	3
2-3- Les modèle de recherche d'information .....	5
2-3-1 Le modèle booléen .....	5
2-3-2- Le modèle vectoriel .....	6
2-3-3- Modèle probabiliste .....	7
3- Système de recherche d'information .....	8
3-1- Définition .....	8
3-2-Le processus de Recherche d'Information .....	8
3-2-1- Le processus d'indexation .....	9
4- Conclusion .....	10

## **Chapitre II : La Détection Et Les Entités Nommées**

1-Introduction .....	12
2- Définition .....	12
3- Les formes des entités nommées .....	12
3-1-Les entités nommées simples .....	12
3-2-Les entités nommées composées .....	12
4- Quelques problématiques liées aux entités nommées .....	12
5- Les typologies d'Entités Nommées .....	13
5-1- Noms propres de personnes, lieux et organisations .....	13
5-2- Expressions de temps, adresses et montants .....	13
5-3-Produits, marques, fonctions .....	13
6- Annotation et évaluation des entités nommées .....	13
6-1- Annotation de corpus .....	13
6-2- Métriques d'évaluation .....	14
6-3-Proposition de définition des entités nommées .....	15
7- Approches pour la reconnaissance d'entités nommées .....	15
7-1- Les approches orientées connaissances .....	15

7-2- Les approches orientées données .....	16
8- Travaux sur la catégorisation des EN .....	16
8-1- Catégorisation sémantique (référentielle) .....	17
8-2- Catégorisation syntaxique (graphique) .....	17
9-Détection d'entités nommées .....	17
10-Conclusion .....	17

### **Chapitre III : Classification des Documents Textuels**

1- introduction .....	19
2- Définition de la classification .....	19
3- Les méthodes de classification automatique .....	19
3-1- Classification Supervisé .....	19
3-1-1-Les étapes de classification .....	19
3-2- Classification non Supervisé .....	20
4- Différents modèles de classifieur .....	20
4-1- Machines à support de vecteurs (SVM) .....	21
4-2- Réseau neuronaux .....	21
4-3- Arbres de décision .....	21
4-4- Le Boosting .....	21
4-5- Critères d'évaluation des classificateurs .....	21
4-5-1- Matrice de contingence .....	21
4-5-2- Le rappel .....	22
4-5-3- La précision .....	22
4-5-4- La F-mesure .....	22
5-Classification de textes .....	22
6- Les étapes de représentation .....	22
6-1- Représentation de textes .....	23
6-1-1- Tokenisation .....	23
6-1-2- Elimination des majuscules .....	23
6-1-3- Elimination des mots vides .....	23
6-1-4- Lemmatisation / racinisation .....	23
7- Conclusion .....	23

### **Chapitre IV: Conception et Réalisation**

#### **Partie I**

1-Introduction .....	25
2- Architecture générale du système .....	25

2-1- Prétraitements (Segmentation) .....	26
2-2- Segment dans BDD (stockage) .....	29
2-3- Transformation des mots en synsets .....	29
2-4- Représentation conceptuelle .....	30
2-5- Calcul de similarité et création des classes .....	31
2-5-1 Calcul des poids .....	31
2-5-2 Similarité entre termes .....	33
2-6- Sélection d'algorithme d'apprentissage KNN .....	34
3- Etude sur les entités nommées à partir des corpus .....	36
<b>Partie II</b>	
4- Environnement et outils de développement .....	37
4-1- Langage JAVA .....	38
4-2- Environnement de développement .....	38
4-3- WampServer.....	38
4-4- WordNet .....	39
4-5- Corpus utilisé .....	40
5- Présentation de quelques interfaces de notre application .....	40
5-1- L'interface principale .....	40
5-2- Prétraitements effectués sur les corpus d'apprentissage et de test .....	41
5-3- Recherche d'information .....	42
5-4- Traitement WordNet .....	42
5-5- Classification des corpus .....	43
5-6- Classification des entités nommées .....	44
5-7- Evaluation des performances .....	45
5-7-1- Mesures de classification des corpus .....	45
5-7-2- Mesures de classification des entités nommées .....	46
6- Conclusion .....	48
Conclusion générale .....	49
Références .....	50

## *Liste des figures*

<b>Figure N°</b>	<b>Titre de la figure</b>	<b>Pages</b>
<b>Figure I.1</b>	Fonctionnement de la RI	5
<b>Figure I.2</b>	requêtes de modèle booléennes	6
<b>Figure I.3</b>	système de recherche d'information	8
<b>Figure I.4</b>	Processus de recherche d'information	9
<b>Figure I.5</b>	Indexation d'un document	9
<b>Figure II.1</b>	Eléments d'un processus d'annotation	14
<b>Figure II.2</b>	Architecture générale de Némésis	16
<b>Figure IV.1</b>	Architecture de notre travail	25
<b>Figure IV.2</b>	Prétraitement et représentation des documents	26
<b>Figure IV.3</b>	Liste des mots vides	27
<b>Figure IV.4</b>	Les tableaux des corpus indexés	29
<b>Figure IV.5</b>	Combinatoire des sens	29
<b>Figure IV.6</b>	Exemple d'un groupe de synset	30
<b>Figure IV.7</b>	Calcul de la similarité sémantique	34
<b>Figure IV.8</b>	Notre nuage de points de test	35
<b>Figure IV.9</b>	Le point blanc est une nouvelle entrée	36
<b>Figure IV.10</b>	Les 5 points les plus proches du point que l'on cherche à classer	36
<b>Figure IV.11</b>	Oepn Calais détecter les entités nommées à partir des corpus	37
<b>Figure IV.12</b>	NetBeans IDE 8.2	38
<b>Figure IV.13</b>	Exemple de bases de données	39
<b>Figure IV.14</b>	L'interface principale	40
<b>Figure IV.15</b>	fenêtre de la représentation liste des termes d'apprentissage	41
<b>Figure IV.16</b>	fenêtre de la représentation liste des termes du test	41
<b>Figure IV.17</b>	Recherche sémantique	42
<b>Figure IV.18</b>	Fenêtre calcul d'occurrence et Représentation des mots avec synset	42
<b>Figure IV.19</b>	Les similarités entre les documents	43
<b>Figure IV.20</b>	Fenêtre de similarités entre les documents	43
<b>Figure IV.21</b>	Résultat des mesures de classification des documents	43
<b>Figure IV.22</b>	Fenêtre de la représentation liste des entités nommées	44
<b>Figure IV.23</b>	Fenêtre calcul des Fréquences	44
<b>Figure IV.24</b>	Résultat de classification des entités nommées	45
<b>Figure IV.25</b>	Résultat de la Recherche	45
<b>Figure IV.26</b>	Choix des termes	46
<b>Figure IV.27</b>	Résultat des mesures de classification des corpus	46
<b>Figure IV.28</b>	Matrice de contingence	47
<b>Figure IV.29</b>	Résultat des mesures de classification des entités nommées	47



## *Liste des tableaux*

<b>Tableau N°</b>	<b>Titre du tableau</b>	<b>page</b>
<b>Table IV.1</b>	Représentation matricielle d'un corpus	31
<b>Table IV.2</b>	Les fréquences de tous les termes dans le document	32
<b>Table IV.3</b>	Fréquences des termes (tf).	32
<b>Table IV.4</b>	Fréquences des termes (idf).	33
<b>Table IV.5</b>	Le TF -IDF d'un terme dans un document	33
<b>Table IV.6</b>	Caractéristiques du nombre de mots et de concepts dans WordNet	39
<b>Table IV.7</b>	Répartition des documents du corpus utilisé	40
<b>Table IV.8</b>	Comparaison les résultats	47

## *Liste d'abréviations*

<b>Abréviation</b>	<b>Expression Complète</b>
<b>RI</b>	<b>R</b> echerche d' <b>I</b> nformation
<b>SRI</b>	<b>S</b> ystème de <b>R</b> echerche d' <b>I</b> nformation
<b>CT</b>	<b>C</b> atégorisation de <b>T</b> extes
<b>EN</b>	<b>E</b> ntités <b>N</b> ommées
<b>TAL</b>	<b>T</b> raitement <b>A</b> utomatiques des <b>L</b> angues
<b>REN</b>	<b>R</b> econnaissance des <b>E</b> ntités <b>N</b> ommées
<b>SVM</b>	<b>M</b> achines <b>V</b> ecteurs de <b>S</b> upport
<b>MMC</b>	<b>M</b> odelé de <b>M</b> arkov <b>C</b> ache
<b>MEM</b>	<b>M</b> aximum <b>E</b> ntropy <b>M</b> odel
<b>CRF</b>	<b>C</b> onditional <b>R</b> andom <b>F</b> iel
<b>BDD</b>	<b>B</b> ase <b>D</b> e <b>D</b> onnées
<b>TF</b>	<b>T</b> erm <b>F</b> requency
<b>IDF</b>	<b>I</b> nverse <b>D</b> ocument <b>F</b> requency
<b>KNN</b>	<b>K</b> - <b>N</b> earst <b>N</b> eighbors

## Introduction générale :

La recherche d'information est matérialisée par un ensemble d'outils dont l'objectif est de répondre à un besoin applicatif à partir d'une collection de données d'une manière automatique. On parle alors d'extraction d'information, L'extraction d'information (EI) qui est un sous-domaine du traitement automatique du langage naturel (TALN) consiste à extraire automatiquement, à partir de données non structurées, des informations structurées pertinentes pour une tâche particulière. Dans ce rapport, nous nous intéressons à l'une des sous-tâches de l'EI qui est la reconnaissance et la détection des entités nommées (ENs).

La détection des Entités Nommées (EN) est un élément essentiel à de nombreuses tâches de TAL, comme la recherche d'information ou la traduction automatique. Les entités nommées (EN), est une appellation générique pour les noms propres désignant entre autres des personnes, des lieux ou des organisations. Comme la plupart des unités lexicales considérées en dehors du contexte d'un énoncé, les EN sont polysémiques.

Le traitement des entités nommées s'articule en deux processus : identification ou reconnaissance de ces unités dans les textes tout d'abord, catégorisation ou typage selon des catégories sémantiques et des catégories syntaxique.

L'objectif de notre travail est de développer un processus de classification des documents et reconnaissance des entités nommées afin d'améliorer la qualité de la classification automatique de textes.

Nous avons décomposé notre mémoire en trois chapitres. Le premier chapitre vise à présenter la Recherche d'information qui décrit le problème de la Recherche d'information. Tel que on présente les concepts de base de la RI.

Le deuxième chapitre se base sur la détection et les entités nommées dans lequel nous présentons quelques notions de base liées au domaine d'extraction des **ENs**.

Dans le troisième chapitre, nous allons exposer la classification automatique des documents, plus en détail la catégorisation de textes.

Le quatrième chapitre a été consacré à l'utilisation des corpus complets (Apprentissage/test) et les entités nommées en les soumettant à un prétraitement puis nous avons utilisé la méthode K-NN pour les classifier afin d'évaluer les performances des différentes approches implémentées en présentant les résultats obtenus avec interprétation, aussi nous avons décrit les étapes d'implémentation, les outils utilisés ainsi que les résultats.

**Chapitre I:**  
**Recherche d'information (RI)**

## 1-Introduction :

La Recherche d'Information (**RI**) peut être définie comme une activité dont la finalité est de localiser et de délivrer un ensemble de documents à un utilisateur en fonction de son besoin en informations. Le défi est de pouvoir, parmi le volume important de documents disponibles, trouver ceux qui correspondent au mieux à l'attente de l'utilisateur.

L'opérationnalisation de la **RI** est réalisée par des outils informatiques appelés Systèmes de recherche d'Information (**SRI**), ces systèmes ont pour but de mettre en correspondance une représentation du besoin de l'utilisateur (requête) avec une représentation du contenu des documents (fiche ou enregistrement) au moyen d'une fonction de comparaison (ou de correspondance).

Ce chapitre est organisé en trois grandes parties : nous présentons les concepts de base de la **RI** et les différents modèles qui ont été proposés pour fournir un cadre théorique pour la modélisation du processus de **RI**. La deuxième partie nous décrit notamment le processus **RI**, à savoir les étapes d'indexation, d'interrogation, ainsi que Les techniques de reformulation des requêtes. La troisième partie nous présentons également la notion des profils utilisateur et nous donnons une classification des profils et de leurs utilisations en **RI**.

## 2- La recherche d'information :

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.

On peut aujourd'hui dire que la recherche d'information est un champ transdisciplinaire qui peut être étudié par plusieurs disciplines utilisant des approches qui devraient permettre de trouver des solutions pour améliorer son efficacité. [1]

### 2-1- Définitions :

Un système de recherche d'information (**SRI**) permet de retrouver les documents pertinents à une requête d'utilisateur, à partir d'une grande base de documents.

- Un document peut être un texte, un morceau de texte, une page Web, une image, une vidéo, etc.
- Une requête exprime le besoin d'information d'un utilisateur.
- La pertinence: Dans un document pertinent, l'utilisateur doit trouver les informations dont il a besoin. [1]

### 2-2- Concepts de base de la RI :

Afin d'évaluer l'apport de modèles conceptuels à la recherche d'information en amont d'un moteur de recherche général, deux études ont été menées successivement. L'une porte sur la reformulation de requêtes en exploitant les relations entre concepts, l'autre sur la représentation de documents sous forme d'un réseau de concepts puis d'un arbre de concepts. De cette constatation plusieurs concepts clés peuvent être définis, nous avons donc trouvé utile de les clarifier. [2]

- ❖ **Collection de documents** : la collection de documents (ou fond documentaire) constitue l'ensemble des informations exploitables et accessibles. Elle est constituée d'un ensemble de documents. Dans le cas général et pour un souci d'optimalité, la base constitue des représentations simplifiées mais suffisantes pour ces documents. Ces représentations sont étudiées de telles sortes que la gestion (ajout suppression d'un document) ou l'interrogation (recherche) de la base se font dans les meilleures conditions de coût.
- ❖ **Document** : Le document peut être directement pensé et créé sous forme numérique ou bien numérisé à partir de son support original. Dans l'optique de cette thèse nous entendrons par « document » document textuel numérique. Les analyses effectuées sont faites sur des collections de documents fréquemment utilisées par les chercheurs du domaine. De manière analogue à la requête, les documents des collections étudiées, doivent être « indexés » pour être traité par un **SRI**.
- ❖ **Requête** : la requête exprimée, il est nécessaire de lui donner une forme utilisable par un **SRI** pour entamer le processus de recherche. Divers types de langages d'interrogation sont proposés dans la littérature. Une requête est un ensemble de mots clés, mais elle peut être exprimée en langage naturel, booléen ou graphique. [3]
- ❖ **Besoin d'information** : le besoin d'information de l'utilisateur devient plus clair et plus précis au cours du processus de recherche. Trois types de besoin utilisateur ont été définis par :
  - Besoin vérificatif** : l'utilisateur cherche à vérifier le texte avec les données connues qu'il possède déjà. Il recherche donc une donnée particulière, et sait même souvent comment y accéder.
  - Besoin thématique connu**: l'utilisateur cherche à clarifier, à revoir ou à trouver de nouvelles informations dans un sujet et un domaine connus. Le besoin peut aussi s'exprimer de façon incomplète, c'est-à-dire que l'utilisateur n'énonce pas nécessairement tout ce qu'il sait dans sa requête mais seulement un sous-ensemble.
  - Besoin thématique inconnu**: cette fois, l'utilisateur cherche de nouveaux concepts ou des nouvelles relations en dehors des sujets ou des domaines qui lui sont familiers. Le besoin est intrinsèquement variable et est toujours exprimé de façon incomplète. [4]
- ❖ **Modèle de représentation** : un modèle de représentation est un processus d'appariement consiste à comparer la représentation de la requête avec les représentations des documents. Il calcule pour chaque couple requête document, une mesure appelée pertinence système qui reflète le degré de similarité entre la requête et le document considéré. Le processus d'appariement se base sur une fonction de similarité (ou de correspondance) noté RSV (Retrieval Status Value). Cette fonction est différente d'un modèle de recherche d'information à un autre. D'ailleurs un modèle de **RI** est caractérisé par sa fonction de similarité et son modèle d'indexation. [5]. La figure 1.1 présente Fonctionnement de la **RI**.

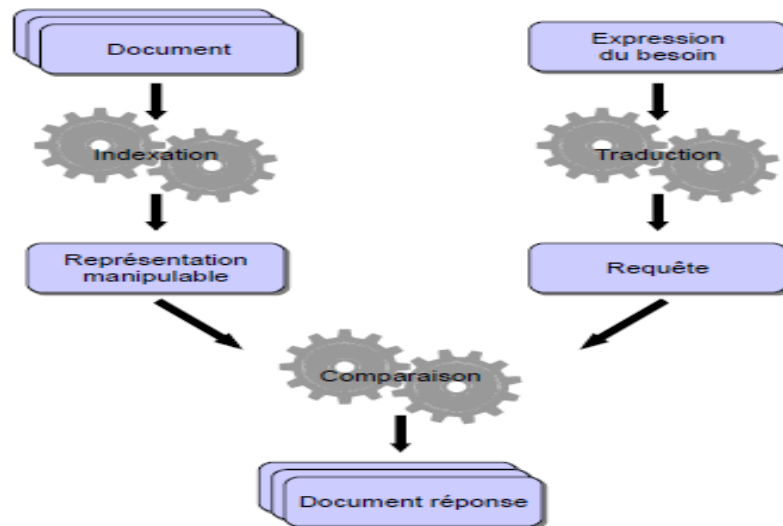


Figure I.1 : Fonctionnement de la RI

## 2-3- Les modèles de recherche d'information

Un modèle de **RI** a pour rôle de fournir une formalisation du processus de **RI**. C'est lui qui détermine le comportement clé d'un **SRI**. De nombreux modèles existent. Dans la suite nous présenterons d'abord le modèle booléen qui est historiquement un des premiers modèles étudiés et qui a servi de point de départ aux recherches du domaine puis le modèle vectoriel (approche algébrique) qui sert de base à notre modèle (approche basée sur les graphes) et enfin, le modèle probabiliste qui, bien qu'étant une approche différente de la notre permettra justement la comparaison avec notre approche.

