

République Algérienne Démocratique et Populaire  
Université Abou Bakr Belkaid– Tlemcen  
Faculté des Sciences  
Département d'Informatique

**Mémoire de fin d'études  
pour l'obtention du diplôme de Master en Informatique**

*Option: Système d'Information et de Connaissances (S.I.C)*

## ***Thème***

***Amélioration du produit scalaire via les  
mesures de similarités sémantiques dans le  
cadre de la catégorisation des textes***

**Réalisé par :**

- *Rimouche Nour El Houda*
- *Hachemi Hadjira*

**Présenté le 15 Décembre 2015 devant le jury composé de :**

- *M.Smahi Smail* (Président)
- *M.Bentallah Mohammed Amine* (Encadreur)
- *Mme.Elyebedri Zineb* (Examineur)
- *Mme.Halfaoui Amal* (Examineur)

# REMERCIEMENTS

**E**n préambule nous tenons à remercier ALLAH qui nous a aidés et nous à donner la patience et le courage durant cette année.

**N**ous tenons à remercier tous ceux qui nous ont aidés, d'une manière ou d'une autre, pendant ce travail d'étude et de recherche.

**N**ous tenons d'abord à remercier très chaleureusement Monsieur **BENTAALLAH MOHAMMED AMINE** qui nous a permis de bénéficier de son encadrement. Les conseils qu'il nous a prodigués, la patience, la confiance qu'il nous a témoignés ont été déterminants dans la réalisation de notre travail de recherche.

**N**os vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre modeste travail Et de l'enrichir par leurs propositions.

**N**os remerciements s'étendent également à tous nos enseignants durant les années des études.

**M**erci à Tous et à Toutes.

# Dédicaces

**JE DÉDIE CE MÉMOIRE**

**À MES CHERS PARENTS MA MÈRE ET MON PÈRE  
POUR LEUR PATIENCE, LEUR AMOUR, LEUR  
SOUTIEN ET LEURS ENCOURAGEMENTS.**

**À MA SŒUR IMEN ET MON FRÈRE ALI**

**À TOUS MES AMIS ET À TOUS MES CAMARDES  
SANS OUBLIER TOUS MES PROFESSEURS  
DE L'ENSEIGNEMENT SUPÉRIEUR**

**ET À TOUT CEUX QUI M'ONT AIDÉ DANS  
L'ÉLABORATION DE CE TRAVAIL.**

**HOUDA**

# Dédicaces

**JE DÉDIE CE MÉMOIRE**

**À MES CHERS PARENTS MA MÈRE ET MON PÈRE  
POUR LEUR PATIENCE, LEUR AMOUR, LEUR  
SOUTIEN ET LEURS ENCOURAGEMENTS.**

**À MON MARI **AMAR****

**À MES SŒURS **ASMA** ET **IMANE** ET MON FRÈRE  
**ALI****

**À TOUS MES AMIS ET À TOUS MES CAMARDES  
SANS OUBLIER TOUS MES PROFESSEURS  
DE L'ENSEIGNEMENT SUPÉRIEUR**

***ET* À TOUT CEUX QUI M'ONT AIDÉ DANS  
L'ÉLABORATION DE CE TRAVAIL.**

**Hadjira**

# Table des matières

Introduction générale.....	8
Chapitre I : Classifications des textes	
II. Définition :.....	10
III. Processus de la catégorisation de textes : .....	11
III.1.1.Représentation en sac de mots (bag of words) :.....	12
III.1.2.Représentation avec les racines lexicales : .....	12
III.1.3.Représentation avec les lemmes :.....	12
III.1.4.Représentation avec les n-gramme :.....	13
III.1.4.Représentation par phrases .....	13
III.1.5.Représentation conceptuelle : .....	14
III.2.La pondération des termes : .....	14
III.2.1Mesure TF (Term Frequency): .....	15
III.2.2Mesure TFIDF (Term Frequency Inverse Document Frequency): .....	15
III.2.3.La mesureTFC .....	15
III.3.La réduction de la taille du vocabulaire : .....	16
III.4.Choix de classificateur : .....	17
III.4.1.Machine à vecteur support : .....	17
III.4.2.Les k plus proches voisins :.....	19
III.4.3.Méthode de Rocchio : .....	19
III.4.4.Naïve bayes : .....	20
III.4.5.Les arbres de décision : .....	20
III.4.6.Les réseaux de neurone :.....	21
IV. Evaluation du processus de catégorisation : .....	22
V. Les applications de la catégorisation des textes : .....	23
VI. Problèmes de la catégorisation de textes : .....	24
VII. Conclusion : .....	25
<u>Chapitre II: Les mesures de similarités sémantiques et l'amélioration du produit scalaire</u>	
I. Introduction :.....	27
II. Les mesures de similarités sémantiques : .....	27
II.1.Définition :.....	27
II.2.Objectifs :.....	27
II.3.Les Différentes approches de la similarité sémantique : .....	28

# Table des matières

II.3.1.Approche basé sur les arcs : .....	28
II.3.1.1.Mesure de Wu & Palmer : .....	28
II.3.2.Approche basé sur les nœuds (contenue informationnel) : .....	29
II.3.2.1 .Mesure de Resnik : .....	29
II.3.2.2. Hirst & Onge .....	30
II.3.3.Hybride : .....	30
II.3.3.1.Jiang & Conrath : .....	31
II.3.3.2.Leacock et Chodorow : .....	31
III. Architecture de notre travail : .....	31
III.1. Les étapes de représentation : .....	34
III.1.1. Représentation en sacs de mots : .....	34
III.1.2.Transformation des mots en synsets : .....	37
III.1.2.1.Definition : .....	37
III.1.2.2.Représentation conceptuelle : .....	38
III.1.3.Enrichissement : .....	39
III.1.4.Classification : .....	41
IV. Exemple de déroulement de notre programme : .....	42
IV.1.Indexation .....	42
IV.2.Classification .....	45
V. Environnement et outils de développement : .....	48
V.1.Language JAVA : .....	48
V.2.Evironnement de développement : .....	49
V.3.WordNet : .....	49
V.4.JWNL : .....	50
V.I.Conclusion : .....	50
Conclusion générale.....	50

# Introduction générale

# Introduction générale

La révolution de l'internet a fait exploser les informations textuelles, qui sont un patrimoine vivant des entreprises, des administrations et des particuliers, il est devenu indispensable aux utilisateurs du web de trouver les documents pertinents, pour cette raison il devient de plus en plus important de disposer de solutions efficaces pour conserver, chercher et classer ces informations, afin d'assister les utilisateurs à trouver leurs besoins et faciliter leur travail dans certaines tâches qui sont devenues impossible à traiter manuellement.

Donc il est très intéressant de compter sur une application automatique qui est la classification et la catégorisation des textes qui a comme objectif de rassembler les textes similaires selon certain critères (par thèmes, par auteurs, par langue, par sens ou par d'autres critères de classification) au sein d'une même classe.

Notre projet traite l'évaluation de l'utilisation des mesures de similarité sémantique pour la classification des textes, qui consiste à représenter les documents classés et non classés par une bonne méthode de représentation. L'objectif principal est de calculer la mesure de similarité entre les documents classés et le document non classé. Et aussi notre projet permet de comparer ses résultats avec les résultats obtenus en utilisant les méthodes de la classification statistiques.

Nous avons décomposé notre mémoire en deux chapitres. Le premier chapitre vise à présenter le processus de la catégorisation des textes et les principales phases de ce dernier, sans oublier les applications liées à la catégorisation des textes.

Le deuxième chapitre se basent sur deux étapes la première étape concerne les mesures de similarités sémantiques et leurs approches, et la deuxième étape consiste à exposer la description des approches implémentées, ainsi que l'architecture et le déroulement de notre projet et nous terminons par présenté les résultats obtenus



# Chapitre I

## Classifications des textes

## I. Introduction :

La Catégorisation de textes (C.T) est aujourd'hui un domaine de recherche bien établi et très actif. Les travaux portent depuis une quinzaine d'année sur les systèmes avec apprentissage des catégories à partir de corpus pré- étiquetés.

Dans ce chapitre, nous présentons d'abord une définition de la catégorisation des textes; ainsi que le processus de C.T qui est constitué de plusieurs étapes : la représentation des textes, la pondération des termes , la réduction de la taille du vocabulaire , choix de classificateur , évaluation du processus de catégorisation(rappel, précision).

Nous exposons par la suite les applications de la CT et enfin nous citons les principales difficultés liées à la C.T.

## II. Définition :

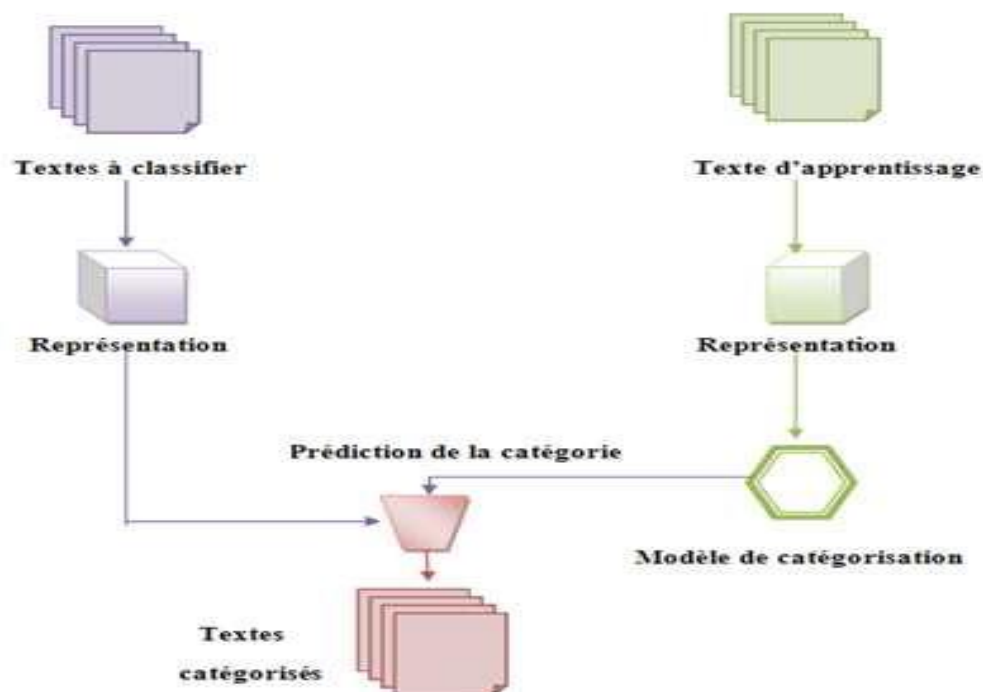
Plusieurs définitions de la C.T ont vu le jour depuis son apparition, nous citons dans ce contexte les deux définitions suivantes :

- **Définition1** : [1] La CT est une relation bijective qui consiste à "chercher une liaison fonctionnelle entre un ensemble de textes et un ensemble de catégories (étiquettes, classes)".
- **Définition2** : [2] La CT est le processus qui consiste à associer une valeur booléenne à chaque paire  $(d_j, c_i) \in D \times C$ , où D est l'ensemble des textes et C est l'ensemble des catégories. La valeur V(Vrai) est alors associée au couple  $(d_j, c_i)$  si le texte  $d_j$  appartient à la classe  $c_i$  tandis que la valeur F (Faux) est associée dans le cas contraire.