Pour chacune des approches décrites, les deux points importants seront définis :

- la représentation et la comparaison.
- la représentation interne des documents et de la requête, les principaux modèles utilisent une représentation par mots-clés, et c'est dans la comparaison des représentations que chaque approche a sa propre manière de faire.

Nous présentons dans la suite les principaux modèles de RI : le modèle booléen, le modèle vectoriel et le modèle probabiliste.

### 2-3-1 Le modèle booléen

Le modèle booléen, [6], est historiquement le premier modèle de **RI**, et est basé sur la théorie des ensembles. Un document est représenté par une liste de termes (termes d'indexation). Une requête est représentée sous forme d'une équation logique. Les termes d'indexation sont reliés par des connecteurs logiques ET, OU et NON.

Le module de recherche mis en œuvre consiste à effectuer des opérations sur l'ensemble de documents afin de réaliser un appariement exact avec l'équation de la requête. L'appariement exact est basé sur la présence ou l'absence des termes de la requête dans les documents.

La décision binaire sur laquelle est basée la sélection d'un document ne permet pas d'ordonner les documents renvoyés à l'utilisateur selon un degré de pertinence. [7]

La figure 1.3 présente une requête de modèle booléenne.

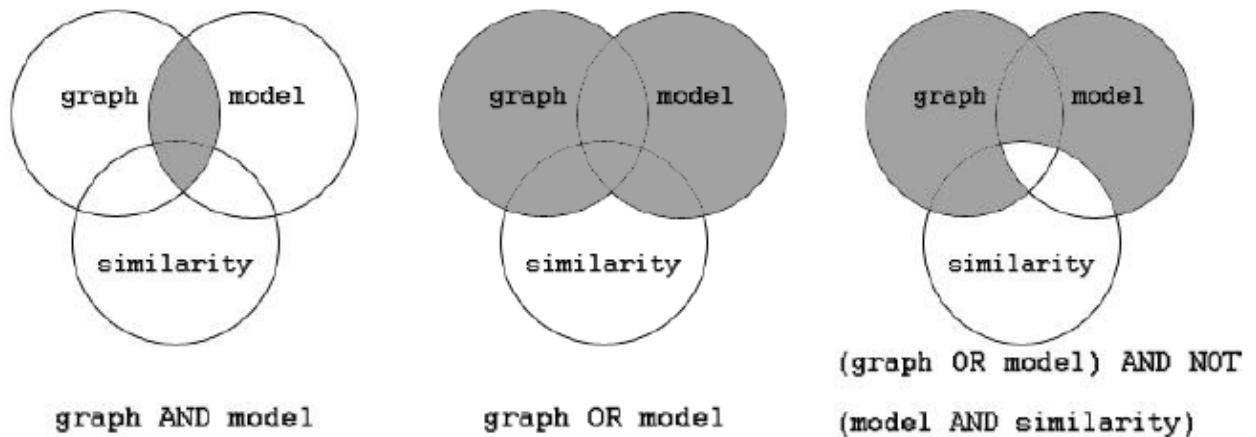


Figure I.2: requêtes de modèle booléennes [7]

Un document est représenté par une liste de termes, par exemple  $d = t_1, t_2, \dots, t_n$ . Une requête est représentée par une expression logique quelconque de termes utilisant les opérateurs AND, OR et NOT.

La correspondance RSV ( $d, q$ ) entre une requête  $q$  et un document  $d$  est déterminée de la façon suivante:

$RSV(d, t_i) = 1$  si  $t_i \in d$ ; 0 sinon.

$RSV(d, q_1 \text{ AND } q_2) = 1$  si  $RSV(d, q_1) = 1$  ET  $RSV(d, q_2) = 1$ ; 0 sinon.

$RSV(d, q_1 \text{ OR } q_2) = 1$  si  $RSV(d, q_1) = 1$  OU  $RSV(d, q_2) = 1$ ; 0 sinon.

$RSV(d, \text{NOT } q_1) = 1$  si  $RSV(d, q_1) = 0$ ; 0 sinon.

### 2-3-2- Le modèle vectoriel

C'est un modèle qui préconise la représentation des requêtes utilisateurs et des documents sous forme de vecteurs, dans l'espace engendré par tous les termes d'indexation, [8]. D'une manière formelle, les documents ( $D_j$ ) et les requêtes  $Q$  sont des vecteurs dans un espace vectoriel des termes d'indexation ( $t_1, t_2, \dots, t_T$ ) de dimension  $T$  et représentés comme suit :

$D_j = [d_{j1}, d_{j2}, \dots, d_{jT}]$ ,  $Q = [q_1, q_2, \dots, q_T]$

Où  $d_{ji}$  et  $q_i$  sont respectivement les poids des termes  $t_i$  dans le document  $D_j$  et la requête  $Q$ .

D'après ce modèle, le degré de pertinence d'un document relativement  $\mu$  à une requête est perçu comme le degré de corrélation entre les vecteurs associés. Ceci nécessite alors la spécification d'une fonction de calcul de similarité entre vecteurs mais également d'une fonction de pondération des termes. La plus répandue est celle de Sparck et Needham, [9], qui définit le poids d'un terme  $t_i$  dans un document  $d_j$  comme suit :

$d_{ij} = tf_{ji} * idf_i$

Où :  $tf_{ji}$  : est la fréquence relative du terme  $t_i$  dans le document  $D_j$ .

$idf_i$  : est l'inverse de la fréquence absolue du terme  $t_i$  dans la collection.

$idf_i = \log \frac{n}{n_i}$  avec  $n_i$  le nombre de documents contenant le terme  $t_i$  Et  $N$  est le nombre total de documents dans la collection. La fonction de similarité permet de mesurer la ressemblance des documents et de la requête. La mesure la plus répandue est celle du cosinus. [6]

$$RSV(Q, D_j) = \frac{\sum_{i=1}^T q_i d_{ji}}{\sqrt{\sum_{i=1}^T d_{ji}^2} \sqrt{\sum_{i=1}^T q_i^2}}$$



### 2-3-3- Modèle probabiliste

Ce modèle est fondé sur le calcul de la probabilité d'apparition d'un évènement, par exemple la probabilité de pertinence  $P(R)$  est formalisée au travers du concept d'expérimentation qui est le procédé par lequel l'observation est faite. L'ensemble des valeurs que peut prendre un fait constitue l'espace de départ. [10]

Pour  $P(R)$  l'espace de départ est {pertinent, non-pertinent}. Le modèle probabiliste considère que les termes d'indexation sont indépendants c'est-à-dire que leur probabilité d'apparition est la même avec ou sans la présence des autres termes. Sous cette hypothèse, on cherche à estimer la probabilité qu'un document soit pertinent par rapport à une requête. *PERT* et *NPERT* représentent respectivement la pertinence et la non-pertinence (ou de façon équivalente, l'ensemble de documents pertinents et l'ensemble de documents non pertinents). [11]

Le modèle probabiliste tente d'estimer la probabilité  $P(PERT/D)$  (resp.  $P(NPERT/D)$ ) qu'un document  $d$  appartienne à la classe des documents pertinents (resp. non pertinents). Autrement dit, on observe la pertinence ou le non pertinence sachant le document  $D$ . Seules la présence et l'absence de termes dans les documents et dans les requêtes sont considérées comme des caractéristiques observables. Autrement dit, les termes ne sont pas pondérés, mais prennent seulement les valeurs 0 (absent) ou 1 (présent).

On suppose que l'on a une requête fixe. On tente de déterminer les caractéristiques de  $R$  et  $NR$  pour cette requête donnée.

La correspondance RSV ( $d, q$ ) entre une requête  $q$  et un document  $d$  est déterminée de la façon suivante :

$$RSV(d, q) = O(D) = \frac{P(PERT/D, Q)}{P(NPERT /D, Q)}$$

Plus cette proportion est élevée pour un document, plus ce document est pertinent pour la requête. Cependant, les deux probabilités nécessaires ne sont pas directement calculables. En utilisant les règles de Bayes suivantes :

$$P(PERT|D, Q) = \frac{P(D, Q|PERT) * P(PERT)}{P(D, Q)}$$

$$P(NPERT|D, Q) = \frac{P(D, Q|NPERT) * P(NPERT)}{P(D, Q)}$$

$P(PERT)$  : est la probabilité qu'un document choisi au hasard soit pertinent.

$P(D|PERT, Q)$  : est la probabilité d'observer  $D$  sachant que l'on observe la pertinence en présence de  $Q$ .

$P(D|NPERT, Q)$  : est la probabilité d'observer  $D$  sachant que l'on observe la non-pertinence en présence de  $Q$ .

$P(D, Q)$  : est la probabilité conjointe du couple  $D, Q$ .

### 3- Système de recherche d'information :

#### 3-1- Définition :

Est un ensemble de programmes informatiques qui a pour but de sélectionner des informations pertinentes répondant à des besoins utilisateurs, exprimés sous forme de requêtes. Un système de filtrage peut être défini comme un processus qui permet d'extraire à partir d'un flot d'informations (News, e-mail, actualités journalières, etc.), celles qui sont susceptibles d'intéresser un utilisateur ou un groupe d'utilisateurs ayant des besoins en information relativement stables. [12]. La figure 1.3 présente un système de recherche d'information.

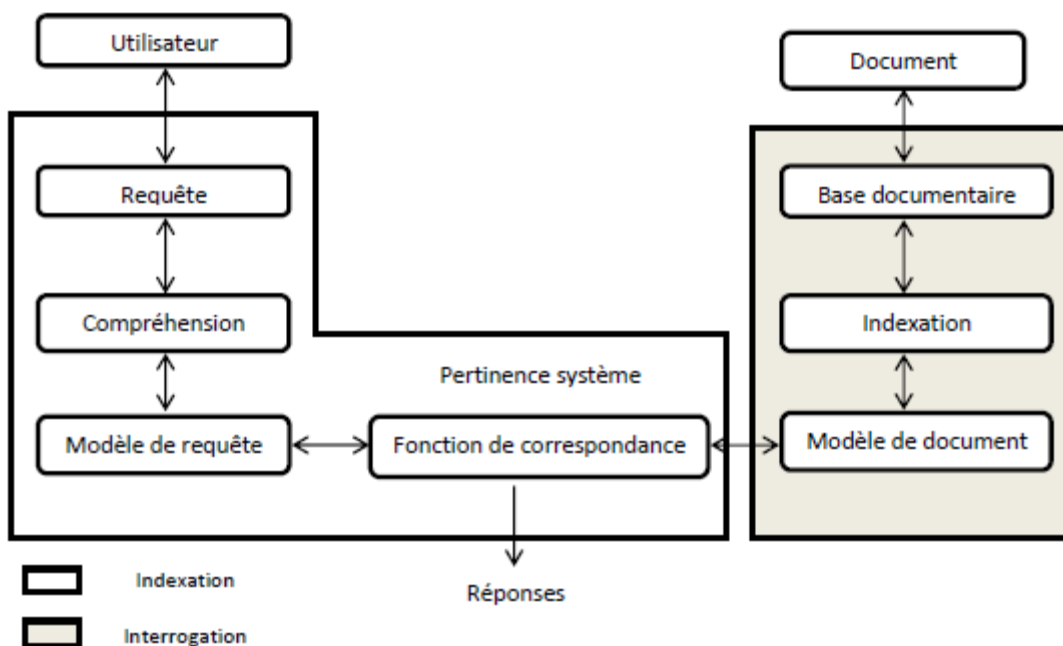
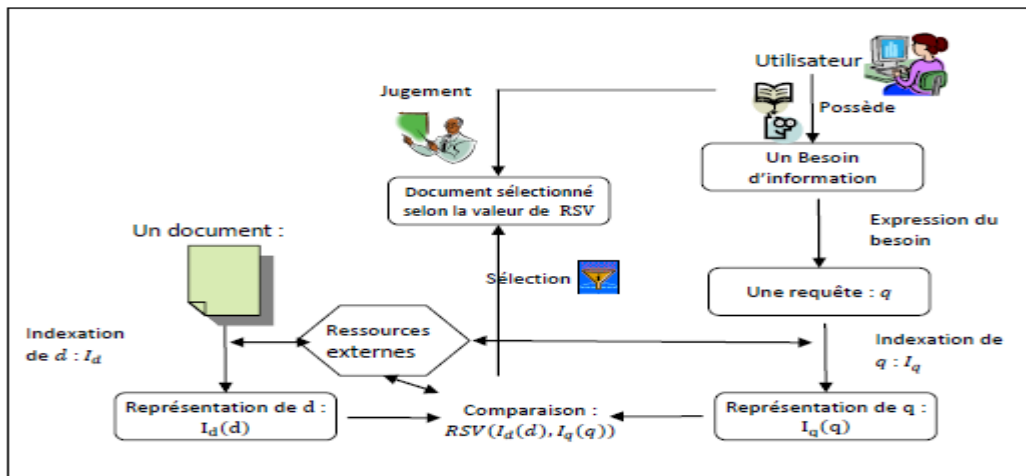


Figure I.3 : système de recherche d'information [12]

#### 3-2-Le processus de Recherche d'Information :

Le processus de Recherche d'Information a pour but la mise en relation des informations disponibles d'une part, et les besoins de l'utilisateur d'autre part. Ces besoins sont traduits de façon structurée par l'utilisateur sous forme de requêtes. La mise en relation des besoins utilisateurs et des informations est effectuée grâce à un Système de Recherche d'Information (SRI), dont le but est de retourner à l'utilisateur le maximum de documents pertinents par rapport à son besoin (et le minimum de documents non-pertinents). La notion de pertinence est difficile à automatiser, car elle est fortement subjective, c'est à dire dépendante de l'utilisateur. Le but du SRI est alors de faire correspondre au mieux la pertinence système avec la pertinence utilisateur. [13] La figure 1.4 présente les Processus de recherche d'information.



La figure I.4 : Processus de recherche d'information [13]

### 3-2-1- Le processus d'indexation

Dans un processus de **RI**, la requête et les documents à l'état brut sont difficilement exploitables. Afin de rendre la recherche possible, une étape primordiale s'avère nécessaire. Cette étape consiste à construire une représentation interne pour chaque document de la collection et de même pour la requête. Ces représentations seront utilisées ultérieurement (dans la fonction de correspondance) par le SRI. Pour ce faire des techniques et des modèles sont mis en œuvre.

Ces techniques permettent de décrire les documents et la requête par un ensemble de termes d'indexation ou de descripteurs. Ces descripteurs reflètent au mieux le contenu du document. Cette étape est appelée l'indexation. [14]

L'indexation se décompose en trois phases :

- L'extraction des termes du document.
- La sélection des termes discriminatifs pour un document.
- La pondération des termes.

La figure 1.5 présente l'indexation d'un document.

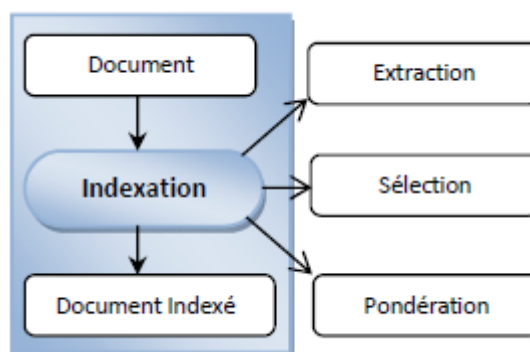


Figure I.5 : Indexation d'un document [14]

**4- Conclusion**

Dans ce chapitre nous avons présenté les principales notions et concepts de la recherche d'information, des systèmes de recherche d'information. A travers les différentes sections que nous avons présentées, nous concluons que la recherche d'information, s'attache à définir des modèles et des systèmes afin de faciliter l'accès à un ensemble de documents se trouvant dans des bases documentaire, il s'agit donc de définir la classification ainsi que les notions nécessaires pour l'entame de la suite de ce mémoire.

**Chapitre II :**  
**La Détection Et Les Entités**  
**Nommées**

## 1-Introduction :

Le concept d'entité nommée est apparu dans les années 90, les entités nommées (EN) se situent à un niveau intermédiaire entre le syntagme (groupe syntaxique) et l'énoncé.

Nous cherchons dans ce chapitre à situer cette La détection des entités nommées (EN) est un élément essentiel pour de nombreuses tâches de traitement automatique des langues (TAL), Nous allons essayer de toucher brièvement certaines des bases d'entités nommées, et les méthodes pour les détecter. Dans cette section, nous commençons à définir l'EN ensuite, nous présentons quelques notions de base liées au domaine d'extraction des ENs.

## 2- Définition :

Définies en tant qu'unités faisant référence à une entité unique et concrète et réalisées par des noms propres (noms de personnes, d'organisations, d'artefacts ou de lieux). Les expressions temporelles et les expressions de quantité sont généralement ajoutées à cette liste, moins en raison de leurs propriétés sémantiques que pour des considérations d'ordre pratique. [14]

## 3- Les formes des entités nommées :

Il y a deux formes d'EN ; Les ENs simples et les ENs composées. Chaque forme est traitée différemment.

### 3-1-Les entités nommées simples :

Une EN simple est une EN qui est composée d'un seul mot, comme les noms de lieu «Roma» et « Algérie » ou le nom de personne «Mohamed».

### 3-2-Les entités nommées composées :

Une EN composée est une EN qui est composée de deux ou plusieurs mots, comme par exemple le nom de personne 'Adam Smith' et le nom de lieu 'Afrique du Sud'.

## 4- Quelques problématiques liées aux entités nommées :

Comme nous l'avons vu, la notion d'entité nommée est mouvante et fait appel à de nombreux domaines : noms propres, modèles applicatifs en RI, D'une part, détecter et résoudre les entités nommées.

D'autre part, ces difficultés sont à chaque fois plus saillantes alors que le périmètre recouvert est continuellement étendu (adresses, produits, événements, etc.). Le besoin de clarifier cette notion et de disposer d'un module de traitement dédié à leur sujet se fait alors de plus en plus ressentir, par exemple pour les applications suivantes :

- **Indexation et recherche d'information** : les entités nommées détectées dans des documents peuvent permettre de construire des index que pourront exploiter les moteurs de recherche.
- **Annotation en rôles sémantiques** : dans le cadre d'un mécanisme de compréhension, déterminer les rôles (agent, patient, objet, instrument, lieu, destination, etc.) peut être conditionné par les types d'entités nommées reconnues.