Le but de la catégorisation des textes est de construire une procédure (modèle, classificateur) notée :  $\Phi : D \times C \rightarrow \{V, F\}$  qui associe une ou plusieurs étiquettes (catégories) à un document  $d_j$

### III. Processus de la catégorisation de textes :

La **figure I.1** résume le processus de catégorisation des textes qui comporte deux phases : l'apprentissage et le classement.



**Figure I.1** : Processus de la catégorisation des textes [1]

Comme il vient d'être mentionné, le but de la catégorisation automatique des textes est d'apprendre à une machine à classer un texte dans la bonne catégorie en se basant sur son contenu. Pour identifier la catégorie d'un texte, ou la classe à laquelle un texte est associé, un ensemble d'étapes est habituellement suivies.

#### III.1. La représentation des textes :

La représentation des textes est une étape très importante dans le processus de C.T, pour cela il est nécessaire d'utiliser une technique de représentation efficace permettant de représenter les textes sous une forme exploitable par la machine. La représentation la plus couramment utilisée est celle du modèle vectoriel [3] dans laquelle chaque texte est représenté par un vecteur de  $n$  termes pondérés. Les différentes méthodes qui existent pour la représentation des textes sont :

### III.1.1.Représentation en sac de mots (bag of words) :

Les textes sont transformés simplement en vecteurs dont chaque composante représente un mot [21]. Utiliser les mots comme termes a comme avantage d'exclure toute analyse grammaticale et toute notion de distance entre les mots, mais présente plusieurs inconvénients qui sont les suivants :

- les mots composés allemands peuvent être très complexes, ex : *Lebensversicherungsgesellschaftsangestellter* (employé d'une société d'assurance vie).
- le chinois et le japonais ne séparent pas les mots par des espaces, ce qui peut mener à plusieurs segmentations.
- l'arabe et l'hébreu sont écrits de droite à gauche, mais certains éléments tels que les nombres sont écrits de gauche à droite.

### III.1.2.Représentation avec les racines lexicales :

Cette méthode consiste à remplacer les mots du document par leurs racines lexicales, et à regrouper les mots de la même racine dans une seule composante. Ainsi, plusieurs mots du document seront remplacés par la même racine, cette méthode peut être réalisée en utilisant un des algorithmes les plus connus pour la langue anglaise qui est l'algorithme de Porter [4] de normalisation de mots qui sert à supprimer les affixes de ces derniers pour obtenir une forme canonique. Néanmoins la transformation automatique d'un mot à sa racine lexicale peut engendrer certaines anomalies. En effet, une racine peut être commune pour des mots qui portent des sens différents tel que les mots jour, journalier, journée ont la même racine « jour » mais se rendent à trois notions différentes, cette représentation dépend aussi de la langue utilisée.

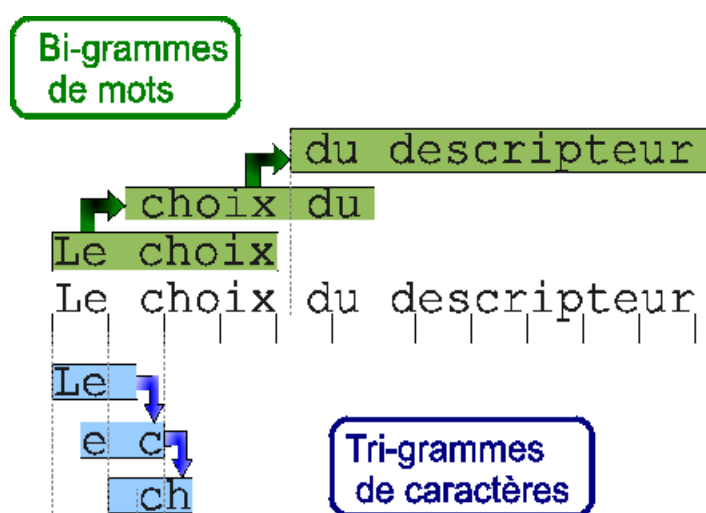
### III.1.3.Représentation avec les lemmes :

La lemmatisation consiste à utiliser l'analyse grammaticale afin de remplacer les verbes par leur forme infinitive et les noms par leur forme au singulier. En effet, Un mot donné peut avoir différentes formes dans un texte, mais leur sens reste le même. Par exemple, les mots jouons, joueurs, jouet seront remplacés par leurs lemmes : « jouer », « joueur » et « jouet » selon le contexte. Cette représentation est simple mais elle peut causer une perte d'informations donnée par le contexte

nécessaire à la distinction des lemmes polysémiques (possèdent plusieurs sens) et la présence de synonymes, considérés comme des lemmes différents même s'ils font référence au même concept. [5]

#### III.1.4.Représentation avec les n-gramme :

Cette méthode consiste à représenter le document par des n-grammes. Le n-gramme est une séquence de n caractères consécutifs. Elle consiste à découper le texte en plusieurs séquence de n caractère en se déplaçant avec une fenêtre d'un caractère. Nous présentons ci-dessous **FigureI.2** les deux types de n-grammes, caractères et mots



**FigureI.2** : Exemple de N-grammes de mots et de caractères.

Cette technique présente plusieurs avantages. Les n-grammes capturent automatiquement les racines des mots les plus fréquents sans passer par l'étape de recherche des racines lexicales, de plus c'est une méthode indépendante de la langue. [6]

#### III.1.4.Représentation par phrases

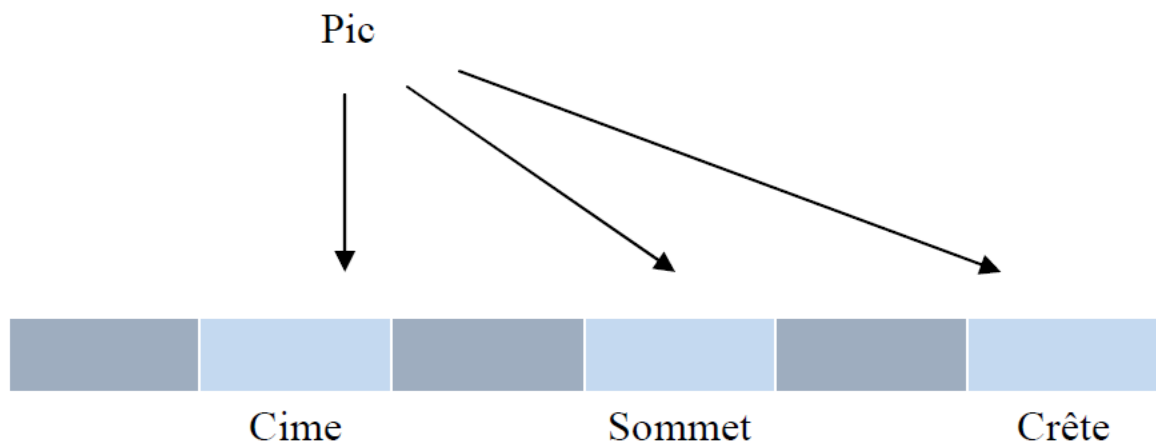
Un certain nombre de chercheurs proposent d'utiliser les phrases comme unité de représentation au lieu des mots comme le cas dans la représentation « sac de mot », puisque les phrases sont plus informatives que les mots seuls, par exemple « recherche d'information », « world wide web », ont un degré plus petit d'ambiguïté

que les mots constitutifs, et aussi que les phrases ont l'avantage de conserver l'information relative à la position du mot dans la phrase. [7] [8]

### III.1.5.Représentation conceptuelle :

La représentation conceptuelle sert à représenter le document sous forme d'un ensemble de concepts, ces derniers peuvent être capturés en utilisant les réseaux sémantiques

Prenons l'exemple de la **Figure I.3** en peut regrouper les trois termes (cime, sommet, crête) du vecteur dans le concept pic.



**FigureI.3** : La représentation conceptuelle du mot « pic ».

Cette méthode a comme avantage de réduire l'espace de représentation car les mots qui sont synonymes partagent le même concept. Cependant, l'inconvénient majeur de cette représentation est qu'il n'existe pas des bases lexicales pour toutes les langues. [9]

### III.2.La pondération des termes :

La pondération des termes permet de mesurer l'importance d'un terme dans un document. Cette importance est souvent calculée à partir de considérations et interprétations statistiques. L'objectif est de trouver les termes qui représentent le

mieux le contenu d'un document. Pour calculer la pondération on distingue les méthodes suivantes :

### III.2.1 Mesure TF (Term Frequency):

Cette mesure est proportionnelle à la fréquence du terme dans le document (pondération locale). Elle peut être utilisée telle quelle ou selon plusieurs déclinaisons ( $\log(tf)$ , *présence/absence*, . . .).

### III.2.2 Mesure TFIDF (Term Frequency Inverse Document Frequency):

- **idf (Inverse of Document Frequency) :**

$$\text{idf} = \log\left(\frac{N}{Df}\right)$$

Ou :

**Df:** Le nombre de documents contenant le terme.

**N :** Le nombre total de documents de la base documentaire.

Un terme qui apparaît souvent dans la base documentaire ne doit pas avoir le même impact qu'un terme moins fréquent. La mesure TFIDF est une bonne approximation de l'importance du terme dans le document, particulièrement dans les corpus de documents de taille homogène. La mesure TFIDF est calculé comme suit :

$$TFIDF(T, D) = TF(T, D) \cdot \log\left(\frac{N}{Df(T)}\right)$$

Ou

**TF(T, D) :** la fréquence du terme dans le document.

### III.2.3. La mesure TFC

Le codage  $TF \times IDF$  ne corrige pas la longueur des documents. Pour ce faire, le codage TFC est similaire à celui de  $TF \times IDF$  mais il corrige les longueurs des textes par la normalisation en cosinus, pour ne pas favoriser les documents les plus longs.

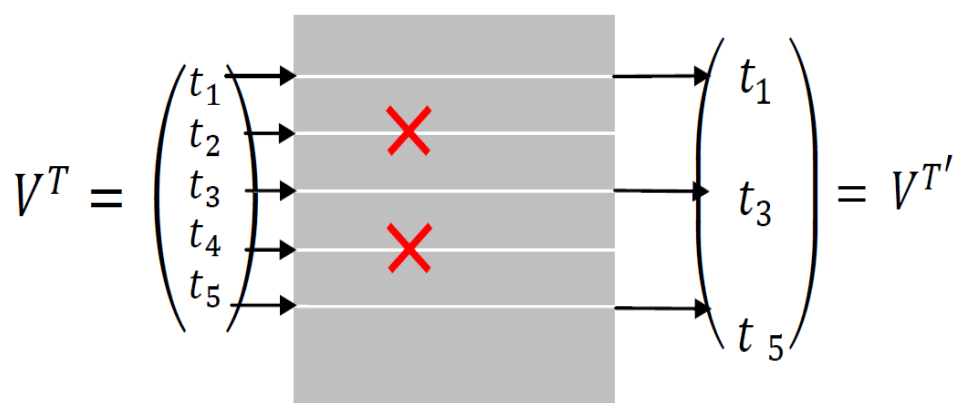
$$TFC = (t_k, d) = \frac{TF \times IDF(t_k, d)}{\sqrt{\sum^{|r|} (TF \times IDF(t_s, d))^2}}$$

### III.3. La réduction de la taille du vocabulaire :

Le problème qu'on doit inévitablement résoudre est celui de la taille du vocabulaire car si on utilise tous les mots présents dans les documents de l'espace d'apprentissage, on se retrouve face à un espace vectoriel ayant une dimension très large. Le traitement d'un tel espace nécessitera beaucoup de mémoire et de temps de calcul et pourra nous empêcher d'utiliser des algorithmes d'apprentissage plus puissants. Pour résoudre ce problème on utilise les techniques de réduction de la taille du vocabulaire qui ont pour objectif de sélectionner ou d'extraire un sous-ensemble optimal de caractéristiques pertinentes pour un critère fixé. Ces techniques de réduction sont classées comme suit :

- **Sélection d'attributs :**

La réduction de la taille de l'espace d'apprentissage par la méthode de sélection d'attributs vise à diminuer la taille de l'espace d'apprentissage de  $|T|$  à une taille  $|T'| \ll |T|$  en sélectionnant uniquement un sous-ensemble des attributs existants.



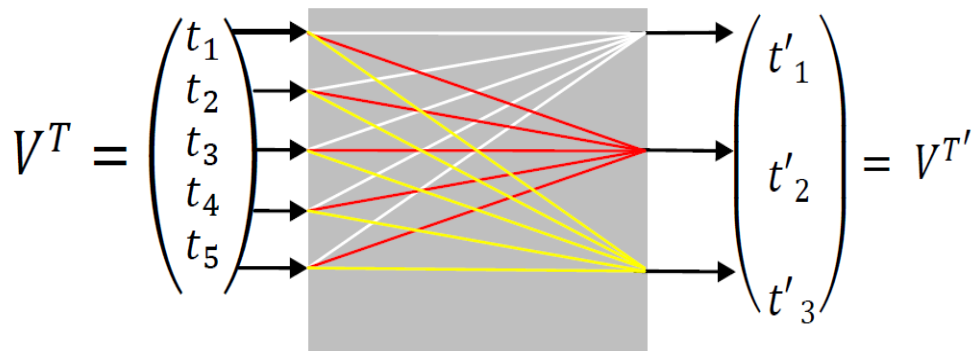
**Figure I.4** Sélection des attributs

Pour simplifier les choses, considérons la **Figure I.4** sélection des attributs où on dispose d'un ensemble d'attributs  $V^T = \{t_1, t_2, t_3, t_4, t_5\}$  réduit en  $V^{T'}$  par le biais d'une sélection. On constate que les attributs n'ont subi aucune transformation.  $V^T$  est réduite en  $V^{T'}$  sélectionnant uniquement un sous-ensemble des attributs de  $V^T$ .



- **Extraction d'attribut :**

Contrairement aux techniques de sélection d'attributs qui visent à proposer par sélection un sous ensemble des attributs existants, l'extraction des attributs a, par définition, pour objectif de proposer, via une synthétisation, un sous ensemble  $|T'| \ll |T|$  composé de nouveaux attributs à partir des attributs existants. Ce processus consiste à créer à partir des attributs originaux un sous ensemble d'attributs synthétiques qui maximise l'efficacité de la classification et qui élimine les problèmes liés aux synonymies, homonymies, et polysémie. [1]



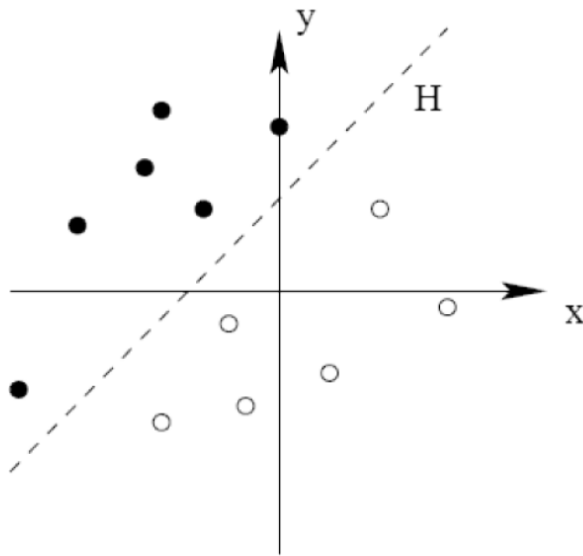
**FigureI.5** :Extraction d'attributs

### III.4.Choix de classificateur :

La catégorisation de textes comporte un choix de technique d'apprentissage (ou classificateur) disponibles. Parmi les méthodes d'apprentissage les plus souvent utilisées figurent : l'analyse factorielle discriminante, la régression logistique, les réseaux de neurones, les plus proches voisins, les arbres de décision, les réseaux bayésiens, les machines à vecteurs supports et, plus récemment, les méthodes dites de *boosting*.

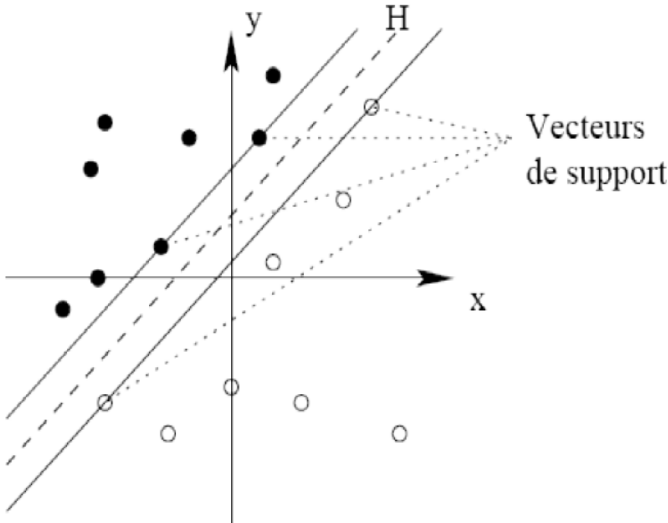
#### III.4.1.Machine à vecteur support :

Le but de SVM est de trouver un classificateur qui sépare au mieux les données et maximise la distance entre ces deux classes. Ce dernier est un classificateur linéaire appelé hyperplan. Comme montré dans la **FigureI.6**, cet hyperplan sépare les deux ensembles de points.



**Figure I.6** : la séparation du l'hyper plan par les SVM

Les points les plus proches, qui seuls sont utilisés pour la détermination de hyperplan, sont appelés **vecteurs de support** (voir **Figure I.7**).



**Figure I.7** : Les vecteurs de support

### III.4.2. Les k plus proches voisins :

C'est une méthode très connue dans le domaine de la catégorisation des textes. L'idée de K-plus proches voisins est de représenter chaque texte dans un espace vectoriel, dont chacun des axes représente un élément textuel (peut être un mot sous sa forme brute ou sous une forme lemmatisée). [10]

L'algorithme de catégorisation de K-plus proches voisins est présenté comme suit :

Algorithme : **algorithme de classification par K-PPV**

**Paramètre** : le nombre K de voisin

**Contexte** : un échantillon de T 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 ces K exemples en une classe C.

Fin pour

**Fin**

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

Le choix du paramètre K est primordial pour le bon fonctionnement de cette méthode. [10]

### III.4.3. Méthode de Rocchio :

Cette méthode est facile à implanter et efficace pour des catégorisations où un texte ne peut appartenir qu'à une seule catégorie. Mais elle n'est pas très efficace quand un texte peut appartenir à plusieurs catégories et certains documents du corpus d'apprentissage appartenant à une catégorie  $C_i$  initialement ne seraient pas classés dans  $C_i$  par le classificateur. La méthode de **Rocchio** se base sur la création de profils de catégorie. Le poids des termes est calculé lors de l'apprentissage en fonction des apparitions de ces termes d'une part dans les documents appartenant à la catégorie et d'autre part dans ceux n'y appartenant pas. [11]

$$w_{ki} = \alpha \cdot \sum_{t_j \in POS_i} \frac{w_{kj}}{|POS_i|} - \beta \cdot \sum_{t_j \in NEG_i} \frac{w_{kj}}{|NEG_i|}$$

Avec  $POS_i$  l'ensemble des documents de  $T_r$  appartenant à la catégorie  $C_i$  et  $\overline{POS}_i$  l'ensemble des documents de  $T_r$  n'appartenant pas à la catégorie  $C_i$ . Les valeurs réelles  $\alpha$  et  $\beta$  sont fixées arbitrairement. En général  $\alpha > \beta$ .