- **Question-réponse** : le mécanisme par lequel une machine fournit une réponse à une question donnée peut nécessiter de résoudre des entités dans la question, afin de rechercher la réponse dans des bases de connaissances.
- **Résolution conjointe d'autres tâches de traitement automatique des langues (TAL)** : analyse morphosyntaxique ou syntaxique, reconnaissance de l'écriture et de la parole, résolution d'anaphores sont des tâches qui peuvent interagir avec la détection ou la reconnaissance des entités nommées. [15]

## 5- Les typologies d'Entités Nommées :

### 5-1- Noms propres de personnes, lieux et organisations :

Les besoins en recherche d'information se sont initialement focalisés sur le traitement des noms propres. Ces éléments sont relativement courants dans le langage: dans l'analyse d'une édition du journal Le Monde. [16]

Dans le cadre d'un processus TAL, ces éléments demandent à être reconnus le plus tôt possible, afin qu'il soit possible d'y appliquer des traitements particuliers.

### 5-2- Expressions de temps, adresses et montants :

Les représentations construites à partir d'énoncés peuvent tirer parti d'autres éléments qui désignent de manière plus complexe certains objets mentaux à manipuler. Dans ce contexte, après les noms propres, les expressions de temps ont été étudiées plus en détail, avec l'objectif d'extraire des informations à partir de textes (Ce midi, demain, l'année dernière, le 21 avril 2018 ...).

### 5-3-Produits, marques, fonctions :

Enfin, selon les domaines d'applications considérés, les entités nommées peuvent encore recouvrir diverses expressions linguistiques. Il est évident que les noms propres de personnes, lieux ou organisations ne couvrent pas tous les noms propres. Et même s'ils sont très majoritaires pour certains types de textes (journalistiques), on imagine aisément des situations dans lesquelles la reconnaissance aura intérêt à être étendue à d'autres types (produits, marques, événements, véhicules, etc.).

## 6- Annotation et évaluation des entités nommées :

### 6-1- Annotation de corpus :

L'annotation de corpus est une thématique très active qui fait l'objet de nombreux travaux. Effectivement, celle-ci peut être plus ou moins assistée, guidée, automatisée. De plus, comme le montre [17], ce travail nécessite une grande rigueur et beaucoup de préparation afin d'obtenir une annotation fiable. Dans l'essentiel, trois éléments paraissent indispensables :

- ✓ **Guide d'annotation** : détaille les expressions linguistiques à annoter, selon des critères qui doivent laisser aussi peu de latitude que possible à la personne qui réalisera l'annotation.
- ✓ **Outils d'annotation** : logiciels servant à annoter, dont les interfaces doivent faciliter, mais sans biaiser, le travail de l'annotateur, en incluant éventuellement une phase de pré-annotation automatique.

- ✓ **Mesures d'évaluation de la qualité des annotations :** tests prévus afin de confirmer la fiabilité (ou d'exhiber l'arbitraire) d'une annotation (accord inter-annotateurs) sur les parties annotées par plusieurs personnes (annotation croisée).

Le guide précise les possibilités et les limites pour annoter les entités de manière générale, pour n'importe quel texte. En premier lieu y sont indiqués les types d'entités à annoter. Également, lorsque des portions de textes peuvent recevoir plusieurs annotations, les structures possibles (types multiples) sont généralement contraintes et des directives sont données pour résoudre les cas problématiques. Enfin, si nécessaire, le guide d'annotation peut prévoir la mise en place d'un référentiel (base de données, de connaissances). Ceci est plus particulièrement utile lorsqu'il s'agit de résoudre les entités nommées, ce qui nécessite de faire pointer les entités vers des objets du monde réel référencés. Notons que le processus d'annotation est défini indépendamment de l'annotateur (humain ou machine). La figure 2.2 donne un aperçu des éléments en jeu dans ce processus.

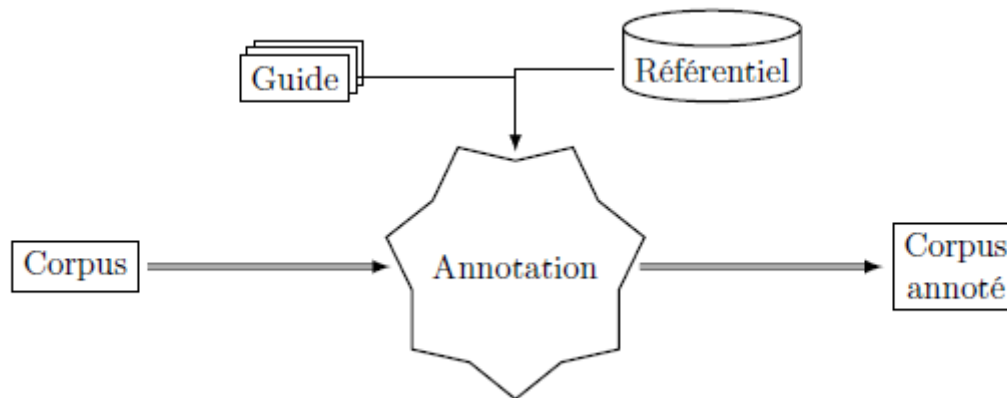


Figure II.1 : Éléments d'un processus d'annotation. [17]

## 6-2- Métriques d'évaluation :

Le rappel, la précision et la F-mesure sont des mesures largement utilisées dans les évaluations. La précision est le pourcentage des résultats corrects parmi les résultats obtenus [18], Le rappel est le pourcentage des résultats corrects parmi les résultats qu'on doit trouver.

La F-mesure est la combinaison de la précision et du rappel et leur pondération. La formule de la F-mesure est:

$$F\text{-mesure} = \frac{2 * (\text{précision} * \text{rappel})}{(\text{précision} + \text{rappel})}$$

Pour le domaine de l'extraction des ENs, les taux de la précision et du rappel sont calculés selon les formules suivantes :

$$\text{Précision} = \frac{\text{Nombre d'ENs correctement reconnues}}{\text{Nombre d'ENs reconnues}}$$

$$\text{Rappel} = \frac{\text{Nombre d'ENs correctement reconnues}}{\text{Nombre d'ENs dans le corpus}}$$



### 6-3-Proposition de définition des entités nommées :

De manière schématique, nous faisons l'hypothèse qu'entre le monde du langage et celui des représentations mentales, la reconnaissance des entités nommées est une interface qui associe des référents à des expressions linguistiques, avant de déterminer les relations logiques sur ou entre ces éléments donnent sens aux énoncés. Plus précisément, voici la formulation de ces deux propriétés que nous proposons d'associer aux entités nommées résolues :

- **Stabilité** : une entité nommée résolue désigne de manière rigide un référent, cette désignation n'évolue pas au long de l'énonciation et ne résulte pas d'inférences logiques.
- **Opérabilité** : les entités nommées résolues ne peuvent à elles seules former des propositions et ont vocation à prendre part à des opérations logiques (prédication, quantification, etc.)

### 7- Approches pour la reconnaissance d'entités nommées :

Nous utilisons des outils et ressources classiques en **TAL** pour fabriquer un système visant à reconnaître automatiquement les entités nommées. Comme exposé précédemment, pour les ressources, nous utilisons en particulier des lexiques, des transducteurs et des corpus.

#### 7-1- Les approches orientées connaissances :

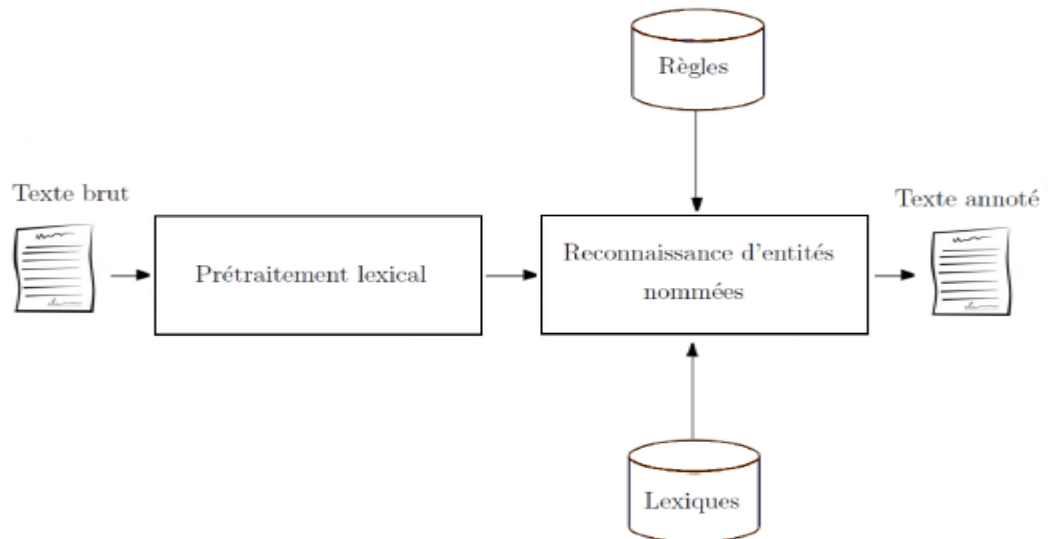
Les systèmes de **REN** orientés connaissances sont fondés sur des lexiques (des dictionnaires de noms propres de personnes, de pays, de villes, d'organisations...) et sur de règles produites manuellement par des experts. Les lexiques peuvent être formalisés à l'aide des règles pour créer des automates ou des grammaires génératives permettant de modéliser les contextes d'apparition des entités. Les approches orientées connaissances sont donc très dépendantes des lexiques.

Les lexiques ne sont pas exhaustifs, puisqu'il est impossible d'établir une liste de toutes les entités nommées à cause de la caractéristique dynamique des langues qui fait que des nouveaux noms propres apparaît tous les jours. La mise à jour et l'entretien des dictionnaires deviennent alors laborieux et peu efficaces. [19]

Voici quelques illustrations de preuves internes :

- ❖ **Noms propres**: le mot commence par une majuscule ('Pompidou').
- ❖ **Personnes**: le premier token appartient est un prénom ('Georges Pompidou').
- ❖ **Dates**: le premier et le dernier token sont composés de chiffres ('5 juillet 2012').
- ❖ **Organisations**: le dernier token est "S.A." ou "SARL" ('Eiffage S.A.').
- ❖ **Lieux**: contient "sur" ou "en" suivi d'un nom de cours d'eau ('Montlouis-sur-Loire').

Nous présentons ici le système français Némésis.



**Figure II.2 :** Architecture générale de Némésis [19]

Némésis : un système orienté connaissances de REN pour le français.

### 7-2- Les approches orientées données :

Les systèmes de **REN** orientés données se fondent sur les mêmes observations (indices) que les systèmes orientés connaissances pour détecter les entités nommées, mais, à la différence de ces derniers, ils apprennent à extraire automatiquement les règles leur permettant d'utiliser les observations.

Cette approche requiert de grandes quantités de données d'entraînement annotées manuellement. En effet, disposer de nombreux exemples sous forme brute (sans annotation) et annotés permet d'apprendre à des systèmes à passer d'un format à l'autre en se fondant sur un ensemble de traits. Différents types de traits peuvent être trouvés dans la littérature, nombre d'entre eux sont communs avec les approches orientées connaissances.

Voilà quelques exemples de traits :

- ❖ morphologiques (les préfixes, suffixes, des n-grammes de suffixes et de préfixes...).
- ❖ syntaxiques (comme les parties du discours, arbres syntaxiques...).
- ❖ sémantiques (sorties de systèmes d'annotations sémantiques).
- ❖ dictionnaires de noms propres.

L'apprentissage des systèmes se fait alors automatiquement grâce à des procédures itératives permettant d'ajuster les configurations du système. Les algorithmes d'apprentissage les plus utilisés sont les machines à vecteurs supports (**SVM**), les modèles de Markov à états cachés (**MMC**), les arbres de décision et les modèles de champs conditionnels aléatoires. [20]

### 8-Travaux sur la catégorisation des EN :

La catégorisation des EN qui consiste à identifier les types des EN en les affectant à des catégories est une étape préliminaire consacrée à tout traitement automatique. En effet, elle permet de réduire la complexité des différentes tâches qui peuvent être effectuées sur les EN telles que la reconnaissance et la traduction. Cependant, la catégorisation n'est pas une tâche triviale dans la

mesure où il est non seulement nécessaire de déterminer les catégories à reconnaître mais aussi les différents constituants de chacune d'elles. [18]

Dans ce qui suit, nous décrivons les différentes catégorisations existantes selon des critères tels que la sémantique, la syntaxe.

### **8-1- Catégorisation sémantique (référentielle) :**

Catégorisations des EN ont été proposées en s'appuyant sur la référence. Dans ce qui suit :

- ✓ **Organisation:** regroupe les entreprises, les institutions gouvernementales et les autres organisations.
- ✓ **Person:** regroupe les noms de personnes ou de familles.
- ✓ **Location:** regroupe les noms de lieux politiquement ou géographiquement définis (villes, pays, régions, etc.).
- ✓ **Time:** regroupe les dates et données temporelles.
- ✓ **Nombre:** regroupe les données numériques comme les sommes d'argent ou les pourcentages.

### **8-2- Catégorisation syntaxique (graphique) :**

La distinction des EN, suivant des critères graphiques, est intéressante dans une optique de reconnaissance automatique. Suivant la graphie, l'identification et la classification des EN entraîneront des traitements différents. Nous distinguons, ainsi, les formes des entités nommées simples et complexes.

## **9-Détection d'entités nommées :**

De façon générale, deux grands types de technique peuvent être utilisés pour la conception d'un système de NERC. Le premier type regroupe les techniques basées sur des ensembles de règles grammaticales et syntaxiques qui ont été construites manuellement pour chaque type d'entité nommée considéré. Le deuxième type regroupe les techniques basées sur des modèles statistiques (Modèle de Markov Cache (**MMC**), Maximum Entropy Model (**MEM**) ou encore Conditional Random Field (**CRF**)) qui seront entraînées avec un ensemble de textes dans lesquels les entités nommées à détecter ont déjà été identifiées et classées. Après avoir été entraînées sur ces données, les modèles statistiques peuvent être utilisés pour identifier et classer les entités nommées présentes dans un segment de texte donné. Il existe également des systèmes hybrides qui utilisent à la fois un ensemble de règles grammaticales et syntaxiques et un ou des modèle(s) statistique(s) pour effectuer cette tâche. [21]

## **10-Conclusion :**

Nous avons présenté dans ce chapitre le statut théorique des entités nommées ainsi que les indices permettant leur identification. Plusieurs systèmes ont été développés pour différentes langues. La plupart d'entre eux utilisent soit des méthodes orientées connaissances, soit des méthodes orientées données.

Ayant présenté les différentes approches utilisées pour la REN, nous procédons dans le chapitre suivant à l'analyse de quelques systèmes selon la modalité du texte traité.

**Chapitre III:**  
**Classification des Documents**  
**Textuels**

## 1-introduction :

La classification des documents a deux méthodes différentes: la classification manuelle et la classification automatique. Dans la classification manuelle des documents, les utilisateurs interprètent la signification du texte, identifient les relations entre les concepts et catégorisent les documents.

Dans ce chapitre nous allons exposer la classification automatique de texte, plus en détail la catégorisation de textes. Nous présentons quelques définitions sur la classification et les différents jeux de mots utilisés : classification supervisée ou non supervisée, ensuite les différents objectifs de la classification.

## 2- Définition de la classification :

La classification des documents est définie comme une opération qui identifie des classes d'équivalence entre des segments de textes en tenant compte de leur contenu informationnel. [22]

## 3- Les méthodes de classification automatique :

L'objectif de la Catégorisation de texte (CT) est de classer de façon automatique les documents dans des catégories qui ont été définies soit préalablement par un expert, il s'agit alors de classification supervisée, soit de façon automatique, il s'agit alors de classification non supervisée.

### 3-1- Classification Supervisé :

Le cadre général de l'apprentissage supervisé consiste, à partir de l'observation d'un ensemble de couple de données de la forme  $[(x(i), y(i)), (i = 1 \rightarrow n)]$ , à induire la valeur de  $y$  pour de nouvelles valeurs de  $x$ . Dans un cadre probabiliste, chaque  $x(i)$  représente une observation d'une variable aléatoire  $X$ .

Suivant les valeurs de la variable aléatoire  $Y$ , deux cas de figures peuvent être distingués : Lorsqu'elle prend des valeurs discrètes.

Ainsi, la catégorisation de textes correspond à la procédure d'affectation d'une ou de plusieurs catégories ou classes prédéfinies à un texte. Elle correspond à la classification supervisée pour l'apprentissage automatique et à la discrimination en statistiques alors que la recherche d'informations utilise des termes plus proches de l'application concernée : filtrage ou routage.

Aujourd'hui, cette problématique utilise largement des méthodes issues de l'apprentissage automatique et beaucoup d'algorithmes d'apprentissage supervisé lui ont été appliqués (K-plus proches voisins, arbres de décision, machines à vecteurs support). [22]

### 3-1-1- Les étapes de classification :

Classer les documents revient en réalité à déterminer les paramètres de la fonction de classement. Voici l'idée globale de ce qu'on doit faire :

- Il faut disposer d'un corpus d'apprentissage, qui va servir d'entrée à un algorithme d'apprentissage.
- On sélectionne un autre corpus qui sert pour l'évaluation (corpus de test)
- Il faut fournir à l'ordinateur un type de fonctions de classement lui permettant d'associer une catégorie à un texte (machines à support de vecteurs (SVM), Arbres de décision).

- On infère, à partir des données, et par des méthodes mathématiques complexes, les paramètres de la fonction de classement utilisé, qui peuvent être :
  - Coefficients de l'hyperplan dans les **SVM**.
  - Distributions de probabilité dans les classificateurs probabilistes.
  - Règles dans les règles de décision.
  - Conditions et branchements dans les arbres de décision.
- On se fonde sur la connaissance préalable des bonnes catégories pour les documents du corpus d'apprentissage (apprentissage supervisé).

### 3-2- Classification non Supervisé :

La classification non supervisée consiste à trouver de manière automatique une organisation cohérente à un groupe de documents homogènes pour construire des regroupements cohérents (des classes ou clusters), elle correspond en statistiques au regroupement, qui est également le terme utilisé en recherche d'informations.