#### III.4.4. Naïve bayes :

Cette méthode se base sur le théorème de Bayes permettant de calculer les probabilités conditionnelles. Dans le cas de la CT, la méthode **Naïve bayes** est utilisée comme suit : on cherche la classification qui maximise la probabilité d'observer les mots du document. Lors de la phase d'entraînement, le classificateur calcule les probabilités qu'un nouveau document appartient à telle catégorie à partir de la proportion des documents d'entraînement appartenant à cette catégorie. Il calcule aussi la probabilité qu'un mot donné soit présent dans un texte, sachant que ce texte appartient à telle catégorie. Quand un nouveau document doit être classé, on calcule les probabilités qu'il appartienne à chacune des catégories à l'aide de la règle de Bayes. [9]

La formule :

$$P(A/B) = \frac{P(A \cap B)}{p(B)}$$

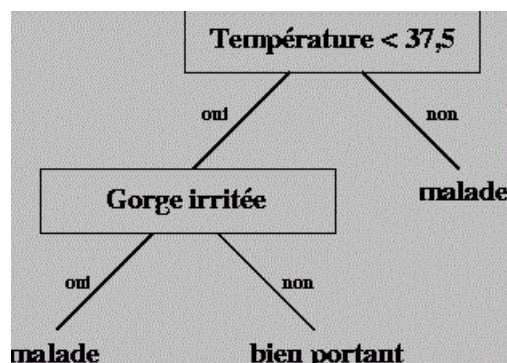
#### III.4.5. Les arbres de décision :

Les arbres de décision sont composés d'une structure hiérarchique en forme d'arbre. Un arbre de décision est un graphe orienté sans cycles, dont les nœuds portent une question, les arcs des réponses et les feuilles des conclusions ou des classes terminales.

Un classificateur de texte basé sur la méthode d'arbre de décision est un arbre de nœuds internes qui sont marqués par des termes, les branches qui sortent des nœuds sont des tests sur les termes et les feuilles sont marquées par catégories. [12]

Une méthode pour effectuer l'apprentissage d'un arbre de décision pour une catégorie  $C_i$  consiste à vérifier si tous les exemples d'apprentissage ont la même étiquette. Dans le cas contraire, nous sélectionnons un terme  $T_k$ , et nous partitionnons l'ensemble d'apprentissage en classes de documents qui ont la même valeur pour  $T_k$ , et à la fin on crée les sous arbres pour chacune de ces classes. Ce processus est répété récursivement sur les sous arbres jusqu'à ce que chaque feuille de l'arbre généré de cette façon contienne des exemples d'apprentissage attribués à la même catégorie  $C_i$ ,

qui est alors choisie comme l'étiquette de la feuille. L'étape la plus importante est le choix du terme de pour effectuer la partition.



**FigureI.8** : Exemple d'arbre de décision

#### III.4.6. Les réseaux de neurone :

Les réseaux de neurones artificiels sont habituellement utilisés pour des tâches de classification. Par analogie avec la biologie, ces unités sont appelées neurones formels. Un neurone formel est caractérisé par :

- ❖ Le type des entrées et des sorties.
- ❖ Une fonction d'entrée.
- ❖ Une fonction de sortie.

Le connexionnisme peut être défini comme le calcul distribué d'unités simples, regroupées en réseau. Un réseau de neurone est un ensemble d'éléments ou unités extrêmement simples (neurones) se comportant comme des fonctions de seuil, suivant une certaine architecture ;

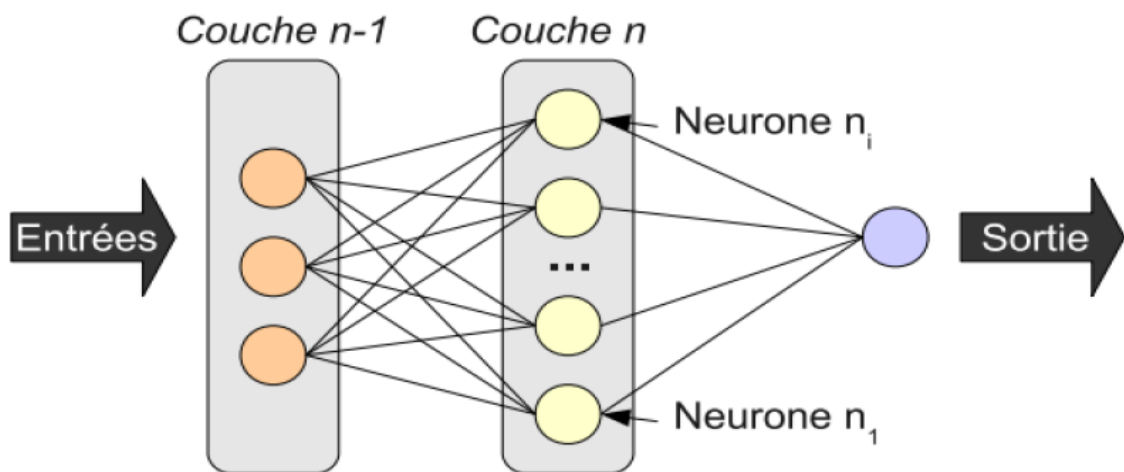
Chaque neurone prend en entrée une combinaison des signaux de sortie de plusieurs autres neurones, affectés de coefficients (les poids) ;

L'apprentissage s'effectue sous le contrôle des associations prédéfinies entre documents (entrées du réseau) et classes (sorties du réseau) qui fixent le comportement du réseau souhaité. La différence entre le comportement réel et désiré est une erreur qui sera à la base de l'apprentissage sous la forme d'une fonction de coût ou d'un signal d'erreur. Dans ce cas, l'apprentissage s'effectue en réajustant chaque fois les poids  $W_i$ .

Donc les algorithmes d'apprentissage permettent de calculer automatiquement les poids qui correspondent en réalité à des paramètres permettant de définir les frontières des classes.

Une structuration en couches effectue en cascade différents traitements sur un ensemble de données. Ces données sont présentées sur une couche terminale, appelée couche d'entrée ; elles sont ensuite traitées par un nombre variable de couches intermédiaires ou couches cachées. Le résultat est exposé sur l'autre couche terminale, la couche de sortie. [13]

Le principe général d'une approche neuronale est présenté ci-dessous. :



**Figure I.9 :** Architecture générale d'un réseau de neurones artificiels.

Un réseau de neurones artificiels est composé d'une ou de plusieurs couches se succédant dont chaque entrée est la sortie de la couche qui la précède comme illustré sur la **figure I.9**.

#### IV. Evaluation du processus de catégorisation :

Certains principes d'évaluation sont utilisés de manière courante dans le domaine de catégorisation de textes. Les performances en termes de classification sont généralement mesurées à partir de deux indicateurs traditionnellement utilisés à savoir

les mesures de rappel et précision. Initialement elles ont été conçues pour les systèmes de recherche d'information, mais par la suite la communauté de classification de textes les a adoptées. Formellement, pour chaque classe  $C_i$ , on calcule deux probabilités qui peuvent être estimées à partir de la matrice de contingence correspondante, ainsi ces deux mesures peuvent être définies de la manière suivante :

- ❖ Le rappel est la Proportion des solutions pertinentes qui sont trouvées. Mesure la capacité du système à donner toutes les solutions pertinentes.

$$R = \frac{V_p}{V_p + F_n}$$

- ❖ La précision est la Proportion de solutions trouvées qui sont pertinentes. Mesure la capacité du système à refuser les solutions non-pertinentes

$$P = \frac{V_p}{V_p + F_p}$$

$V_p$  : le nombre de documents correctement attribués à la catégorie.

$F_p$ : le nombre de documents incorrectement attribués à la catégorie.

$F_n$ : le nombre de documents qui auraient dû lui être attribués mais qui ne l'ont pas été.

## V. Les applications de la catégorisation des textes :

La catégorisation de textes peut être un support pour différentes applications parmi lesquelles :

- l'identification de la langue.
- la reconnaissance d'écrivains et la catégorisation de documents multimédia
- l'étiquetage de documents,
- le filtrage (consistant à déterminer si un document est pertinent ou non (décision binaire))
- le routage (consistant à affecter un document à une ou plusieurs catégories parmi  $n$ . [13])

## VI. Problèmes de la catégorisation de textes :

Plusieurs difficultés peuvent s'opposer au processus de catégorisation de textes [13], les principales sont les suivantes :

### a. **La redondance :**

La redondance et la synonymie permettent d'exprimer le même concept par des expressions différentes, plusieurs façons d'exprimer la même chose. Cette difficulté est liée à la nature des documents traités exprimés en langage naturel contrairement aux données numériques. LE FEVRE illustre cette difficulté dans l'exemple du chat et l'oiseau : *mon chat mange un oiseau, mon gros matou croque un piaf et mon félin préféré dévore une petite bête à plumes* (Lefèvre, 2000). La même idée est représentée de trois manières différentes, différents termes sont utilisés d'une expression à une autre mais en fin compte c'est bien le malheureux oiseau qui est dévoré par ce chat.

### b. **L'ambiguïté :**

A la différence des données numériques, les données textuelles sont sémantiquement riches, du fait qu'elles sont conçues et raisonnées par la pensée humaine. À cause de l'ambiguïté, les mots sont parfois de mauvais descripteurs ; par exemple le mot avocat peut désigner le fruit, le juriste, ou même au sens figuré, la personne qui défend une cause.

### c. **La graphie :**

Un terme peut comporter des fautes d'orthographe ou de frappe comme il peut s'écrire de plusieurs manières ou s'écrire avec une majuscule. Ce qui va peser sur la qualité des résultats. Parce que si un terme est orthographié de deux manières dans le même document (Ghelizane, Relizane), la simple recherche de ce terme avec une seule forme graphique néglige la présence du même terme sous d'autres graphies.

### d. **Complexité de l'algorithme d'apprentissage :**

Un texte est représenté généralement sous forme de vecteur contenant les nombres d'apparitions des termes dans ce texte. Or, le nombre de textes qu'on va traiter est très important sans oublier le nombre de termes composant le même texte donc on peut bien imaginer la dimension du tableau (textes \* termes) à traiter qui va compliquer considérablement la tâche de classification en diminuant la performance du système.