Le regroupement consiste donc, à diviser les objets (dans notre cas des textes) en groupes sans connaître a priori leurs classes d'appartenance. Les techniques pour réaliser de tels regroupements constituent un domaine d'étude très riche, qui a donné lieu à de multiples propositions dont le recensement n'est pas l'objet de ce document. . [22]

L'apprentissage non supervisé est utilisé dans plusieurs domaines tels que :

- Traitement d'images.
- Classification de documents.
- Médecine : Découverte de classes de patients présentant des caractéristiques physiologiques communes.

Dans la littérature il existe plusieurs types d'algorithmes d'apprentissage non supervisé tels que les algorithmes de partitionnements et les algorithmes de classification hiérarchique :

- **Le partitionnement:** consiste au regroupement des données suivant leur degré de similarité. L'algorithme le plus célèbre appartenant à cette classe est **K-means** : c'est un algorithme qui permet de partitionner un ensemble de données automatiquement en K clusters.
- **La classification hiérarchique:** il existe deux types de classification hiérarchique : Ascendante et descendante. La classification ascendante consiste à utiliser une matrice de similarité afin de partir d'une répartition fine vers un groupe unique. La classification descendante se présente comme l'inverse de la classification ascendante.

### 4- Différents modèles de classifier :

La catégorisation de textes comporte un choix de technique d'apprentissage (ou classificateur) disponibles. Parmi les méthodes d'apprentissage les plus souvent utilisées figurent : Machines à support de vecteurs (SVM), les réseaux de neurones, les arbres de décision, les méthodes dites de Boosting et les Critères d'évaluation des classificateurs :

#### 4-1- Machines à support de vecteurs (SVM) :

L'algorithme **SVM** est une méthode d'apprentissage supervisée relativement récente introduite pour résoudre un problème de reconnaissance de formes à deux classes.

La méthode SVM est un classificateur linéaire utilisant des mesures de distance.

- **Hyperplan** : est un séparateur d'objets des classes. Cet hyperplan est appelé marge se définit comme la plus petite distance entre les exemples de chaque classe et la surface séparatrice  $S$  :

$$\text{Marge}(S) = \sum_{c_j \in C} \min_{x_i \in C_j} (d(x, S))$$

- **Vecteurs Support** : ce sont les points qui déterminent l'hyperplan tels qu'ils soient les plus proches de ce dernier.

#### 4-2- Réseau neuronaux :

C'est une technique de type induction c'est-à-dire que, par le biais d'observations limitées, elle essaye de tirer des généralisations plausibles. Elle est basée sur l'expérience qui se constitue une mémoire lors de la phase d'apprentissage (qui peut être aussi non supervisée) appelée entraînement.

#### 4-3- Arbres de décision :

Un arbre de décision a pour but d'expliciter une catégorisation dans le cas où les textes classés sont décrits par un ensemble de termes qui représentent les propriétés caractéristiques des textes. Un arbre de décision est un arbre dont les nœuds sont :

- Soit des feuilles contenant des objets appartenant tous à une même catégorie. Les feuilles représentent donc les classes d'une même catégorie  $C$  (sous ensembles de  $C$ ).
- Soit des nœuds de décision qui partitionnent les données suivant plusieurs sous ensembles, chaque sous-ensemble correspondant à un résultat d'une fonction de test, cette fonction caractérisant le nœud de décision.

#### 4-4- Le Boosting :

Il s'agit d'une méthode de classification émettant des hypothèses qui sont au départ de moindre importance. Plus une hypothèse est vérifiée, plus son indice de confiance augmente. Ce qui prend de l'importance dans la classification.

#### 4-5- Critères d'évaluation des classificateurs :

Nous considérons ici un problème simple de classification pour lequel nous nous intéressons à une classe unique  $C$  et nous voulons évaluer un système qui nous indique si un document peut être associé ou non à cette classe  $C$ . Ce problème est un problème de classification à deux classes ( $C$  et non  $C$ ). Si on peut maîtriser ce problème simple, on pourra fusionner par la suite, les mesures de performance de plusieurs systèmes bi-classes afin d'obtenir une mesure de la performance d'un classifieur multi-classes.

##### 4-5-1- Matrice de contingence :

Pour évaluer un système de classification de ce type, qui fournit 4 informations essentielles :

**Vrai Positif (VP)** : Documents attribués à leurs vraies catégories.

Faux Positif (FP) : Documents attribués à des mauvaises catégories.

Faux Négatif (FN) : Le nombre de documents inconvenablement non attribués.

Vrai Négatif(VN) : Le nombre de documents non attribués à une catégorie convenablement.

#### 4-5-2- Le rappel :

Etant la proportion de documents correctement classés dans par le système par rapport à tous les documents de la classe  $C_i$ .

$$\mathbf{Rappel (Rp)} = \frac{VP}{VP + FN}$$

#### 4-5-3- La précision :

Est la proportion de documents correctement classés parmi ceux classés par le système dans  $C_i$ .

$$\mathbf{Précision (P)} = \frac{VP}{VP + FP}$$

#### 4-5-4- La F-mesure :

Est la combinaison de la précision et du rappel et leur pondération.

$$\mathbf{F-mesure} = \frac{2*(précision *rappel)}{(precision + rappel)}$$

### 5- Classification de textes :

La classification de textes est un domaine où les algorithmes sont appliqués sur des documents de texte. Cette tâche consiste à attribuer un document dans une ou plusieurs classes, en fonction de son contenu. En règle générale, ces classes sont triées sur le volet par les humains. Par exemple, considérons la tâche classifiant l'ensemble de documents comme bon ou mauvais. Dans ce cas, les catégories (ou étiquettes) « bon » et « mauvais » représentent les classes. . [23]

Certaines applications populaires où la classification de textes est appliquée sont les suivantes :

- Classer les nouvelles comme Politique, Sports, Monde, Affaires, Style de vie.
- Classer Les documents de recherche par type de conférence.
- Classer les critiques de films comme bons, mauvais et neutres.

Pour qu'un classifieur apprenne à classer les documents, il faut une sorte d'apprentissage machine. A cet effet, les objets d'entrée sont divisés en données d'apprentissage et des données de test (essai). L'apprenant es t responsable d'appliquer une fonction de classification (F) qui associe les documents (D) à la classe (C), comme suit :

$$F : D \rightarrow C$$

### 6- Les étapes de représentation :

Dans notre projet les documents classés et les documents non classé passent par les étapes [23] suivantes:



## 6-1- Représentation de textes :

Cette étape consiste à mettre en œuvre une série de prétraitements sur les documents classés et les documents non classés pour extraire l'ensemble des mots, les textes sont transformés en vecteur dont chaque composante représente un mot.

### 6-1-1- Tokenisation :

Dans cette étape, il s'agit d'enlever toute la ponctuation. Voici la liste des ponctuations qu'on a utilisé : {+ -\* / ; : ( ) ! , ? < > 0123456789}.

### 6-1-2- Elimination des majuscules :

Dans cette étape il s'agit de transformer les majuscules en minuscules ; en effet le mot "GIRL" et le mot "girl" vont être considérés différents alors qu'ils ont le même sens donc on transforme les majuscules en minuscule.

### 6-1-3- Elimination des mots vides :

Les mots vides sont les mots qui se répètent fréquemment dans tous les documents et qui n'ont aucun pouvoir discriminant lors du processus de la catégorisation de texte. La liste des mots vides contient les pronoms personnels, les prépositions, les articles....etc.

### 6-1-4- Lemmatisation / racinisation :

Est une transformation des mots vers leur forme de racine ou de lemme. Les mots dans le texte existent sous une forme dérivée, représentée par cette racine ou lemme. La racine d'un mot correspond à la partie du mot restante une fois que l'on a supprimé son préfixe ou son suffixe, à savoir son radical. Contrairement, le lemme correspond à un mot réel de la langue. La racinisation ne correspond généralement pas à un mot réel. Par exemple, «ran», «running », «runs» sont tous des dérivés du mot « run ».

## 7- Conclusion :

La classification des documents a fait beaucoup de progrès ces dernières années. Nous avons présenté les principales techniques de classification automatique et manuelle, utilisées pour classer des unités textuelles.

Dans ce chapitre nous avons présenté quelques techniques de la classification automatique des textes. Nous basons dans notre travail sur les méthodes Machines à support de vecteurs (SVM), Réseau neuronal, Arbres de décision et Boosting. Nous avons également introduit les différents moyens d'évaluation d'un classificateur. Qui ont amélioré très significativement les taux de bonne classification.

**Chapitre IV:**  
**Conception et Réalisation**

**Partie I :****1-Introduction :**

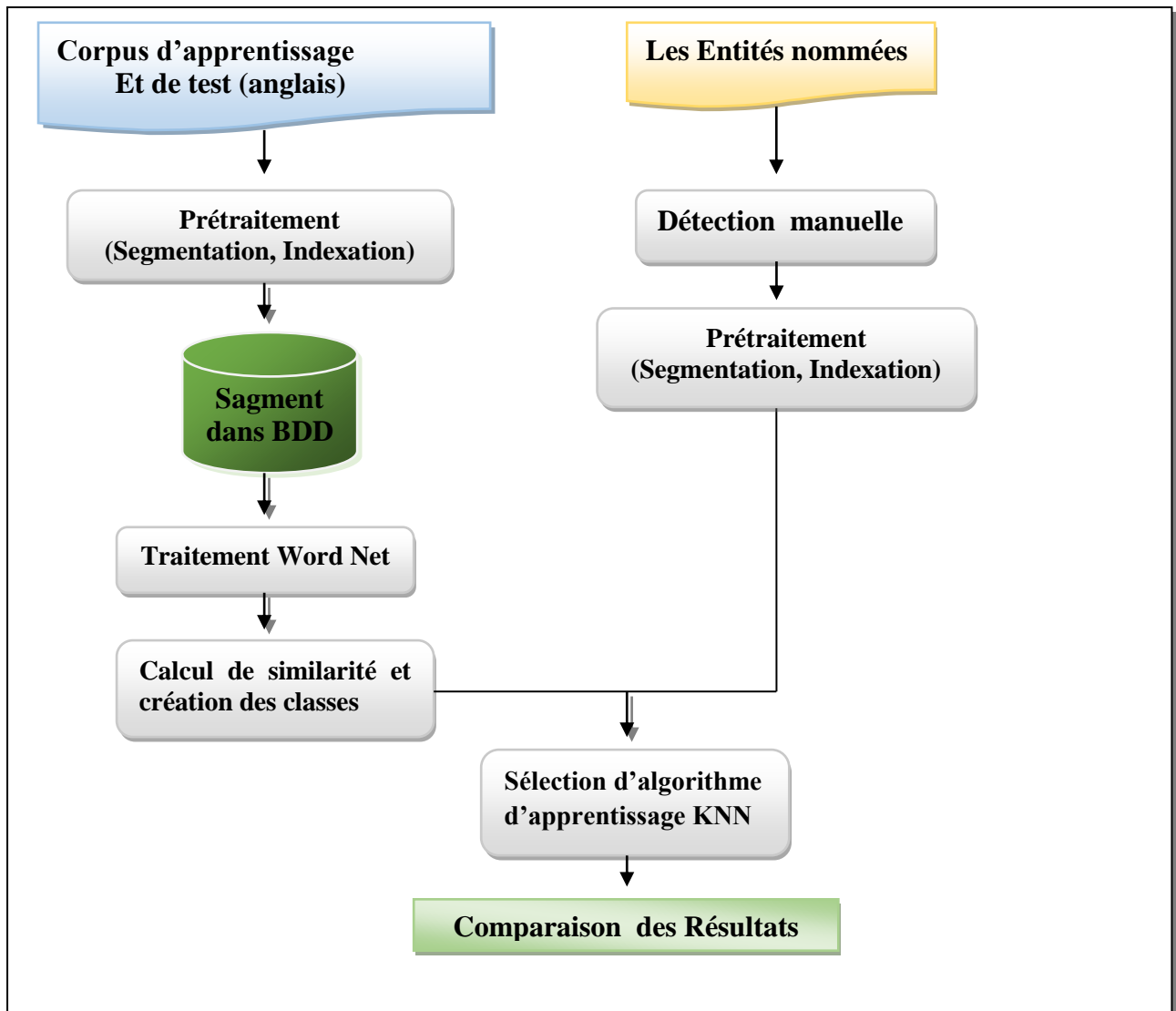
Notre application consiste à concevoir un système de recherche d'information et de classification de documents textuels par détection entité nommées, cette application permet de présenter un système de prédiction de thématique d'un document donnée en entrée.

Pour ce faire, nous avons utilisé un corpus (anglais) qui est un ensemble de documents textuels, et les entités nommées, le wordnet pour traité les documents classés.

Ensuite nous choisissons une méthode de classification dans le but de prédire la catégorie du document à classer. Plusieurs méthodes existent, dans notre travail, on a utilisé la méthode de KNN. Dans ce chapitre, nous verrons tout d'abord l'architecture générale de notre projet dans laquelle nous donnerons les différentes étapes.

**2- Architecture générale du système :**

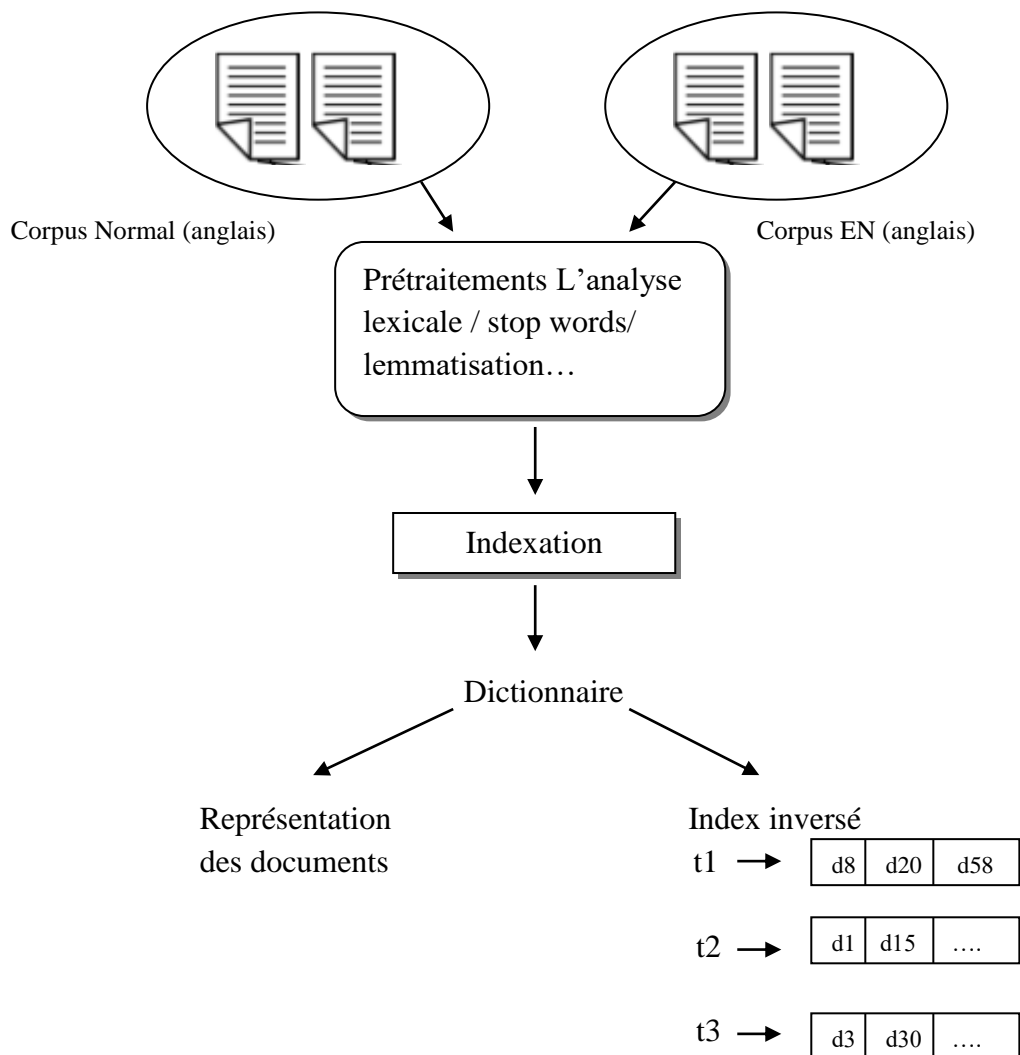
Le système est réalisé en deux phases essentielles : une phase d'apprentissage et une phase de classification. La figure (Figure 3.1.) montre l'architecture générale du système :



**Figure IV.1 :** Architecture de notre travail.

## 2-1- Prétraitements (Segmentation, indexation) :

La segmentation du corpus est une phase essentielle dans l'analyse. Elle permet de découper le corpus et entités nommées (anglais) en segments disjoints (paragraphe, phrases ou mots) dans le but de pouvoir les indexer par la suite:



**Figure IV.2 :** *Prétraitement et représentation des documents.*

- ✓ **Représentation corpus :** Cette étape consiste à mettre en œuvre une série de prétraitements sur les corpus et les ENs (anglais), les textes sont transformés en vecteur dont chaque composante représente un mot.
- ✓ **L'analyse lexicale :** Est un processus qui convertit le texte d'un document en un ensemble de termes ou un terme est considéré comme une unité lexicale. Cette analyse permet de reconnaître les espaces de séparation des chiffres, des mots, des ponctuations, ... etc.
- ✓ **mots vides:** traitement permet de garder seulement les termes significatifs qui représentent le contenu de document. Afin d'éliminer les mots vides de sens, nous avons utilisé une liste, appelée stop Word List (ou parfois anti- dictionnaire) qui contient tous les pronoms personnels, les adverbes, les articles, les conjonctions de coordination, les verbes auxiliaires de la langue anglaise.

a, a's, able, about, above, according, accordingly, across, actually, after, afterwards, again, against, ain't, all, allow, allows, almost, alone, along, already, also, although, always, am, among, amongst, an, and, another, any, anybody, anyhow, anyone, anything, anyway, anyways, anywhere, apart, appear, appreciate, appropriate, are, aren't, around, as, aside, ask, asking, associated, at, available, away, awfully, b, be, became, because, become, becomes, becoming, been, before, beforehand, behind, being, believe, below, beside, besides, best, better, between, beyond, both, brief, but, by, c, c'mon, c's, came, can, can't, cannot, cant, cause, causes, certain, certainly, changes, clearly, co, com, come, comes, concerning, consequently, consider, considering, contain, containing, contains, corresponding, could, couldn't, course, currently, d, definitely, described, despite, did, didn't, different, do, does, doesn't, doing, don't, done, down, downwards, during, e, each, edu, eg, eight, either, else, elsewhere, enough, entirely, especially, et, etc, even, ever, every, everybody, everyone, everything, everywhere, ex, exactly, example, except, f, far, few, fifth, first, five, followed, following, follows, for, former, formerly, forth, four, from, further, furthermore, g, get, gets, getting, given, gives, go, goes, going, gone, got, gotten, greetings, h, had, hadn't, happens, hardly, has, hasn't, have, haven't, having, he, he's, hello, help, hence, her, here, here's, hereafter, hereby, herein, hereupon, hers, herself, hi, him, himself, his, hither, hopefully, how, howbeit, however, I, i'd, i'll, i'm, i've, ie, if, ignored, immediate, in, inasmuch, inc, indeed, indicate, indicated, indicates, inner, insofar, instead, into, inward, is, isn't, it, it'd, it'll, it's, its, itself, j, just, k, keep, keeps, kept, know, knows, known, l, last, lately, later, latter, latterly, least, less, lest, let, let's, like, liked, likely, little, look, looking, looks, ltd, m, mainly, many, may, maybe, me, mean, meanwhile, merely, might, more, moreover, most, mostly, much, must, my, myself, n, name, namely, nd, near, nearly, necessary, need, needs, neither, never,, nevertheless, new, next, nine, no, nobody, non, none, no one, nor, normally, not, nothing, novel, now, nowhere, o, obviously, of, off, often, oh, ok, okay, old, on, once, one, ones, only, onto, or, other, others, otherwise, ought

our, ours, ourselves, out, outside, over, overall, own, p, particular, particularly, per, perhaps, plac\_d, please, plus, possible, presumably, probably, provides, q, que, qv, r, rather, rd, re, really, reasonably, regarding, regardless, regards, relatively, respectively, right, s, said, same, saw, say, saying, says, second, secondly, see, seeing, seem, seemed, seeming, seems, seen, self, selves, sensible, sent, serious, seriously, seven, several, shall, she, should, shouldn't, since, six, so, some, somebody, somehow, someone, something, sometime, sometimes, somewhat, somewhere, soon, sorry, specified, specify, specifying, still, sub, such, sup, sure, t, t's, take, taken, tell, tends, th, than, thank, thanks, thanx , that, that's, that's, the, their, theirs, them, themselves, then, there, there's, thereafter, thereby, therefore, therein, theres, thereupon, these, they, they'd, they'll, they're, they've, think, third, this, thorough, thoroughly, those, though, three, through, throughout, thru, thus, to, together, too, took, toward, towards, tried, tries, truly, try, trying, twice, two, u, un, under, unfortunately, unless, unlikely, until, unto, up, upon, us, use, used, useful, uses, using, usually, uucp, v, value, various, very, via, viz, vs, w, want, wants, was, wasn't, way, we, we'd, we'll, we're, we've, welcome, well, went, were, weren't, what, what's, whatever, when, whence, whenever, where, where's, whereafter, whereas, whereby, wherein, whereupon; wherever, whether, which, while, whither, who, who's, whoever, whole, whom, whose, why, will, willing, wish, with, within, without, won't, wonder, would, would, wouldn't, x, y, ye, yet, you, you'd, you'll, you're, you've, your, yours, yourself, yourselves, z, zero.

Figure IV.3: Liste des mots vides.

- ✓ **Normalisation** : Objectif obtenir une forme canonique pour les différents mots d'une même famille :
  - **la forme textuelle** : Accents, casse, ponctuations, symboles spéciaux, dates.
- ✓ **Stemming** : consiste à réduire un mot à sa racine (stem), qui peut ne pas exister. L'algorithme de Porter est un des plus connus pour la langue anglaise. Il applique une succession de règles (mécaniques) pour réduire la longueur des mots c.-à-d. supprimé la fin des mots (Le stemming est un traitement final, qui n'autorise plus de post-traitements sur les mots).
- **Exemple 1** : On utilise d'abord la règle qui s'applique sur le plus long suffixe.
  - **avant stemming**: 549 caractères.

ohio mattress co said first quarter ending february profits may  
 mln dlrs cts share earned first quarter fiscal company said  
 decline due expenses related acquisitions middle current  
 quarter seven licensees sealy inc well pct outstanding capital  
 stock sealy acquisitions said first quarter sales will substantially  
 higher last years mln dlrs noting typically reports first quarter  
 results late march said report likely issued early april year said  
 delay due administrative considerations including conducting  
 appraisals connection acquisitions reuter

- **après stemming**: 477 caractères.

ohio mattress co said first quarter end february profit may mln dlrs  
 cts share earn first quarter fiscal compani said declin due expens  
 relat acquisit middl current quarter seven license seali inc well pct  
 outstand capit stock seali acquisit said first quarter sale will  
 substanti higher last year mln dlrs note typic report first quarter  
 result late march said report like issu earli april year said delay due  
 administr consider includ conduct apprais connect acquisit reuter

- ✓ **Représentations** : nombre de documents où le terme apparaît au moins une fois rapporté au nombre total de documents (fréquence des termes).
- ✓ **L'index inversé** : À partir de l'ensemble de mots nettoyés et les segments obtenus nous construisons l'index inversé qui sert à lier ces mots aux segments où ils se trouvent. chaque terme de l'index est décrit par le numéro de référence de tous les documents (frequency) qui contiennent ce terme et la position dans ce document du terme (doc frequency).

✓

**Objectif:**

Trier les paires (term\_id, doc\_id) suivant les clés term\_id puis doc\_id pour produire un index inversé.

- Le corpus est partitionné.
- Chaque élément de la partition est analysé doc par doc pour produire un index inversé avec cet ensemble de termes x documents.

**2-2- Segment dans BDD (stockage):**

La configuration d'une base de données pour stocker les corpus indexé (apprentissage, test).

Création les tableaux des liste termes (listetermea, listetermet) et des liste documents (doctermea, doctermet) à chaque corpus indexés (apprentissage, test).

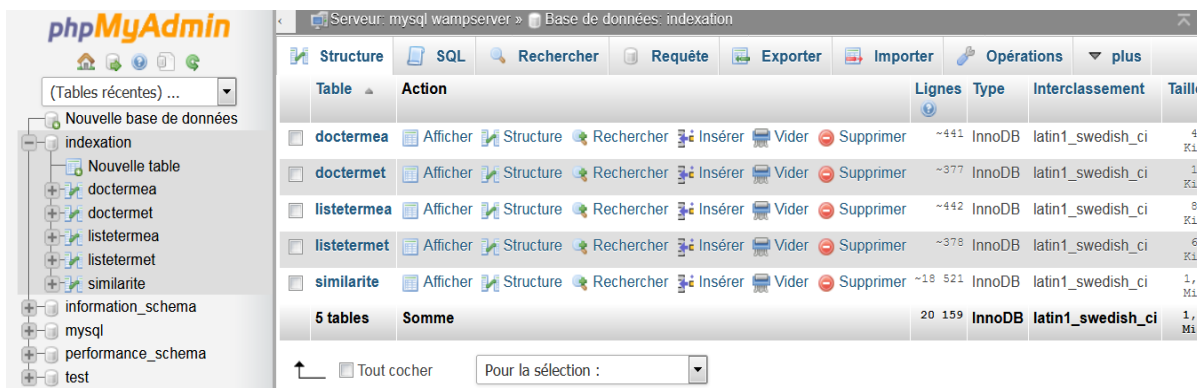
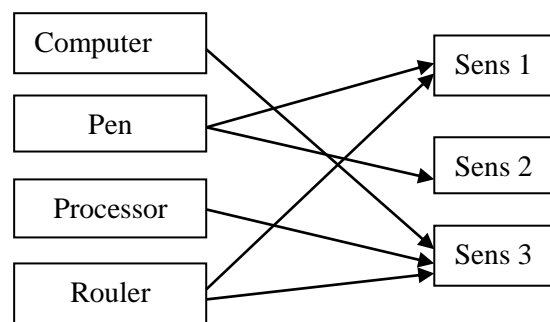


Table	Action	Lignes	Type	Interclassement	Taille
doctermea	Afficher Structure Rechercher Insérer Vider Supprimer	~441	InnoDB	latin1_swedish_ci	45 Kio
doctermet	Afficher Structure Rechercher Insérer Vider Supprimer	~377	InnoDB	latin1_swedish_ci	14 Kio
listetermea	Afficher Structure Rechercher Insérer Vider Supprimer	~442	InnoDB	latin1_swedish_ci	80 Kio
listetermet	Afficher Structure Rechercher Insérer Vider Supprimer	~378	InnoDB	latin1_swedish_ci	6 Kio
similarite	Afficher Structure Rechercher Insérer Vider Supprimer	~18 521	InnoDB	latin1_swedish_ci	1,5 Mio
<b>5 tables</b>	<b>Somme</b>	<b>20 159</b>	<b>InnoDB</b>	<b>latin1_swedish_ci</b>	<b>1,7 Mio</b>

**Figure IV.4:** Les tableaux des corpus indexés.

**2-3- Transformation des mots en synsets :**

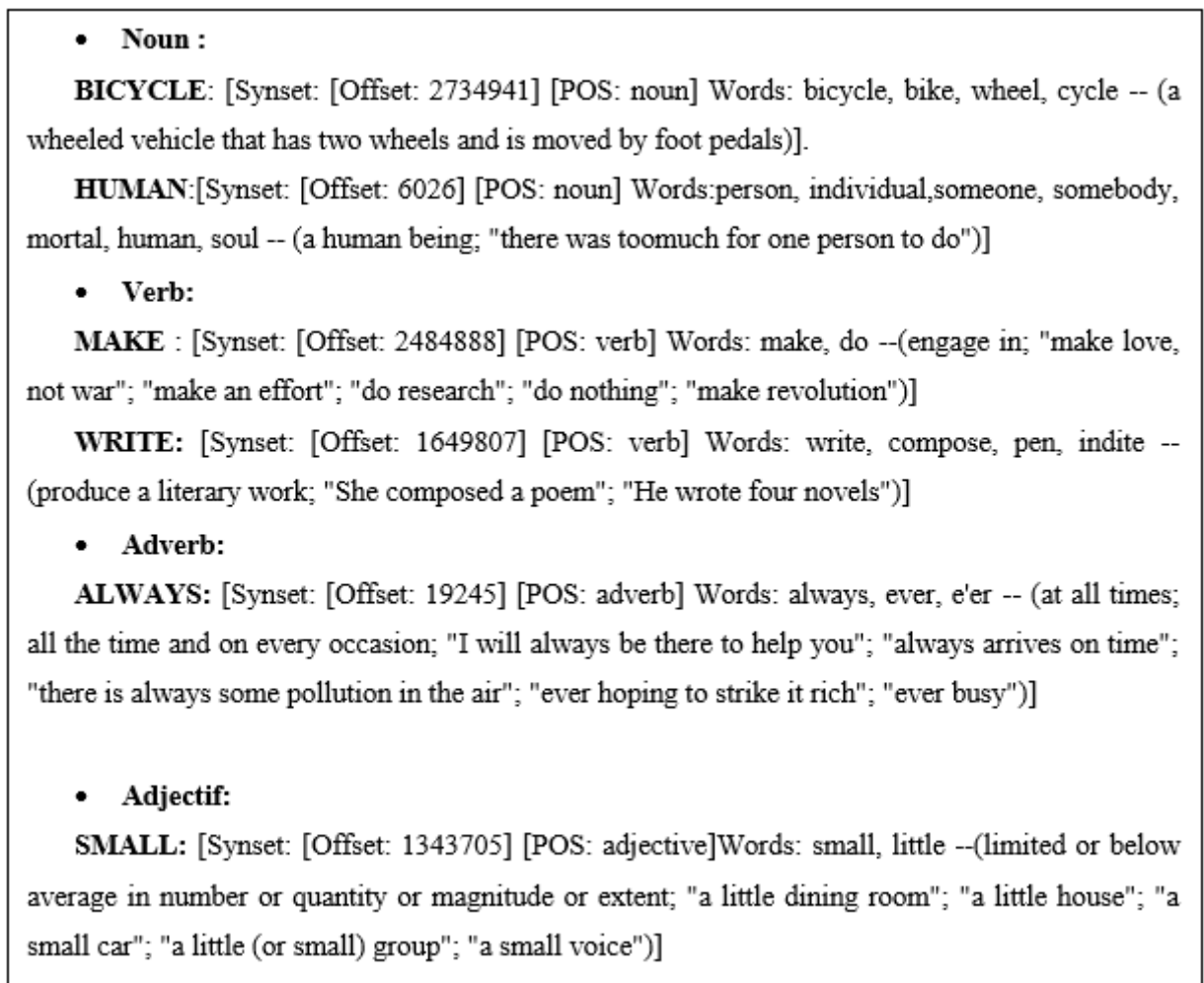
Après la représentation de chaque document par son vecteur, nous passons à l'étape de la transformation des mots en synset et cela grâce à une base lexicographique WordNet dans laquelle les mots sont regroupés au sein de groupes de synonymes appelés synsets qui indique un sens différent du mot. La base lexicographique WordNet renvoie une liste ordonnée de synsets pour chaque mot qui peut ajouter le bruit à la représentation et peut induire une perte d'information. La figure IV.5 ci-dessous montre qu'un mot peut avoir plusieurs sens, pour éviter ce problème de la désambiguïsation de sens.



**Figure IV.5:** Combinatoire des sens.

➤ **Exemple de groupes de synset :**

La figure IV.6 présentée la base lexicographique WordNet renvoie une liste ordonnée de synsets pour chaque mot qui peut être un nom, verbe, adverbe ou adjectif.



**Figure IV.6 :** *Exemple d'un groupe de synset.*

## 2-4- Représentation conceptuelle :

La représentation conceptuelle se base sur le formalisme vectoriel pour représenter les documents. Les éléments de cette représentation ne sont plus des mots, mais plutôt des concepts, et cela grâce à l'étape précédente qui est la transformation des mots en synsets.

➤ **Exemple 3 :**

La représentation matricielle d'un corpus où les lignes représentent les documents du corpus, les colonnes représentent les termes, Comptabiliser la présence de chaque terme dans le document, sans se préoccuper du nombre d'occurrences (de la répétition).

D1. {image databases can get huge}.

D 2. {most image database store image permanently}

D3. {image databases store image baby}

D4. {image databases store image image databases store image image databases store image}



termes Doc	databases	huge	image	permanently	store
D1	1	1	1	0	0
D2	1	0	1	1	1
D3	1	0	1	0	1
D4	1	0	1	0	1

Table IV.1: Représentation matricielle d'un corpus.

## 2-5- Calcul de similarité et création des classes :

Les documents sont représentés par des ensembles de vecteurs de termes. Chaque unité de contexte génère un vecteur. Les poids des termes sont calculés en fonction de leur distribution dans les balises. Il est calculé pour un document et un contexte (à savoir la balise) donnés.

### 2-5-1 Calcul des poids :

Dans le but d'attribuer à chaque terme un poids qui mesure son importance au sein du corpus d'apprentissage et de test (Listes des termes, (Documents x termes)), on a opté pour l'utilisation de

✓ **formule de pondération suivante :**

- **La fréquence du terme (noté : tf) :** représente le nombre de fois qu'un terme est apparu dans le texte sur le nombre de tous les termes du texte. (la fréquence de terme dans le document ou la requête)

Binaire  $\rightarrow$   $tf = 1$  ou  $0 \rightarrow \{1 \text{ si terme E Doc ou } 0 \text{ si terme Il Doc}\}$

Fréquentielle normalisé simple  $\rightarrow$   $tf(t,d) = \frac{F_{t;d}}{\sum_{t'} f_{t',d}}$

- **La fréquence documentaire (noté : idf) :** représente la mesure l'importance du terme dans l'ensemble de la collection. Cette métrique valorise les termes les moins fréquents dans un corpus.

$idf(t,d) = \log(N/n)$

N : le nombre total de documents de la collection.

n : le nombre de documents contenant le terme.

$w_i = tf \times idf = tf \times \log(N/n)$

- **Exemple 3 (suite):**

1- Les normalisations des fréquences permettent d'amortir les écarts et/ou de tenir compte de la longueur des documents (tf).

termes Doc	Databases	huge	image	permanently	Store
D1	1	1	1	0	0
D2	1	0	2	1	1
D3	1	0	2	0	1
D4	3	0	6	0	3

**Table IV.2 :** Les fréquences de tous les termes dans le document.

2- Normalisation simple : On pondère la fréquence par le nombre de termes présents dans le document.

➤ **Les valeurs tf :**

databases { (D1) :  $1/3 = 0.33$ , (D2) :  $1/5 = 0.20$ , (D3) :  $1/4 = 0.25$ , (D4) :  $3/12 = 0.25$  }

huge { (D1) :  $1/3 = 0.33$ , (D2) :  $0/5 = 0.00$ , (D3) :  $0/4 = 0.00$ , (D4) :  $3/12 = 0.00$  }

image { (D1) :  $1/3 = 0.33$ , (D2) :  $2/5 = 0.40$ , (D3) :  $2/4 = 0.50$ , (D4) :  $6/12 = 0.50$  }

permanently { (D1) :  $0/3 = 0.00$ , (D2) :  $1/5 = 0.20$ , (D3) :  $0/4 = 0.00$ , (D4) :  $0/12 = 0.00$  }

store { (D1) :  $0/3 = 0.00$ , (D2) :  $1/5 = 0.2$ , (D3) :  $1/4 = 0.25$ , (D4) :  $3/12 = 0.25$  }

termes Doc	Databases	Huge	image	permanently	Store
D1	0.33	0.33	0.33	0	0
D2	0.20	0.00	0.40	0.20	0.20
D3	0.25	0.00	0.50	0.00	0.25
D4	0.25	0.00	0.50	0.00	0.25

**Table IV.3:** Fréquences des termes (tf).

3- Un terme présent dans presque tout le corpus (D) influe peu quand il apparaît dans un document (Inverse document frequency (IDF)).