#### e. **Présence-Absence de termes :**

La présence d'un mot dans le texte indique un propos que l'auteur a voulu exprimer, on a donc une relation d'implication entre le mot et le concept associé, quoiqu'on sache très bien qu'il y a plusieurs façons d'exprimer les mêmes choses, dès lors l'absence d'un mot n'implique pas obligatoirement que le concept qui lui est associé est absent du document.

Cette réflexion pointue nous amène à être attentifs quant à l'utilisation des techniques d'apprentissage se basant sur l'exclusion d'un mot particulier.

#### f. **Les mots composés :**

La non prise en charge des mots composés comme : comme Arc-en-ciel, peut-être, sauve-qui-peut, etc.. Dont le nombre est très important dans toutes les langues, et traiter le mot Arc-en-ciel par exemple en étant 3 termes séparés réduit considérablement la performance d'un système de classification néanmoins l'utilisation de la technique des n-grammes pour le codage des textes atténue considérablement ce problème des mots composés.

## VII. Conclusion :

Dans ce chapitre nous avons présenté quelques techniques de la catégorisation automatique des textes ainsi que leurs avantages et leurs inconvénients. Nous avons également introduit les différents moyens d'évaluation d'un classificateur.

La catégorisation de texte a essentiellement progressé ces dix dernières années grâce à l'introduction des techniques héritées de l'apprentissage automatique qui ont amélioré très significativement les taux de bonne classification.

Le chapitre suivant présente les différentes mesures de similarité et ces objectifs et ces différentes approches qui peuvent être basés sur les arcs (Wu&palmer), les nœuds (Resnik, hirst&Ongé), ou hybride qui combine l'approche basé sur les arcs et l'approche basé sur les nœuds (jiang&contrah) ainsi que leur évaluation de plus nous interpréterons notre application.

# Chapitre II

## Les mesures de similarités sémantiques et l'amélioration du produit scalaire

## I. Introduction :

La majorité des méthodes utilisent les mesures de similarité statistiques pour la C.T. Ces mesures telles que le produit scalaire et le cosinus ont plusieurs inconvénients. Car si on prend l'exemple de deux mots de même sens (l'un se trouve dans les documents classés et l'autre dans le document non classé), alors ces mesures n'arrivent pas à classer ce document. Car elles ne trouvent pas le même mot dans les deux documents, et elles ne prennent pas en considération les sens des mots, de plus elles considèrent les mots comme étant indépendants alors que dans la langue naturelle les termes de la langue sont dépendants.

Pour notre travail on s'intéresse à l'utilisation des mesures de similarités sémantiques afin de pallier aux inconvénients des mesures de similarités statistiques.

Dans ce chapitre nous allons commencer par définir les mesures de similarités sémantiques ; par la suite nous allons décrire les étapes de notre application.

## II. Les mesures de similarités sémantiques :

### II.1. Définition :

La similarité dans un réseau sémantique peut être calculée en se basant sur les liens taxonomiques « is-a ». Plus généralement, le calcul de la similarité entre les concepts peut être basé sur les liens hiérarchiques de spécialisation/généralisation. Un moyen des plus évidents pour évaluer la similarité sémantique dans une taxonomie est alors de calculer la distance entre les concepts par le chemin le plus court. [14]

### II.2. Objectifs :

L'objectif des mesures de similarité sémantique est :

- Évaluer la proximité sémantique entre les concepts.

- Permet de déterminer s'ils sont similaires c'est-à-dire s'ils atteignent un certain niveau de ressemblance ou dissimilaire qui peuvent être également liés sémantiquement par des relations lexicales : antonymie, spécialisation, etc.

### II.3. Les Différentes approches de la similarité sémantique :

#### II.3.1. Approche basé sur les arcs :

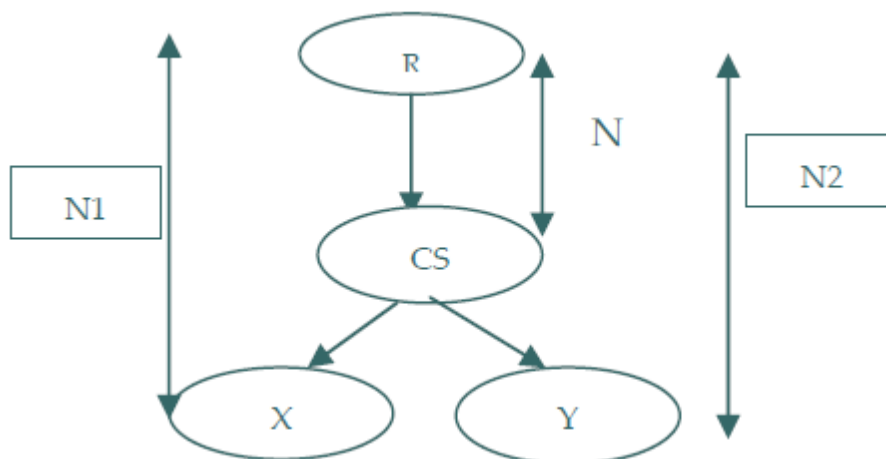
Ce type de mesure se sert de la structure hiérarchique de l'ontologie qui est représentée par un graphe dont les nœuds sont des concepts, et les arcs sont les liens entre ces concepts, et cela pour déterminer la similarité sémantique entre les concepts qui peut être calculée à partir du nombre de liens qui séparent les deux concepts. Parmi les mesures de ces approches on peut citer les suivantes :

##### II.3.1.1. Mesure de Wu & Palmer :

La mesure de similarité de [15] est basée sur le principe suivant :

Etant donnée une ontologie  $\Omega$  formée par un ensemble de nœuds et un nœud racine R (**Figure II.1**). Soit X et Y deux éléments de l'ontologie dont nous allons calculer la similarité. Le principe de calcul de similarité est basé sur les distances (N1 et N2) qui séparent les nœuds X et Y du nœud racine et la distance qui sépare le concept subsumant<sup>2</sup>(CS) de X et de Y du nœud R. sa formule est :

$$Sim_{wup}(x,y) = \frac{2 * N}{N1 + N2}$$



**Figure II.1** : Exemple d'un extrait d'ontologie.

[16] a effectué une comparaison entre les méthodes des mesures de similarité. Il en ressort que la mesure de [15] a l'avantage d'être simple à calculer en plus des performances qu'elle présente, tout en restant aussi expressive que les autres, La mesure de [15] est intéressante mais présente une limite car elle vise essentiellement à détecter la similarité entre deux concepts par rapport à leur distance de leur plus petit généralisant, ce qui ne permet pas de capter les mêmes similarités que la similarité conceptuelle symbolique. Cependant, avec cette mesure on peut obtenir une similarité plus élevée entre un concept et son voisinage par rapport à ce même concept et un concept fils, ce qui est inadéquat dans le cadre CT.

### II.3.2.Approche basé sur les nœuds (contenue informationnel) :

Cette approche prend en considération le contenu informatif (IC) des concepts de l'ontologie. La similarité est alors calculée à partir de l'information partagée par les concepts. Le contenu informatif est défini par :

$$IC(c) = -\log p(c).$$

Parmi les mesures basées sur le contenu informationnel on peut citer :

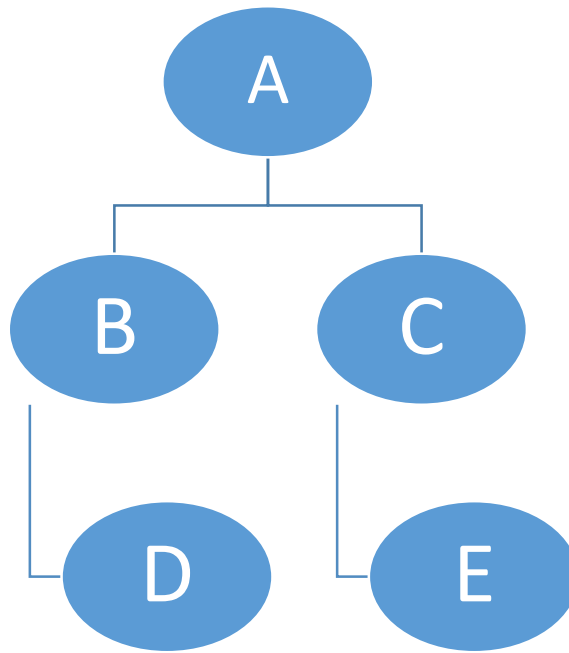
#### II.3.2.1 .Mesure de Resnik :

La notion du contenu informationnel (CI) a été initialement introduite par [17] qui a prouvé qu'un objet (mot) est défini par le nombre des classes spécifiées et que la similarité sémantique entre deux concepts est mesuré par la quantité de l'information qu'ils partagent. Pour évaluer la pertinence d'un objet il faut calculer le contenu informationnel. Le contenu informationnel est obtenu en calculant la fréquence de l'objet dans le corpus (Wordnet). La formule proposée par Resnik est définie par :

$$Sim (c1, c2) = CI (ppg (c1, c2))$$

Ou ppg=plus petit généralisant.

Cette mesure Offre de très bonne performance mais elle est un peu sommaire car elle dépend que du concept le plus spécifique.



**Figure II.2 :** Exemple montrant l'inconvénient majeur de la mesure de Resnik

Comme montré dans **Figure II.2** cette mesure considère la similarité entre C et E comme entre B et C.

#### II.3.2.2. Hirst & Onge

L'idée de cette mesure [18] est que deux concepts lexicalisés sont sémantiquement étroits si leurs ensembles synonymes (synsets) de WordNet sont reliés par un chemin qui n'est pas trop long et qui "ne change pas la direction trop souvent". Avec cette mesure, toutes les relations contenues dans un réseau Wordnet sont prises en considération.

Cette mesure est calculée comme suit :

$$\text{Sim}(c1,c2) = c - \text{len}(c1,c2) - k * T(c1,c2)$$

C et k sont des constant.

Len : le plus court chemin.

T : changement de direction.

#### II.3.3. Hybride :

Ces approches sont fondées sur un modèle qui combine entre les approches basées sur les arcs (distances) et les approches basées sur les nœuds en plus du contenu informationnel qui est considéré comme facteur de décision.