➤ **Les valeurs idf :**

databases:  $\log(4/4) = 0.000$

permanently:  $\log(4/1) = 0.602$

huge:  $\log(4/1) = 0.602$

store:  $\log(4/3) = 0.125$

image:  $\log(4/4) = 0.0002$

termes / Doc	Databases	Huge	image	permanently	Store
D1	1	1	1	0	0
D2	1	0	2	1	1
D3	1	0	2	0	1
D4	3	0	6	0	3
n_t	4	1	4	1	3
Idf(t,d)	0.000	0.602	0.000	0.602	0.125

**Table IV.4 :** *Fréquences des termes (idf).*

4- Pondération TF-IDF : Relativiser l'importance d'un terme dans un document (TF) par son importance dans le corpus (IDF).

➤ **Les valeurs TF-IDF :**

databases {(D1) :  $1 * 0.0 = 0.0$ , (D2) :  $1 * 0.0 = 0.0$ , (D3) :  $1 * 0 = 0.0$ , (D4) :  $3 * 0.0 = 0.0$ }

huge {{{(D1) :  $1 * 0.60 = 0.60$ , (D2) :  $0 * 0.60 = 0.00$ , (D3) :  $0 * .60 = 0.00$ , (D4) :  $0 * 0.60 = 0.00$ }}

image {{{(D1) :  $1 * 0.00 = 0.00$ , (D2) :  $2 * 0.00 = 0.00$ , (D3) :  $2 * 0.00 = 0.00$ , (D4) :  $6 * 0.00 = 0.00$ }}

permanently {(D1) :  $0 * 0.60 = 0.00$ , (D2) :  $1 * 0.60 = 0.60$ , (D3) :  $0 * 0.60 = 0.00$ , (D4) :  $0 * 0.60 = 0.00$ }

store {(D1) :  $0 * 0.12 = 0.00$ , (D2) :  $1 * 0.12 = 0.12$ , (D3) :  $1 * 0.12 = 0.12$ , (D4) :  $3 * 0.12 = 0.37$ }

termes / Doc	Databases	Huge	image	permanently	Store
D1	0.00	0.60	0.00	0.00	0.00
D2	0.00	0.00	0.00	0.60	0.12
D3	0.00	0.00	0.00	0.00	0.12
D4	0.00	0.00	0.00	0.00	0.37

**Table IV.5 :** *Le TF -IDF d'un terme dans un document.*

### 2-5-2 Similarité entre termes :

Dans le but de créer des classes contenant des documents similaires on a calculé la similarité entre vecteurs de documents.

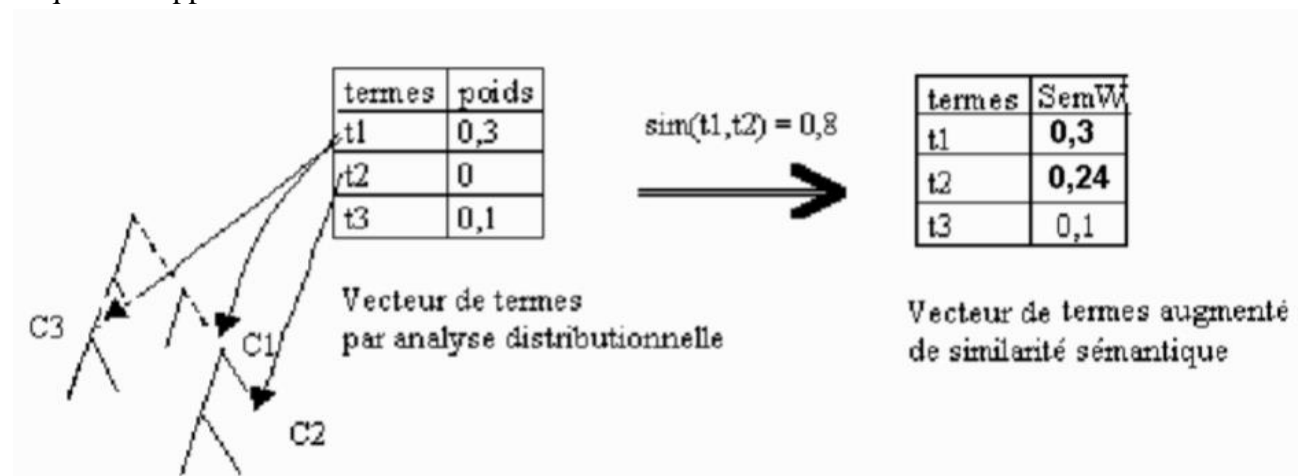
Après avoir testé plusieurs mesures de similarité sur un échantillon de « 5 » documents, le choix est tombé sur la mesure de cosinus qui a donné de meilleurs de calcul de similarité. Les

classes trouvées correspondent à un regroupement de documents similaire guidé en quelque sorte par un seuil.

La similarité entre deux documents d'une même classe doit être inférieure ou égale au seuil, et la similarité entre deux documents de classes différentes est strictement supérieure au seuil. Il est calculé pour un document. Ce poids noté  $SemW(t,b,d)$  est calculé de la manière suivante :

$$SemW(t,b,d) = TF-ITDF(t,b,d) + \sum_{i=1}^n Sim(t, ti) * TF - ITDF(ti, b, d) / n .$$

avec  $Sim(t,ti) > seuil$  ; n le nombre de termes dans la balise b et seuil une valeur qui fixe la similarité à un certain voisinage, nous la fixons dans un premier temps à la similarité entre le concept de t et le concept contexte . TF-ITDF (Term Frequency– Inverse Tag and Document Frequency) est le poids initial attribué aux termes en fonction du document et de la balise dans lesquels ils apparaissent.



**Figure IV.7 :** Calcul de la similarité sémantique.

Le calcul de la similarité entre les termes co-occurents dans la même balise nous permet de gérer aussi en partie le problème d'ambiguïté sémantique. En effet, dans la **Figure** le terme t1 est rattaché à deux concepts différents.  $Sim(t1, t2)$  est égal à la somme de  $sim(C1, C2)$  et  $sim(C3, C2)$ , C3 étant loin sémantiquement de C2, il ne sera pas pris en considération et C1 se trouve enrichi seulement par le poids de C2. Il est à noter que le poids de t3 reste inchangé du fait qu'il n'est rattaché à aucun concept. Ceci est très important car nous permet de faire une recherche par concepts ainsi que simplement par mots clés.

## 2-6- Sélection d'algorithme d'apprentissage KNN :

Nous allons passer à la classification des documents avec la méthode des K Nearest Neighbors (K-NN) qui compare les documents textuels et entités nommées. Nous avons choisi cet algorithme pour sa simplicité et sa fréquente utilisation dans le domaine de la catégorisation des documents textuels.

K-NN est un algorithme basé sur la similarité qui s'est révélé être très efficace pour divers domaines problématiques, notamment la catégorisation du texte. Dans un document de test, l'algorithme k-NN recherche les k voisins les plus proches parmi les documents de formation et utilise les catégories de k voisins pour pondérer les candidats de la catégorie. Le score de similarité de chaque document voisin par rapport au document est utilisé comme poids des catégories du document voisin. [24]

**Algorithme K-NN :**

L'algorithme de KNN comparé avec ceux déjà classés en cherchant ses K plus proches voisins. Une fois ces derniers déterminés, le nouveau document est classé dans la catégorie qui inclut le maximum de voisins parmi les K trouvés.

Deux paramètres sont utilisés : le nombre K et la fonction de similarité pour comparer le nouveau document à ceux déjà classés.

Le fonctionnement de l'algorithme KNN, est le suivant :

**Paramètre :** le nombre K de voisins

**Contexte :** un échantillon de L textes classés en  $C = c_1, c_2, \dots, c_n$  classes.

**Début**

Pour chaque texte T faire

Transformer le texte T en vecteur  $T = (x_1, x_2, \dots, x_m)$ ,

Déterminer les K plus proches textes du texte T selon une métrique de distance,

Combiner les classes de ses K exemples en une classe C.

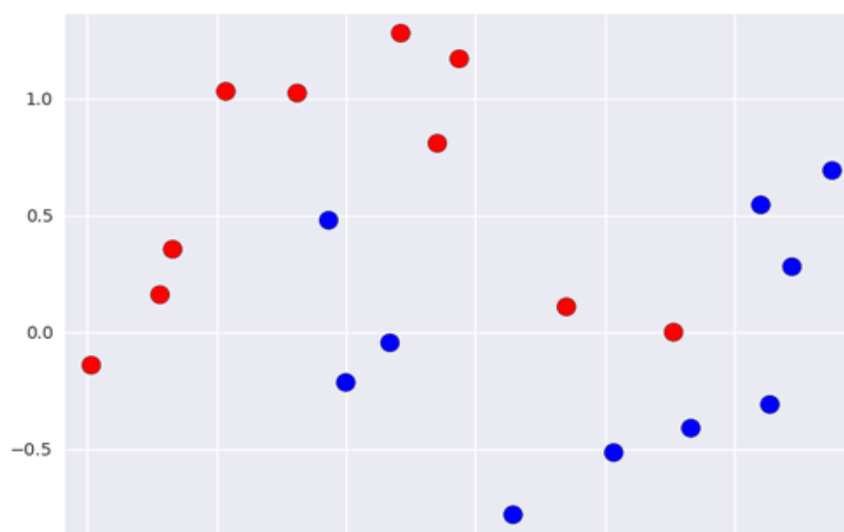
Fin pour

**Fin**

**Sortie :** le texte T associé à la classe C.

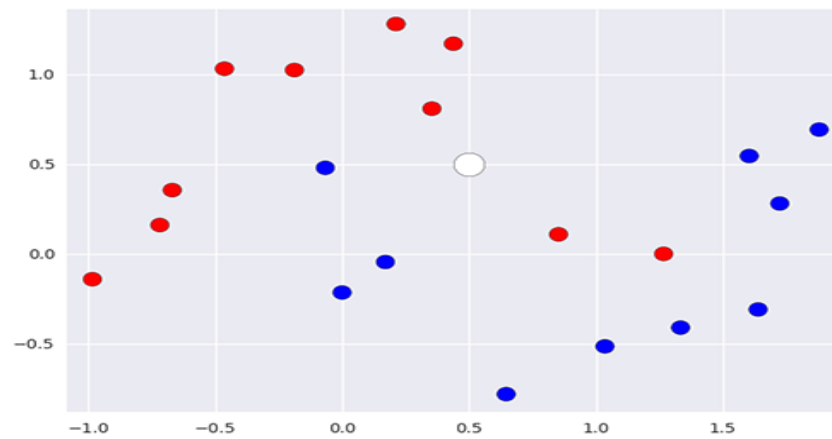
- **Exemple :**

1- Ci-dessous j'ai représenté un jeu de données d'entraînement, avec deux classes, rouge et bleu :



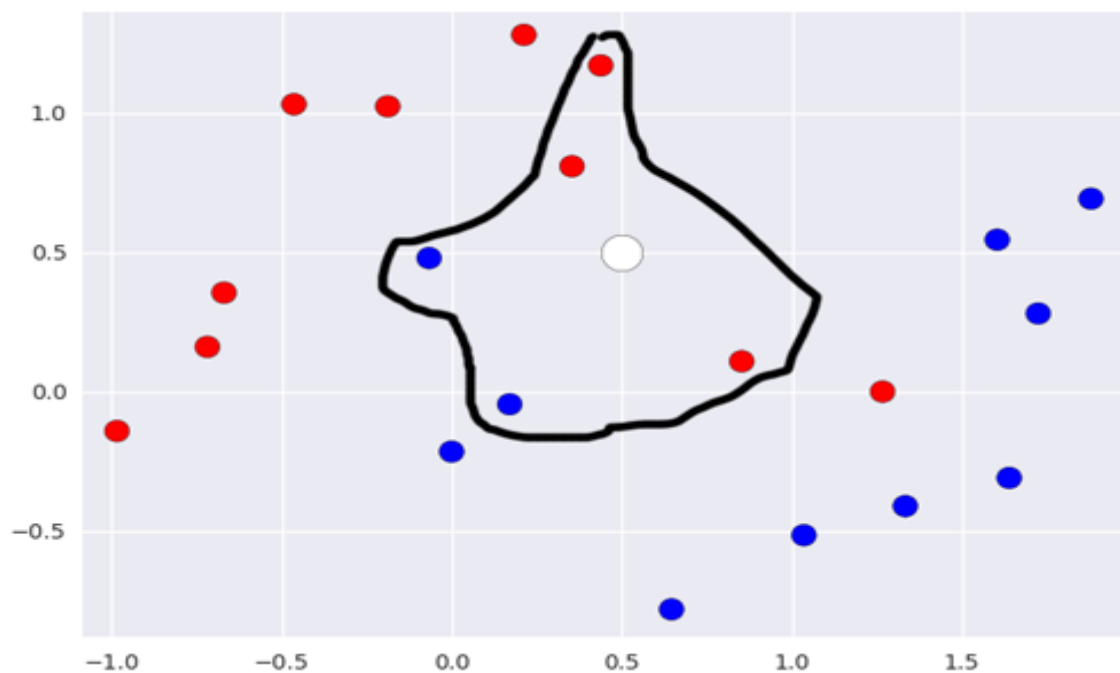
**Figure IV.8 :** Notre nuage de points de test.

2- Une nouvelle entrée dont on veut prédire la classe :



**Figure IV.9 :** *Le point blanc est une nouvelle entrée.*

3- les k voisins les plus proches de ce point et regarder quelle classe constitue la majorité de ces points, afin d'en en déduire la classe du nouveau point. Par exemple ici, si on utilise le 5-NN, on peut prédire que la nouvelle donnée appartient à la classe rouge puisqu'elle a 3 rouges et 2 bleus dans son entourage.



**Figure IV.10 :** *Les 5 points les plus proches du point que l'on cherche à classer.*

### 3- Etude sur les entités nommées à partir des corpus :

Une étude complémentaire a été menée sur les entités nommées. Le mode opératoire pour cette annotation automatique s'est déroulé comme suit :

- Segmentation : délimitation des entités.
- Typage des entités nommées.

Nouvelle mesure pondérée par le poids des entités. Le poids d'une entité est uniquement fonction de son type (date, lieu, personne, organisation, ...). Cela permet de rendre compte du caractère discriminant d'un type spécifique d'entité.

J'ai utilisé la méthode manuelle pour détecter les entités nommées à travers le logiciel Open Calais online (L'annotation de ce corpus en entités nommées a été effectuée manuellement).

Comment fonctionne Open Calais?

Open Calais analyse automatiquement votre texte saisi et effectue les processus suivants:

Reconnaissance des entités nommées et des relations - Open Calais identifie et balise les mentions (chaînes de texte) telles que des sociétés, des personnes, des transactions, des lieux, des industries, des noms, des organisations, des produits, etc., en fonction d'une liste de types de métadonnées prédéfinis. Le Figure représenter les type entités.

**Figure IV.11:** *Open Calais détecter les entités nommées à partir des corpus.*

City: CRANFORD.

Company: United States Lines Inc, Crowley Mariotime Corp.

Contry: U.S

Oraganization: Bankruptcy, Court.

## Partie II :

### 4- Environnement et outils de développement:

Dans cette section, nous allons présenter Les outils et les langages utilisés pour la manipulation des données ainsi que l'implémentation sont décrits comme suit :

## 4-1- Langage JAVA :

Notre choix pour le langage de programmation s'est porté sur le langage JAVA, et cela parce qu'il est un langage orienté objet simple ce qui réduit les risques d'incohérence et il possède une riche bibliothèque de classes comprenant des fonctions diverses telles que les fonctions standards, le système de gestion de fichiers ainsi que beaucoup de fonctionnalités qui peuvent être utilisées pour développer des applications diverses. Il existe une multitude de bibliothèques développées et fournies pour être utilisées en JAVA.

Les API (Application Programming Interface) des autres langages autres que JAVA ne sont pas finalisées et doivent encore être mises à jour.

## 4-2- Environnement de développement :

L'environnement de développement utilisé, est NetBeans IDE 8.2, il possède de nombreux avantages qui sont à l'origine de son énorme succès dont les principaux sont :

- ❖ Un environnement de développement intégré (EDI).
- ❖ Permet de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).
- ❖ La construction incrémentale des projets JAVA grâce à son propre compilateur, qui permet en plus de compiler le code même avec des erreurs, de générer des messages d'erreurs personnalisés, de sélectionner la cible, ...

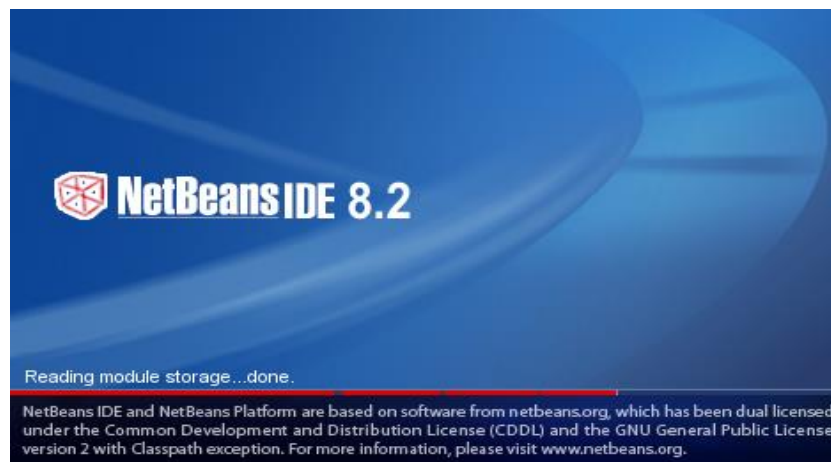


Figure IV.12: *NetBeans IDE 8.2.*

## 4-3- WampServer:

WampServer est une plateforme de développement web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. Wamp Server n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (apache et MySQL), un interpréteur de script (PHP), ainsi que PhpMyAdmin pour l'administration web des bases MySQL. Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs à travers un tray-icon (icône près de l'horloge de Windows).



## ➤ Exemple de bases de données :

The screenshot shows the phpMyAdmin interface. On the left, a tree view shows the database 'indexation' with tables like 'doctermea', 'doctermet', 'listetermea', etc. The main window displays the 'listetermea' table with the following data:

idTm	Document	Termes	Frequence
1	C:/2019/Mosta/2019 Hassane/CorpusT/training/acq/00...	march	1
2	C:/2019/Mosta/2019 Hassane/CorpusT/test/acq/000961...	dir	2
3	C:/2019/Mosta/2019 Hassane/CorpusT/test/acq/000961...	year	3
4	C:/2019/Mosta/2019 Hassane/CorpusT/test/acq/000961...	mln	1
5	C:/2019/Mosta/2019 Hassane/CorpusT/test/acq/000961...	april	1
6	C:/2019/Mosta/2019 Hassane/CorpusT/test/acq/000961...	u.	4
7	C:/2019/Mosta/2019 Hassane/CorpusT/test/acq/000961...	test	1
8	C:/2019/Mosta/2019 Hassane/CorpusT/test/acq/000961...	ha	3
9	C:/2019/Mosta/2019 Hassane/CorpusT/training/alum/0...	tonn	3
10	C:/2019/Mosta/2019 Hassane/CorpusT/training/barley...	export	7
11	C:/2019/Mosta/2019 Hassane/CorpusT/test/acq/000961...	pct	3
12	C:/2019/Mosta/2019 Hassane/CorpusT/training/acq/00...	trade	1
13	C:/2019/Mosta/2019 Hassane/CorpusT/training/acq/00...	market	1
14	C:/2019/Mosta/2019 Hassane/CorpusT/test/acq/000961...	price	1
15	C:/2019/Mosta/2019 Hassane/CorpusT/test/acq/000961...	end	1
16	C:/2019/Mosta/2019 Hassane/CorpusT/training/acq/00...	sale	2
17	C:/2019/Mosta/2019 Hassane/CorpusT/test/acq/000961...	increas	1
18	C:/2019/Mosta/2019 Hassane/CorpusT/test/acq/000961...	compani	4

Figure IV.13 : Exemple de bases de données.

## 4-4- WordNet :

Nous avons utilisé WordNet de version 2.0 qui est une base de données lexicographique. Cette dernière est riche et plus générale qui contient tous les domaines, elle est dédiée pour la langue anglaise qui est la langue la plus utilisée dans le monde, il existe d'autre version de WordNet pour d'autres langues. La structure du Wordnet repose sur des ensembles de synonymes appelés synset. Chaque synset représente un sens, un concept de la langue anglaise. Chacun d'eux contient tous les mots synonymes pouvant exprimer le sens auquel il fait référence. Les liens sémantiques ne relient alors pas les mots entre eux mais les synsets aux quels les mots sont affectés. Le choix de WordNet été cause de diverses raisons :

- C'est la base la plus riche et la plus générale qui contient tous les domaines.
- Il utilise la langue anglaise qui est la langue la plus utilisée dans le monde.

Le tableau ci-dessous montre la structure de WordNet d'anglais en nombre de mots, nombre de synsets et nombre de sens.

Position	Mots	Synsets
Nom	117097	81426
Verbe	11488	13650
Adjectif	22141	18877
Adverbe	4601	3644
<b>Total</b>	<b>155327</b>	<b>177597</b>

Table IV.6: Caractéristiques du nombre de mots et de concepts dans WordNet.

#### 4-5- Corpus utilisé :

Dans notre travail, nous avons utilisé le corpus normal en langue anglaise, qui a permis de supprimer les documents présents deux fois, de corriger des erreurs typographiques, déprécier certains formats, et de mieux définir le découpage à considérer pour l'apprentissage et le test.

Il comporte 5 classes avec 25 documents pour l'ensemble d'apprentissage et 22 pour l'ensemble de test.

Corpus	Apprentissage	Test
Acq	5	5
barley	5	5
cocoa	5	5
Dmk	5	3
Grain	5	4

Table IV.7: Répartition des documents du corpus utilisé.

#### 5-Présentation de quelques interfaces de notre application:

##### 5-1- L'interface principale :

Au lancement de l'exécution de l'application une fenêtre est apparue (**Figure IV.14**).



Figure IV.14 : L'interface principale.

### 5-2- Prétraitements effectués sur les corpus d'apprentissage et de test :

Le prétraitement consiste à :

- Parcourir corpus : permet de charger des documents à analyser situés dans de l'ordinateur, choisir une option Apprentissage/Test et les meilleurs termes (j'ai choisi 600 termes).
- Indexation par lucene : lancer indexation des corpus :
  - Convertir les majuscules en minuscules.
  - Enlever les caractères non alphanumériques : les mots sont séparés par des espaces, des signes de ponctuations, des chiffres et les caractères spéciaux...
  - Elimination des mots vides.
- Calcul les fréquences de chaque terme.
- Calcul le temps indexation.

Dans les figures suivantes (IV.15, IV.16) nous présentons les corpus nettoyés.

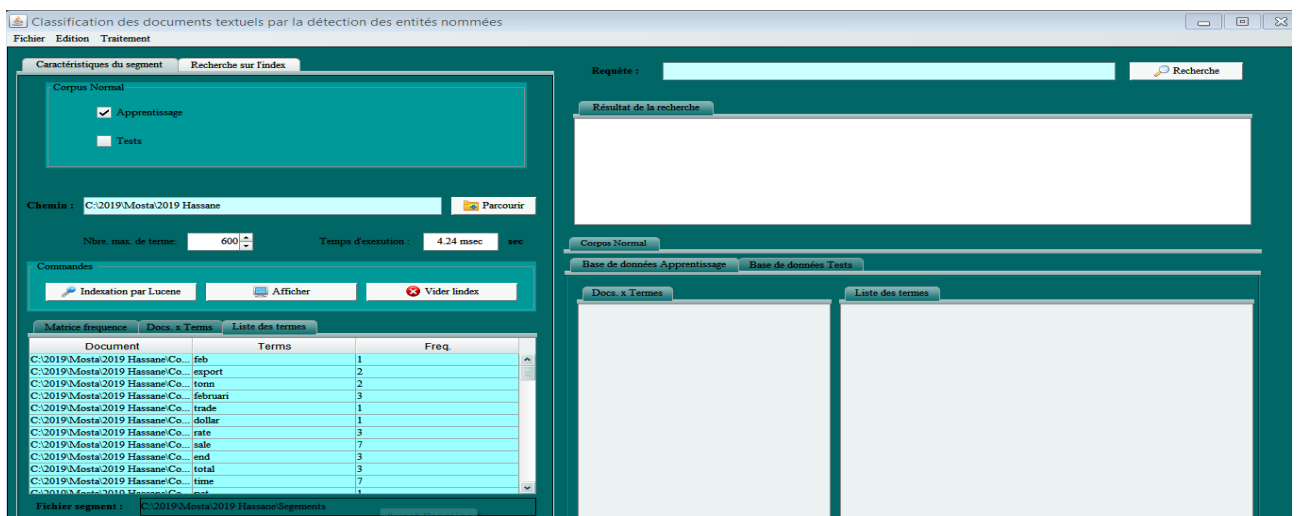


Figure IV.15 : Fenêtre de la représentation liste des termes d'apprentissage.

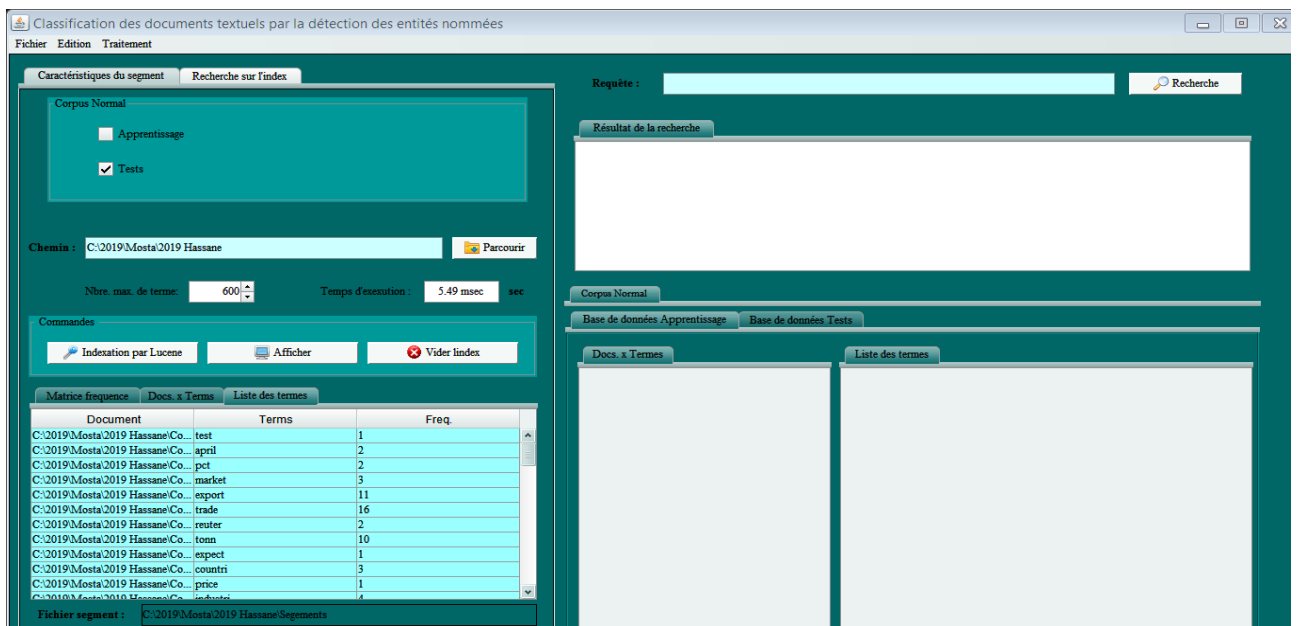


Figure IV.16 : Fenêtre de la représentation liste des termes du test.

### 5-3- Recherche d'information:

La recherche dans la base de données un mot mais la recherche ici est une recherche sémantique c'est-à-dire en indexant le terme (**Figure IV.17**).

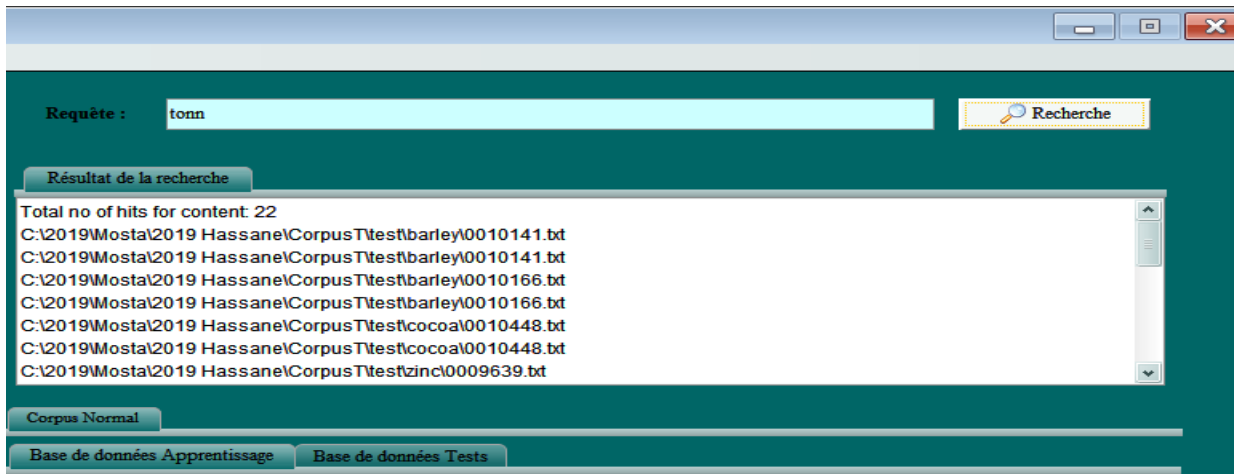


Figure IV.17 : Recherche sémantique.

### 5-4- Traitement WordNet:

Traitement consiste à :

- Nombre d'occurrence : présente l'étape de calcul d'occurrence des termes toujours dans les documents.
- Synonymes : chaque mot avec son synset à l'aide du WordNet.
- Comptabiliser la présence de chaque terme dans le document.
- Similarité entre les documents : Cet enrichissement consiste à calculer des mesures de similarités sémantiques entre concepts afin de pouvoir comparer les concepts qui se rapprochent, on se basant sur un seuil choisi qui est égale à 0.09

Dans les figures suivantes (IV.18, IV.19, IV.20) nous présentons les synonymes et la représentation matricielle d'un corpus, calcule similarité.

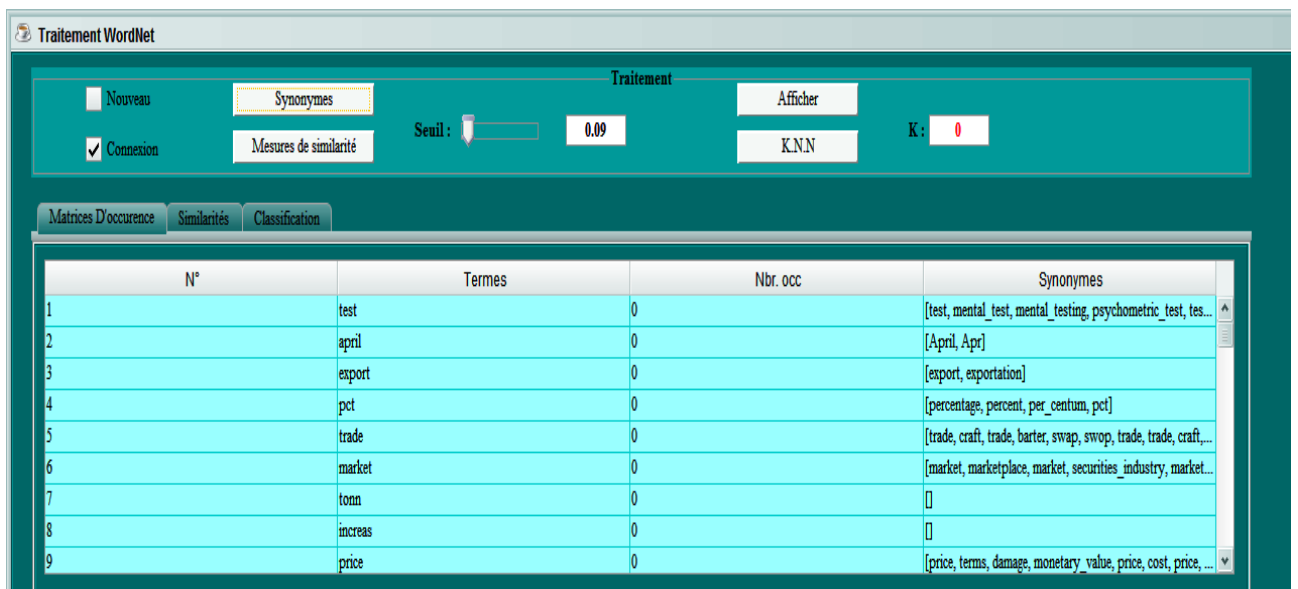


Figure IV.18 : Fenêtre calcul d'occurrence et Représentation des mots avec synset.


Trm/Doc	barley	acq	dmk	lumber	cocoa	alum	grain	trade	platinum	zinc
test	0	0	0	0	0	0	0	0	0	0
april	0	0	0	0	0	0	0	0	0	0
export	1	1	1	1	1	1	1	1	1	1
pot	1	1	1	1	1	1	1	1	1	1
trade	1	1	1	1	1	1	1	1	1	1
market	1	1	1	1	1	1	1	1	1	1
torn	0	0	0	0	0	0	0	0	0	0
increas	0	0	0	0	0	0	0	0	0	0

Figure IV.19 : Matrice de représentation du corpus.

	0010141.tx	0010141.tx	0009613.tx	0010141.tx	0010141.tx	0010141.tx	0010141.tx	0009618.tx	0009613.tx	0010302.tx	0010215.tx	0009613.tx	0010302.tx	0009613.tx	0010302.tx	0009613.tx		
0010141.tx	1	0	0	0	0.2	0.11111111...	0	0	0	0	0	0	0	0	0	0.1	0	
0010141.tx	0	1	0.07692307...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0010141.tx	0	0	0	1	0	0	0	0	0	0	0	0.11111111...	0	0	0	0	0	0
0009613.tx	0	0.07692307...	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0010141.tx	0.11111111...	0	0	0	0.14285714...	1	0	0	0	0	0	0	0	0	0	0	0.25	0
0010141.tx	0.2	0	0	0	1	0.14285714...	0	0	0	0	0	0	0	0	0	0	0.125	0
0010141.tx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0009613.tx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0009618.tx	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0010215.tx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0010302.tx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0010302.tx	0	0	0	0.11111111...	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0010302.tx	0.1	0	0	0	0.125	0.25	0	0	0	0	0	0	0	0	0	0	1	0
0009613.tx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0009613.tx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0009613.tx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0011349.tx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0009613.tx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0009613.tx	0	0.09090909...	0.07692307...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0010302.tx	0	0	0	0.33333333...	0	0	0	0	0	0	0	0	0.11111111...	0	0	0	0	0
0010302.tx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0010141.tx	0	0	0	0.16666666...	0	0	0	0	0	0	0	0	0.125	0	0	0	0	0
0009613.tx	0	0	0	0.11111111...	0	0	0	0	0	0	0	0	0.09090909...	0	0	0	0	0

Figure IV.20: Fenêtre de similarités entre les documents.

### 5-5- Classification des corpus:

- Une fois la mesure de similarité, cette étapes nécessite à cliquer sur le bouton  pour classé le document, on se basant sur un K-plus proche choisi qui est égale à 3.