#### II.3.3.1. Jiang & Conrath :

Cette mesure [19] est basée sur la combinaison d'une source de connaissance riche (thesaurus) avec une source de connaissance pauvre (corpus). Notons que cette formule est définie par l'inverse de la distance sémantique.

$$Sim(X, Y) = \frac{1}{distance(X, Y)}$$

Sachant que la distance entre X et Y est calculée par la formule suivante :

$$distance(X, Y) = CI(X) + CI(Y) - (2 \cdot CI(LCS(X, Y)))$$

Où CI : contenue informationnelle

LCS : Ancêtre commun

Cette mesure sert à résoudre le problème de Resnik et elle est courante et efficace

#### II.3.3.2. Leacock et Chodorow :

Cette mesure [20] est basée sur la longueur du plus court chemin entre deux synsets de Wordnet. Les auteurs ont limité leur attention à des liens hiérarchiques «is-a» ainsi que la longueur du chemin par la profondeur globale P de la taxonomie. La formule est définie par :

$$Sim(x, y) = -\log\left(\frac{cd(x, y)}{2 * M}\right)$$

Où

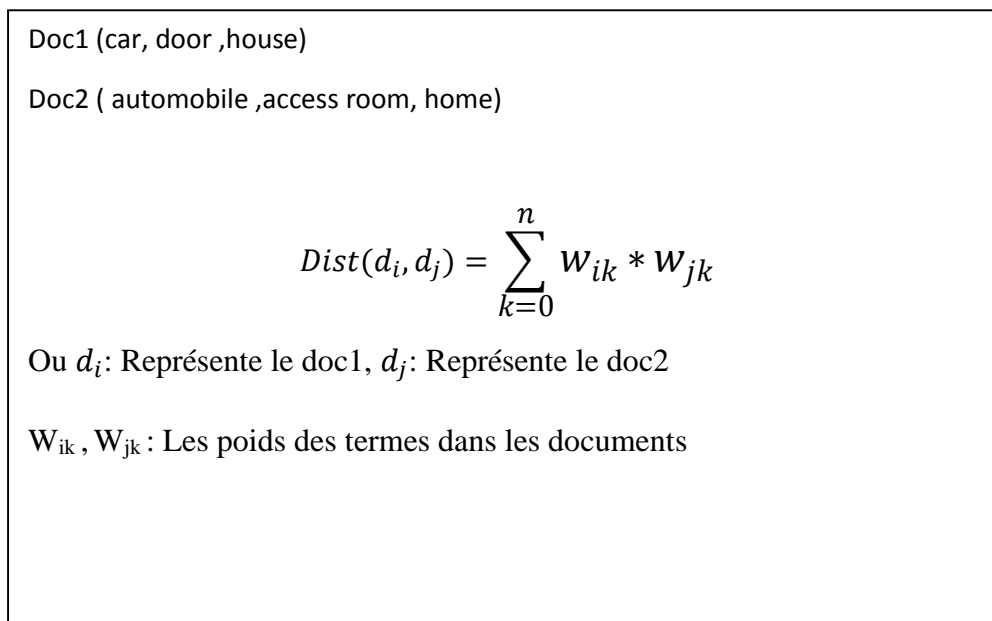
M : La longueur du chemin le plus long qui sépare le concept racine, de l'ontologie, du concept le plus en bas (les arcs).

cd (X, Y) : la longueur du chemin le plus court qui sépare X de Y (les nœuds).

Cette mesure n'est pas complète car elle ne prend en considération que les hyperonymes et les hyponymes.

### III. Architecture de notre travail :

Notre travail entre dans le cadre de la classification automatique des textes. Plus précisément notre travail consiste à classer les documents en utilisant les mesures de similarités sémantiques afin de pallier à l'inconvénient des mesures de similarité statistiques dont le produit scalaire est le plus célèbre.



**Figure II.3** : Exemple montrant l'inconvénient des MS statistiques

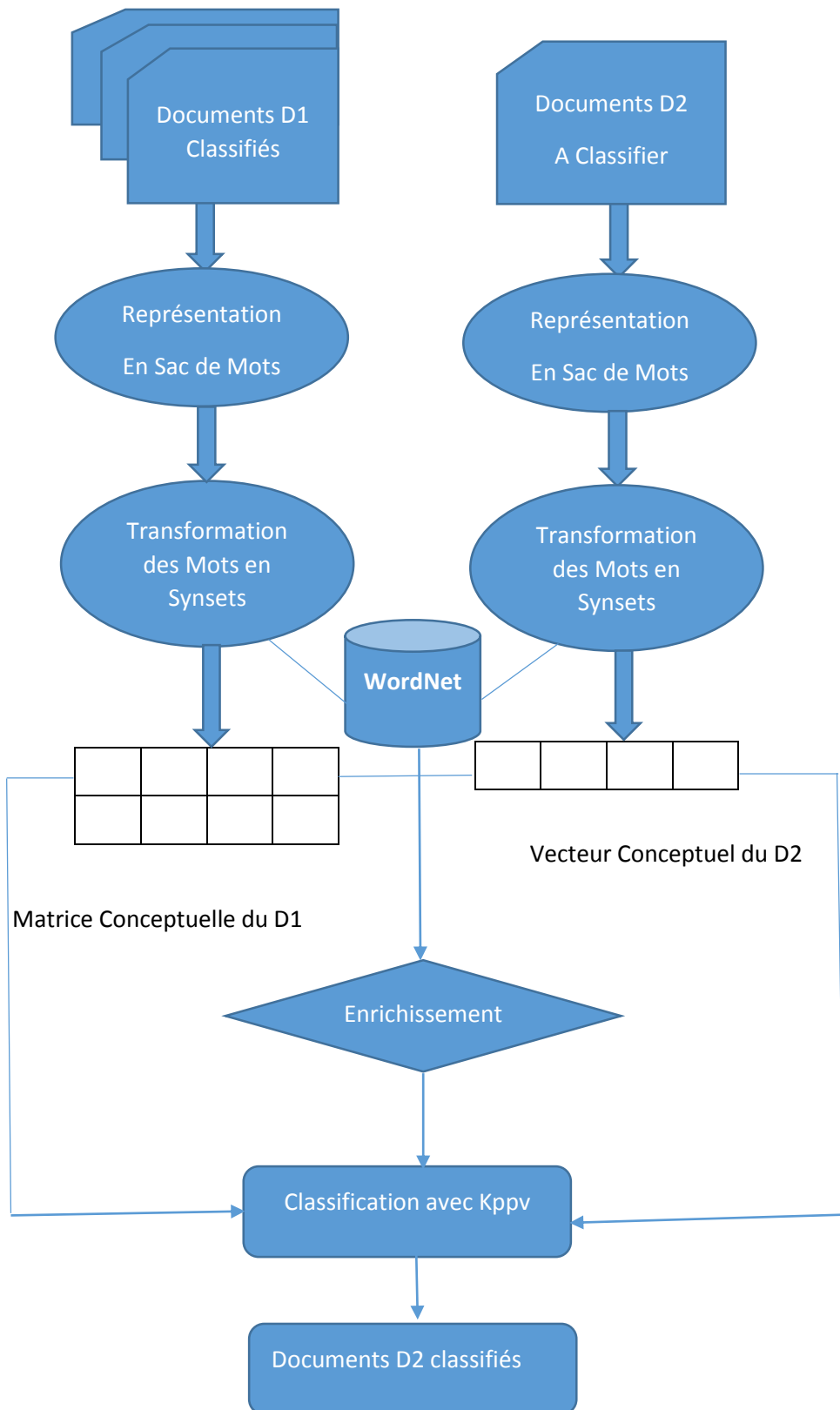
L'exemple de la **Figure II.3** montre l'inconvénient des MS statistiques, en effet dans cette figure on trouve deux documents qui sont proche l'un à l'autre, alors qu'ils ne partagent aucun mot commun, et de fait le PS aura la valeur 0.

Pour répondre à ce besoin on a essayé d'apporter une nouvelle mesure de similarité qui permet d'ajouter au produit scalaire une notion sémantique.

De ce fait nous avons implémenté une méthode de représentation qui est basée sur le WordNet pour traiter les documents classés et les documents non classés à fin de faire la représentation conceptuelle dans laquelle l'unité de vecteur serait un concept (groupe des synonymes). Ensuite nous passons à l'étape d'enrichissement dans le but est d'enrichir l'espace de représentation par des concepts qui n'existent pas dans les document, mais qui ont une relation avec ces derniers ; dans cette étape d'enrichissement il s'agit d'utiliser les relations sémantique entre les concepts (Hypernyms ,Hyponyms ,Meronyms ,Holonyms).

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 K-plus proches voisins (Kppv) pour associer une ou plusieurs catégories à un document non classé.





**Figure II.4 : Processus de représentation**

### III.1. Les étapes de représentation :

Dans notre projet les documents classés et les documents non classés passent par les étapes suivantes :

#### III.1.1. Représentation en sacs de mots :

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.

Le prétraitement consiste à rendre les documents sous une forme exploitable par la machine. Ces prétraitements sont les suivants :

GIRL?.girl !.house, 01 !! doctor ?? ..25 PEOPLE ..people ?? are 866 she

**Figure II.5** : Exemple d'un document

- **Tokenisation :**

Dans cette étape, il s'agit d'enlever toute la ponctuation. Voici la liste des ponctuations qu'on a utilisé : {+ -\* / ; : ( ) ! , ? < > 0 1 2 3 4 5 6 7 8 9 }

GIRL girl house doctor PEOPLE people are she

**Figure II.6** : Texte sans ponctuation

La **Figure II.6** présente le résultat de la tokenisation du document de la **Figure II.5**

- **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 sont le même sens donc on transforme les majuscules en minuscule.

Girl girl house doctor people people are she

**Figure II.7** : Document sans majuscules

La **Figure II.7** illustre le résultat sur le même exemple **Figure II.5**.

- **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 listes des mots vides contient les pronoms personnels, les prépositions, les articles....etc.

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, placed, 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 II.8** : Liste des mots vides

La **Figure II.8** représente la liste des mots vides qu'on a utilisée Si le mot fait partie de la liste alors le système va le supprimer.

Girl girl house doctor people people

**Figure II.9** : Elimination des mots vides

La **Figure II.9** illustre le résultat de la **Figure II.8** après l'élimination des mots vides.

- **Pondération des mots dans les documents**

Dans cette étape, il s'agit de construire la matrice d'occurrence (terme\*document) comme montré dans le **Tableau II.1** ci-dessous .où la ième ligne représente le ième document et la jème colonne représente le jème terme. L'intersection entre la ligne i et la colonne j représente la fréquence du jème terme dans le ième document.

Exemple : si les documents contiennent les textes suivants :

Document 1 (girlgirl house doctor people).

Document 2 (girl house house doctor house house doctor doctor people)

Document 3(house doctor people doctor)

	Girl	House	Doctor	People
Document 1	2	1	1	1
Document 2	1	4	3	1
Document 3	0	1	2	1

**Tableau II.1** : La fréquence des termes dans les documents.

### III.1.2.Transformation des mots en synsets :

#### III.1.2.1.Definition :

Une fois que la matrice d'occurrence est construite, nous passons à l'étape de la transformation des mots en synsets et cela grâce à la base lexicographique WordNet

dans laquelle les mots sont regroupés au sein de groupes de synonymes appelés synsets qui indique un sens différent du mot.

Etant donné qu'un terme peut avoir plusieurs sens, il est utile de sélectionner adéquat. Dans notre travail on a choisis de prendre en considération tous les sens possible du mot

Comme illustré dans la **Figure II.10**, la base lexicographique WordNet renvoie une liste ordonnée de synsets pour chaque mot qui peut être un nom, verbe, adverbe ou adjectif.

- **Noun :**  
**BICYCLE:** [Synset: [Offset: 2734941] [POS: noun] Words: bicycle, bike, wheel, cycle -- (a wheeled vehicle that has two wheels and is moved by foot pedals)].  
**HUMAN:**[Synset: [Offset: 6026] [POS: noun] Words:person, individual,someone, somebody, mortal, human, soul -- (a human being; "there was toomuch for one person to do")]
- **Verb:**  
**MAKE :** [Synset: [Offset: 2484888] [POS: verb] Words: make, do --(engage in; "make love, not war"; "make an effort"; "do research"; "do nothing"; "make revolution")]  
**WRITE:** [Synset: [Offset: 1649807] [POS: verb] Words: write, compose, pen, indite -- (produce a literary work; "She composed a poem"; "He wrote four novels")]
- **Adverb:**  
**ALWAYS:** [Synset: [Offset: 19245] [POS: adverb] Words: always, ever, e'er -- (at all times; all the time and on every occasion; "I will always be there to help you"; "always arrives on time"; "there is always some pollution in the air"; "ever hoping to strike it rich"; "ever busy")]
- **Adjectif:**  
**SMALL:** [Synset: [Offset: 1343705] [POS: adjective]Words: small, little --(limited or below average in number or quantity or magnitude or extent; "a little dining room"; "a little house"; "a small car"; "a little (or small) group"; "a small voice")]

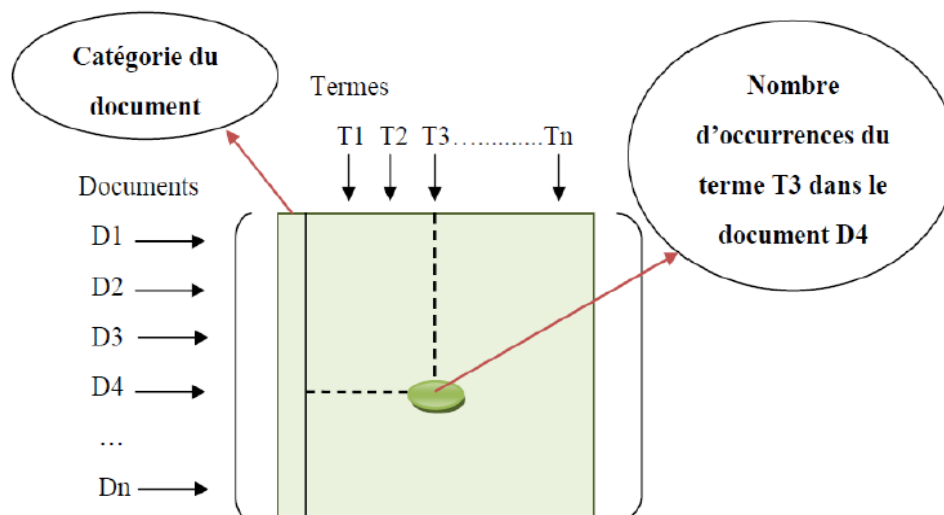
**Figure II.10 :** Exemple d'un groupe de synset.

*III.1.2.2.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.

La **Figure II.11** ci-dessous illustre la représentation matricielle d'un corpus où les lignes représentent les documents du corpus, les colonnes représentent les termes (les concepts), et l'intersection entre un document  $D_i$  et un terme  $T_j$  représente le nombre d'occurrences du terme  $T_j$  dans le document  $D_i$ .



**Figure II.11** : Représentation matricielle d'un corpus

### III.1.3.Enrichissement :

Dans le but d'enrichir l'espace de représentation ainsi que de prendre la relation entre les différents sens cette étape consiste à étendre l'ensemble des sens par l'intermédiaire des relations sémantiques entre les concepts ; dans notre travail on a pris en considération comme relations sémantiques :

- **Hypernymie :**

L'hyperonymie est la relation *sémantique* hiérarchique d'un lexème à un autre selon laquelle l'extension du premier terme, plus général, englobe l'extension du second, plus spécifique. Le premier terme est dit hyperonyme de l'autre, ou super ordonné par rapport à l'autre. C'est le contraire de l'hyponymie.[22]

- **Hyponymie :**

L'hyponymie est une Relation d'inclusion entre deux mots dont l'un est l'hyponyme de l'autre. La relation d'hyponymie est l'expression linguistique de la relation logique d'inclusion d'une classe dans une autre (La Linguistique, Paris, Denoël, 1969, p. 193).

On peut aussi définir les hyponymie comme la relation sémantique d'un lexème à un autre selon laquelle l'extension du premier est incluse dans l'extension du second. Le premier terme est dit hyponyme de l'autre. C'est le contraire de l'hyperonymie. [23]

- **Méronymie :**

La méronymie est une relation sémantique entre mots d'une même langue. Des termes liés par méronymie sont des méronymes. La méronymie est une relation partitive hiérarchisée : une relation de partie à tout. Un méronyme X d'un mot Y est un mot dont le signifié désigne une sous-partie du signifié de Y. La relation inverse est l'holonymie. WordNet inclus trois types de méronymie :

- X est un composante de Y.
- X est un élément de Y.
- X est le matériau dont Y est constitué. [24]

- **Holonymie :**

L'Holonymie est une relation sémantique entre mots d'une même langue. Des termes liés par holonymie sont des holonomes. L'holonymie est une relation partitive hiérarchisée : un holonyme A d'un mot B est un mot dont le signifié désigne un ensemble comprenant le signifié de B. La relation inverse est la méronymie. [25]

Rel. Sem→Hypernyms/Hyponyms/Meronyms/Holonyms

**-Exemples de relations d'hyperonymie et d'hyponymie**

Partant du sens le plus général du mot CAT (le félin), on obtient une liste ordonnée d'ancêtres et de descendants, permettant de déterminer qu'un chat est un carnivore, mammifère, un animal, ect.

**-Exemple de relations d'holonymie et meronymie**

Grâce à ces relations on peut déterminer qu'un chat a des pattes, un pelage, une queue...



---

**Les entrées :** S1 : l'ensemble des synsets des documents classés. S2 : l'ensemble des synsets des documents à classer,  $S \in (S1, S2)$

**Début**

P ← l'ensemble des sens ayant relation avec le sens S

Pour chaque sens  $P \in E$  faire

Si  $p \in (S1, S2)$  Maitre à jour le Tf

Sinon

Ajouter le sens P à l'ensemble (s1, s2)

Fin pour

**Fin**

---

#### III.1.4. Classification :

Dans cette étape il s'agit de classer les documents en utilisant les documents déjà classés. Il existe plusieurs méthodes de classification ; dans notre application on a choisis d'utiliser la méthode K-ppv qui est un algorithme qui permet de calculer la mesure de similarité entre les documents à classer et chaque document déjà classé, cet algorithme permet de sélectionner le K-plus proche document à un document, et de prendre comme classe la classe la plus représenté parmi les K-ppv ; tel qu'il est représenté dans l'algorithme suivant :

---

**Paramètre :** le nombre K de voisin

**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,

Classer le texte T dans la plus proches classe C.

Fin pour

**Fin.**

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

---

Dans l'algorithme de la méthode K-ppv il est nécessaire de déterminer les K-plus proche documents ; pour cela il est nécessaire d'utiliser une mesure de similarité ; notre mesure est sémantique qui permet de pallier au problème du produit scalaire. Le fonctionnement de notre mesure de similarité est résumé dans l'algorithme suivant :

---

**Les entrées** : vecteur du doc1 classé, vecteur du doc2 non classé, S1 synsets du doc1 classé, S2 synsets du doc2 non classé

**Les paramètres** : mesure de similarité sémantique.

**Début** :

Pour chaque sens du dictionnaire faire

Si le sens existe dans le doc2 alors

$$PS = \sum_{k=0}^n w_{ik} * w_{jk}$$

Sinon

Pour chaque sens du doc2 faire

Calculer la mesure de similarité sémantique (S1, S2)