Document	Classe
001014	barley
000961	acq
001111	dmk

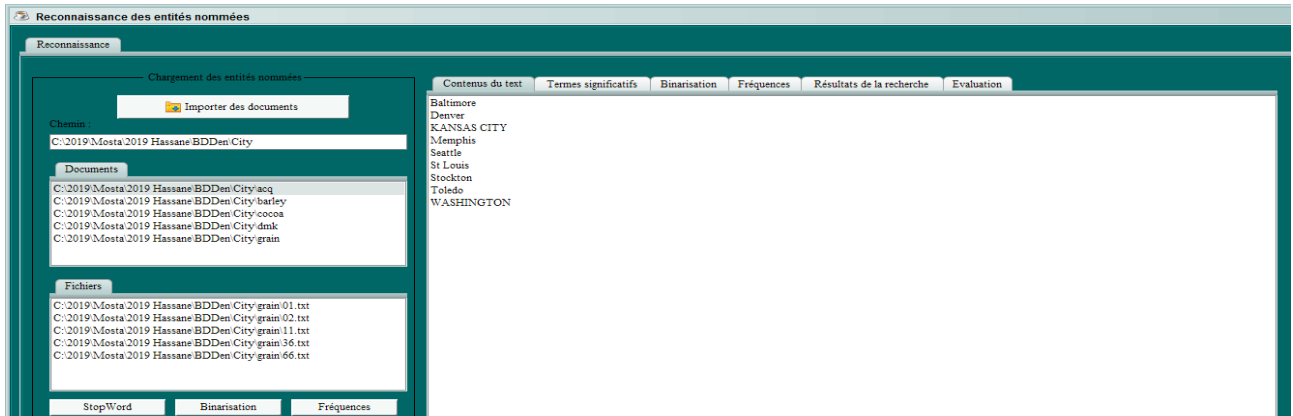
Figure IV.21: Résultat des mesures de classification des documents.

**5-6- Classification des entités nommées :**

La classification consiste à :

- Parcourir ENs : permet de charger des entités nommées à analyser situés dans de l'ordinateur.
- Prétraitement : éliminations des mots vides.

Dans la figure suivante (IV.24) nous présentons les entités nommées nettoyés.



**Figure IV.22:** Fenêtre de la représentation liste des entités nommées.

- Calcul les fréquences de chaque EN et le nombre document pour créer la matrice de contingence.
- Classification les entités nommées (EN → Doc → Classe).

Dans les figures suivantes (IV.25, IV.26) nous présentons calcul des Fréquences et classification des entités nommées.

Reconnaissance des entités nommées									
Reconnaissance									
Chargement des entités nommées									
Contenus du texte									
Termes significatifs									
Binarisation									
Fréquences									
Résultats de la recherche									
Evaluation									
N°	Termes	acq	barley	cocoa	dmk	grain	Freq.	NDoc	
1	Osaka	1	0	0	0	0	1	1	
2	TOKYO	1	2	1	1	0	5	4	
3	CLEVELAND	2	0	0	0	0	2	1	
4	CRANFORD	2	0	0	0	0	2	1	
5	HOUSTON	2	0	0	0	0	2	1	
6	YORK	1	0	0	0	0	1	1	
7	WASHINGTON	1	0	0	0	3	4	2	
8	Portland	1	0	0	0	0	1	1	
9	SUFFIELD	1	0	0	0	0	1	1	
10	PARIS	0	5	0	0	1	6	2	
11	LONDON	0	1	2	0	0	3	2	
12	BRUSSELS	0	1	0	0	0	1	1	
13	York	0	0	3	1	0	4	2	
14	SALVADOR	0	0	1	0	0	1	1	
15	CHICAGO	0	0	1	0	0	1	1	
16	Rio	0	0	1	0	0	1	1	
17	Janeiro	0	0	1	0	0	1	1	
18	ANKARA	0	0	1	3	0	4	2	
19	JAKARTA	0	0	1	1	0	2	2	
20	GHANA	0	0	1	0	0	1	1	
21	FRANKFURT	0	0	0	2	0	2	1	

**Figure IV.23 :** Fenêtre calcul des Fréquences.

N°	Termes	Document	Classes
1	Osaka	05.txt	acq
2	TOKYO	05.txt	acq
3	CLEVELAND	05.txt	acq
4	CLEVELAND	07.txt	acq
5	CRANFORD	07.txt	acq
6	CRANFORD	27.txt	acq
7	HOUSTON	27.txt	acq
8	YORK	28.txt	acq
9	WASHINGTON	28.txt	acq
10	HOUSTON	44.txt	acq
11	Portland	44.txt	acq
12	SUFFIELD	44.txt	acq
13	PARIS	47.txt	barley
14	PARIS	47.txt	barley
15	TOKYO	47.txt	barley
16	PARIS	55.txt	barley
17	TOKYO	55.txt	barley
18	LONDON	63.txt	barley
19	BRUSSELS	63.txt	barley
20	PARIS	77.txt	barley
21	PARIS	77.txt	barley

Figure IV.24 : Résultat de classification des entités nommées.

- Dans cette étape, rechercher les documents qui ont ces entités nommées.

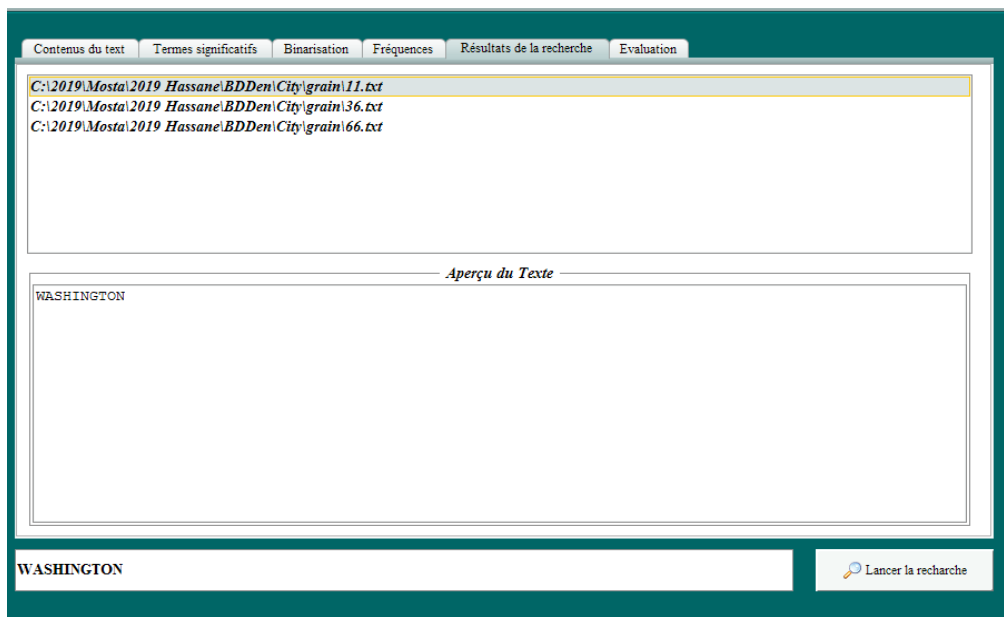


Figure IV.25 : Résultat de la Recherche.

**5-7- Evaluation des performances:**

**5-7-1- Mesures de classification des corpus :**

- Dans cette étape j’ai utilisé la table de confusion résulte de la confrontation entre la classe 1 et la classe 0 (sur l’échantillon test c’est mieux) pour calculer les valeurs de rappel et de précision et prend les résultats.
  - Classe 0: Le nombre des termes non attribués à une catégorie (120 termes).

- Classe 1 : le nombre des termes attribués à une catégorie (100 termes).

Figure IV.26 : Choix des termes.

- Calcul rappel et précision, temps d'exécution pour tirer des conclusions sur les classifiées.

Figure IV.27 : Résultat des mesures de classification des corpus.

### 5-7-2- Mesures de classification des entités nommées :

- Dans cette étape j'ai utilisé la matrice de contingence Pour évaluer un système de classification.
  - Vrai Positif (VP) : Le nombre de documents attribués à une catégorie convenablement.
  - Faux Positif (FP) : Le nombre de documents attribués à une catégorie inconvenablement.
  - Faux Négatif (FN) : Le nombre de documents inconvenablement non attribués.
  - Vrai Négatif(VN) : Le nombre de documents non attribués à une catégorie convenablement.



Contenus du text	Termes significatifs	Binarisation	Fréquences	Résultats de la recherche	Evaluation
Fichiers	VP	VN	FP	FN	
05.txt	0	1	1	0	
07.txt	1	0	1	0	
27.txt	1	0	1	0	
28.txt	1	0	1	0	
44.txt	1	0	1	0	
47.txt	0	1	0	1	
55.txt	0	1	0	1	
63.txt	1	0	1	0	
77.txt	0	1	1	0	
00.txt	0	1	1	0	
03.txt	0	1	1	0	
18.txt	0	1	1	0	
94.txt	0	1	0	1	
98.txt	1	0	1	0	
14.txt	0	1	1	0	
20.txt	0	1	0	1	
42.txt	0	1	1	0	
68.txt	1	0	1	0	
74.txt	1	0	1	0	
01.txt	1	0	1	0	
02.txt	1	0	1	0	

Figure IV.28 : Matrice de contingence.

- Dans notre évaluation, nous avons choisi les métriques qui nous permettront d'évaluer les résultats de nos travaux et ainsi mesurer les performances du système proposé (calcul rappel et précision, temps d'exécution).

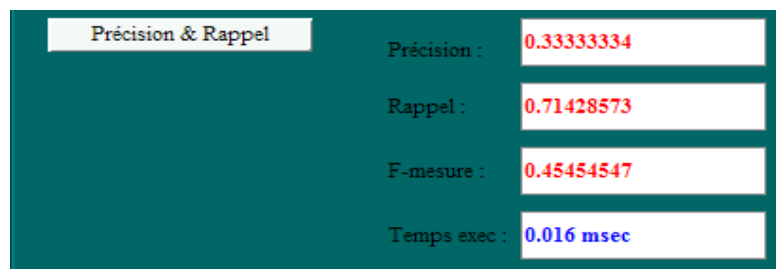


Figure IV.29 : Résultat des mesures de classification des entités nommées.

Les deux figures (IV.27, IV.29) représenté les résultats de classifications des corpus et entités nommées respectivement.

Comme nous observent dans les résultats précédents que les performances de classification des entités nommées mieux sur la classification des corpus.

	Temps d'exécution (ms)	Precision	Rappel	F_measures
<b>Corpus</b>	15.257	0.6	0.003080082135523614	0.006128702757916242
<b>Entités nommées</b>	0.016	0.33333334	0.71428573	0.45454547

Table IV.8: Comparaison les résultats.

Nous discutons les résultats obtenus, pour le temps d'exécution, la classification des corpus prend 15.257 ms Par contre, ce temps d'exécution est 0.016 ms dans le cas des entités nommées. On remarque une légère dégradation des performances de la méthode des corpus par rapport à la méthode qui utilise les entités nommées (0.4545 entités nommées par contre 0.00612 pour F\_mesure).

## **6- Conclusion :**

Dans ce dernier chapitre, on a présenté une vue complète sur notre système à partir des différentes interfaces capturées, ainsi les outils utilisés pour l'implémentation.

Il est nécessaire de mesurer les performances d'un filtre sur un ensemble des entités nommées pour d'une part limiter l'impact des erreurs d'annotations.

On conclut que les résultats obtenir on peut dire que la méthode de k-Nearest Neighbor (K-NN) donne des bonnes résultats avec entités nommées que un corpus.

## Conclusion générale :

La Recherche d'Information a pour objectif de fournir à un utilisateur un accès facile à l'information qui l'intéresse, cette information étant située dans une masse de documents textuels. Afin d'atteindre cet objectif, un système de recherche d'information doit représenter, stocker et organiser l'information puis fournir à l'utilisateur les éléments correspondant au besoin d'information exprimé par sa requête.

Les entités nommées (personnes, lieux, organisations, dates, expressions numériques, marques, fonctions, etc.) sont sollicitées afin de catégoriser, indexer ou, plus généralement, manipuler des contenus.

La reconnaissance d'entités nommées est une tâche dont l'objectif est d'extraire et de typer des éléments informationnels à partir d'un texte donné. Des systèmes de reconnaissance de noms propres à base de ressources linguistiques.

Dans ce travail, nous avons commencé à définir les concepts de base de la recherche d'information (document, collection de document, ...) ensuite nous sommes passés à expliquer le fonctionnement du système de recherche d'information tout en incluant les étapes de son processus, puis nous avons cité les modèles de **RI** (modèle booléen, vectoriel, probabiliste) en suite nous sommes passés à la recherche d'information sur le web en citant les différents types d'outils de recherche sur internet, ensuite le processus de l'extraction d'information avec son principe d'extraction et l'annotation sémantique.

Nous avons expliqué son rôle des entités nommées et leurs différentes formes, puis on cite quelques problèmes et difficultés majeures dans certains domaines (indexation, recherche, ...), ensuite nous sommes passés à expliquer les typologies d'entités nommées (noms propres de personnes, lieux, ...) puis ont défilé dans l'annotation d'entités nommées et ces éléments essentiels, enfin, puis notre intérêt qui il est la reconnaissance et la détection des entités nommées tout en incluant les approches (approche orientée connaissance, approches orientées connaissances).

Ensuite, nous avons présenté la conception classification des documents textuels qui est basée sur les méthodes de classification supervisée et différents modèles de classifier (SVM, réseau neuronal, K-NN, ..), puis on a passé à la classification des corpus.

D'après, nous avons déterminé un processus de classification des documents et reconnaissance des entités nommées et nous avons comparé et discuté les résultats obtenus par les deux classifications.

## Références :

- [1] Baziz. M, "Indexation conceptuelle guidée par ontologie pour la recherche d'information", Thèse de doctorat en informatique, Université Paul Sabatier de Toulouse, 2005.
- [2] AUSSENAC-GILLES. N, BOUGHANEM. M, "Désambiguïsation et Expansion de Requêtes dans un SRI, Etude de l'apport des liens sémantiques". Revue des Sciences et Technologies de l'Information (RSTI) série ISI, Hermes : Paris, V. 8, N. 4/2003, 113-136, Déc. 2003.
- [3] Ingwersen. P, "Polyrepresentation of information needs and semantic entities: elements of a cognitive theory for information retrieval interaction". In Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval., pages 101-110, 1994.
- [4] Ricardo B Y., Berthier R N. Modern information retrieval, "ACM Association for Computing Machinery", (1999).
- [5] Salton G., "A comparaison between manual and automatic indexing methods", In Proceedings of Journal of American documentation, 1971.
- [6] Abbes R., "Filtrage et agrégation d'informations vitales relatives à des entités», Thèse de l'Université Toulouse 3 Paul Sabatier, soutenue le 11/12/2015.
- [7] Salton. G, "Automatic text processing: The transformation, analysis and retrieval of information by computer", Addison-Wesley publishing, MA, 1989.
- [8] Sparck-Jones. K, Needham R., "Automatic theme classification and retrieval", Information Processing and Management, 4: 91,100, 1972.
- [9] Maron. M, and Kuhns J. "On relevance Probabilistic indexing and information retrieval". Journal of the ACM, vol. 7: p. 216-244, 1960.
- [10] Robertson. S, Maron. M, and W. Cooper Probability of relevance: a unification of two competing models for document retrieval. Information Technology: Research and Development, 1: p. 1-21, 1982.
- [11] Tebri. H, "Formalisation et spécification d'un systèmes de filtrage incrémental d'information". PhD thesis, Toulouse: Université Paul Sabatier, 2004.
- [12] J Belkin. N, and Croft. W, "Information filtering and information retrieval: two sides of the same coin? Communications of the ACM", 35(12), Décembre 1992.
- [13] HARRATHI. R, "Recherche d'information conceptuelle dans les documents semi-structurés", GRADE DE DOCTEUR SPECIALITE : INFORMATIQUE, Soutenue à Lyon le 29 Septembre 2010
- [14] SANG E. F. T. K. & MEULDER F. D. "Introduction to the CoNLLshared task:Language-independent named entity recognition. In Proceedings of the seventh

conference on Natural language learning at HLT-NAACL", Edmonton, Canada, 2003 (CONLL'03), p. 142–147,

[15] Lin, B., Shah, R., Frederking, R. et Gershman, A. Cone "Metrics for automatic evaluation of named entity co-reference resolution". In International Conference on Computational Linguistics (COLING'10), pages 931–939, Beijing, China.

[16] Friburger, N. "Reconnaissance automatique des noms propres : application à la classification automatique de textes journalistiques". Thèse de doctorat, Université François-Rabelais Tours, France.

[17] Semeval: "Metonymy resolution at semeval", In Proc. Of Sem Eval, ACL, Prague, 2007.

[18] "Mesures d'évaluation pour entités nommées structurées". In Ateliers joints QDC'2011 - EvalECD'2011. Évaluation des méthodes d'Extraction de Connaissances dans les Données, Brest, France, janvier 2011, P.49-62.

[19] MCDONALD, D. "Internal and external evidence in the identification and semantic categorization of proper names", Corpus processing for lexical acquisition, (1996), pages 21-39.

[20] ISOZAKI, H. ET KAZAWA, H. "Efficient support vector classifiers for named entity recognition". (2002). In Proceedings of the 19th international conference on Computational linguistics-Volume 1, pages 1–7. Association for Computational Linguistics.

[21] Sylvain Goulet, "Technique d'identification d'entités nommées et de classification non-supervisée pour des requêtes de recherche web à l'aide d'informations contenues dans les pages web", Mémoire pour obtention du grade de maître ès sciences, Faculté des sciences Université de Sherbrooke. Sherbrooke, Quebec, Canada, 2014.

[22] Hayes. P, S.P.Weinstein "Construe/Tis: A system for content-based indexing of a database of news stories"

[23] Chen. Y. Balke, W.-T., Xu, J., Xu, W., Jin, P., Lin, X., Tang, T. et Hwang, E. Web-Age Information Management: WAIM 2014 International Workshops: BigEM, HmdBD, DaNoS, HRSUNE, BIDASYS, Ma cau, China, June 16-18, 2014, Revised Selected Papers, volume 8597. Springer.

[24] Tay, b., j.k. hyun, and s.o.s. oh, computational and mathematical methods in medicine, <[https://www.researchgate.net/figure/260397165\\_fig7\\_pseudocode-for-knn-classification](https://www.researchgate.net/figure/260397165_fig7_pseudocode-for-knn-classification)>, jan 2014.