*mesure* = *mesure* +  $w_{ik} * w_{jk} * \max \text{similarité} (S1, S2)$

Fin pour

Fin si


Fin pour

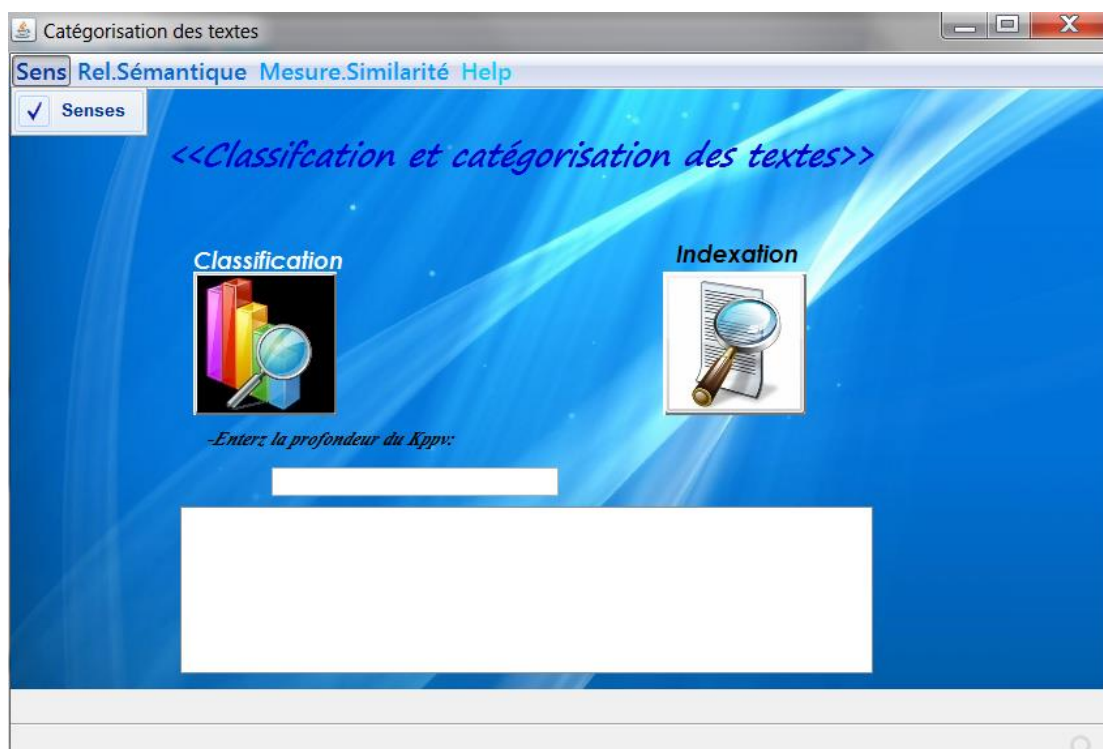
**Fin.**

---

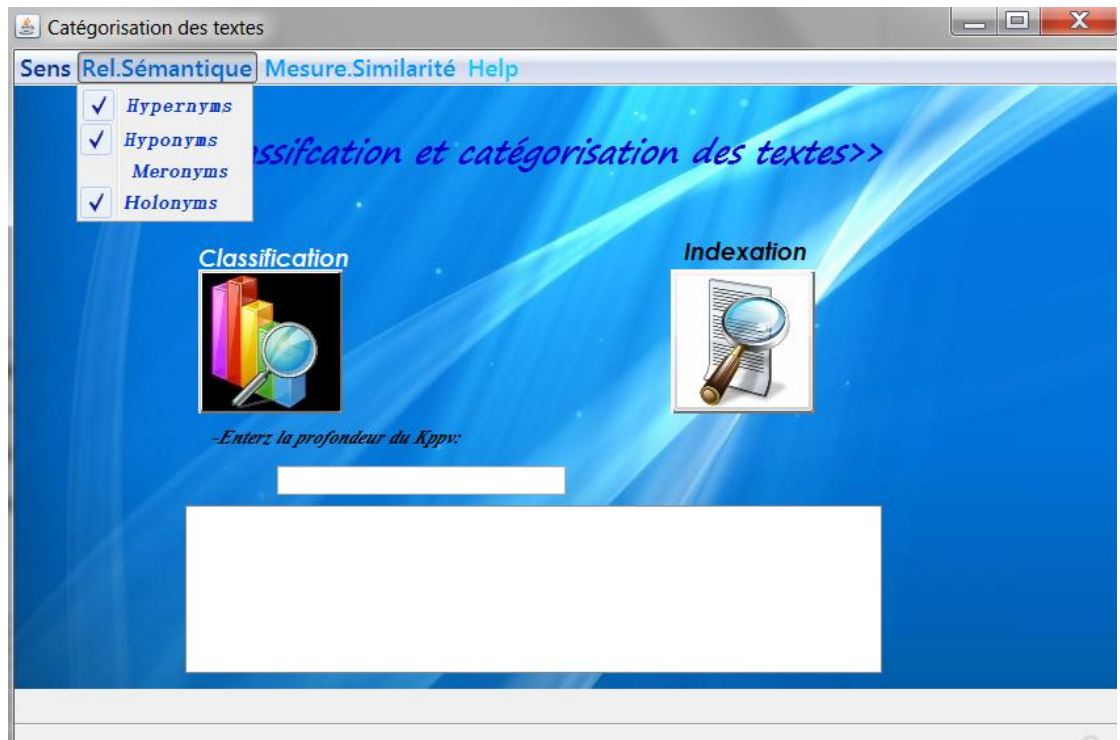
## IV. Exemple de déroulement de notre programme :

### IV.1. Indexation

Dans cette étape notre application nécessite de cliquer sur le bouton  pour l'indexation en donnant la possibilité de prendre le mot tel qu'il est dans le document ou de prendre tous les sens comme montré dans la **Figure II.12** et même de sélectionner le type des relations sémantiques voulu comme illustré dans la **Figure II.13**



**Figure II.12** : La sélection des sens



**Figure II.13** : La sélection des relations sémantiques

Une fois l'indexation est terminée le programme affiche la matrice (concept, document)

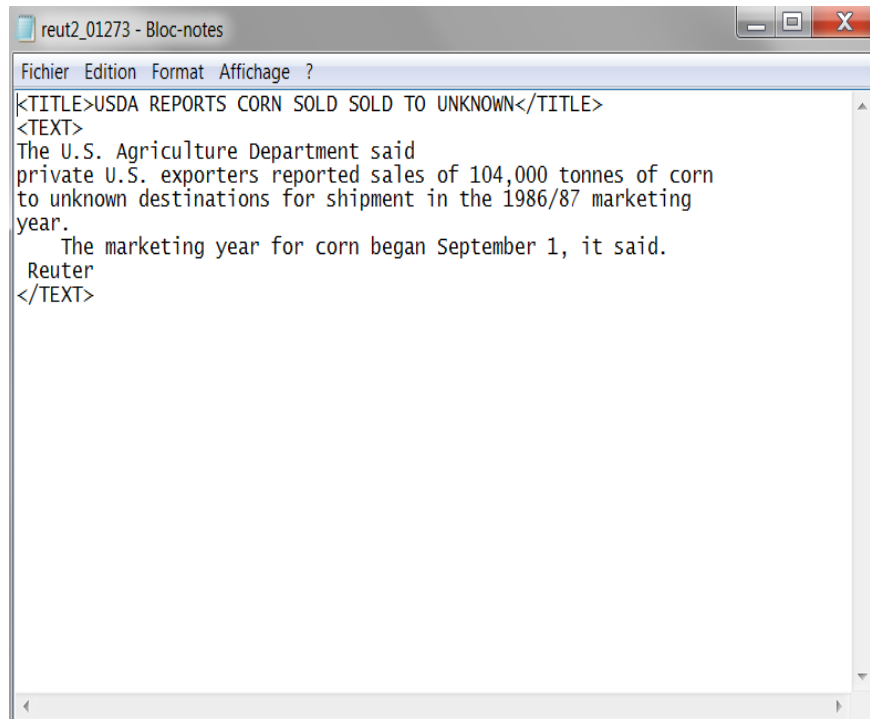
2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	1.0	2.0	1.0	1.0	1.0	1.0	1.0	2.0	2.0	2.0	2.0	2.0	1.	
2.0	2.0	2.0	2.0	2.0	2.0	2.0	5.0	2.0	2.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	2.0	2.0	2.0	1.0	0.
2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	2.0	2.0	2.0	0.0	0.
2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	2.0	2.0	2.0	0.0	0.
2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	2.0	2.0	2.0	1.0	0.
2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	2.0	2.0	2.0	0.0	0.

**Figure II.14** : matrice (concept, document)

Comme montré dans la **Figure II.14** le concept 5952891 se répète 2 fois dans le premier document et une fois dans le deuxième et il n'existe pas dans le 3ème document.


## IV.2. Classification

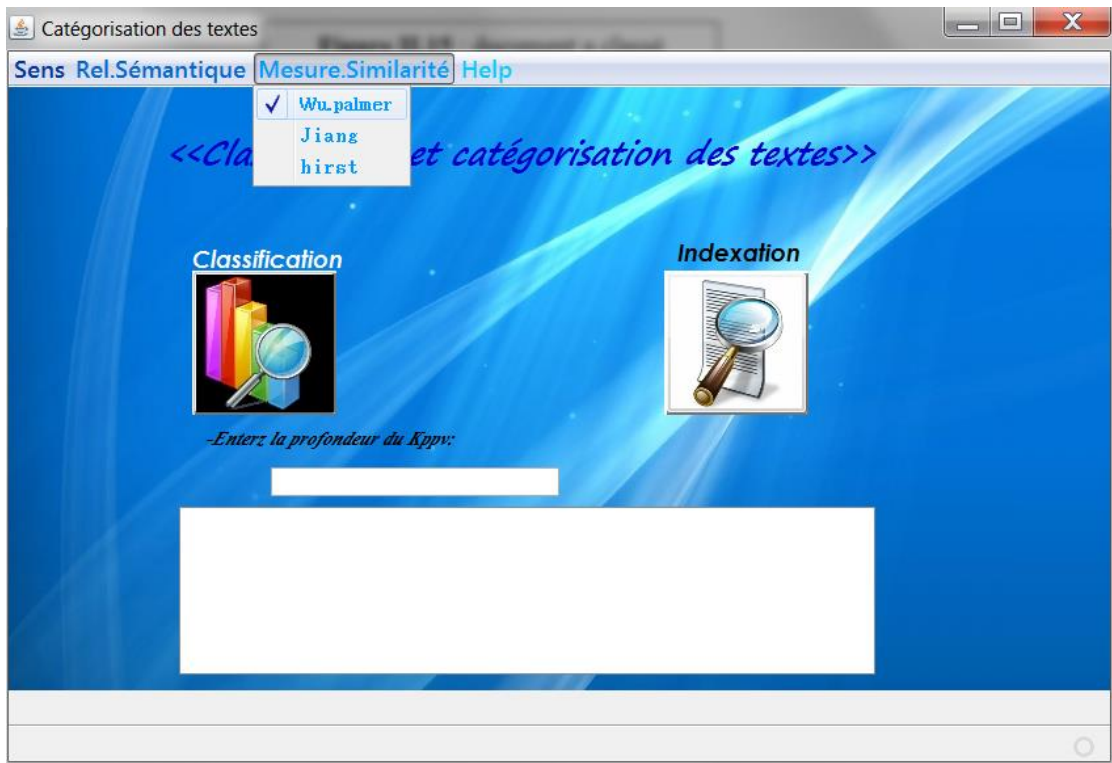
Dans cette étape nous allons prendre un exemple d'un document à classer.



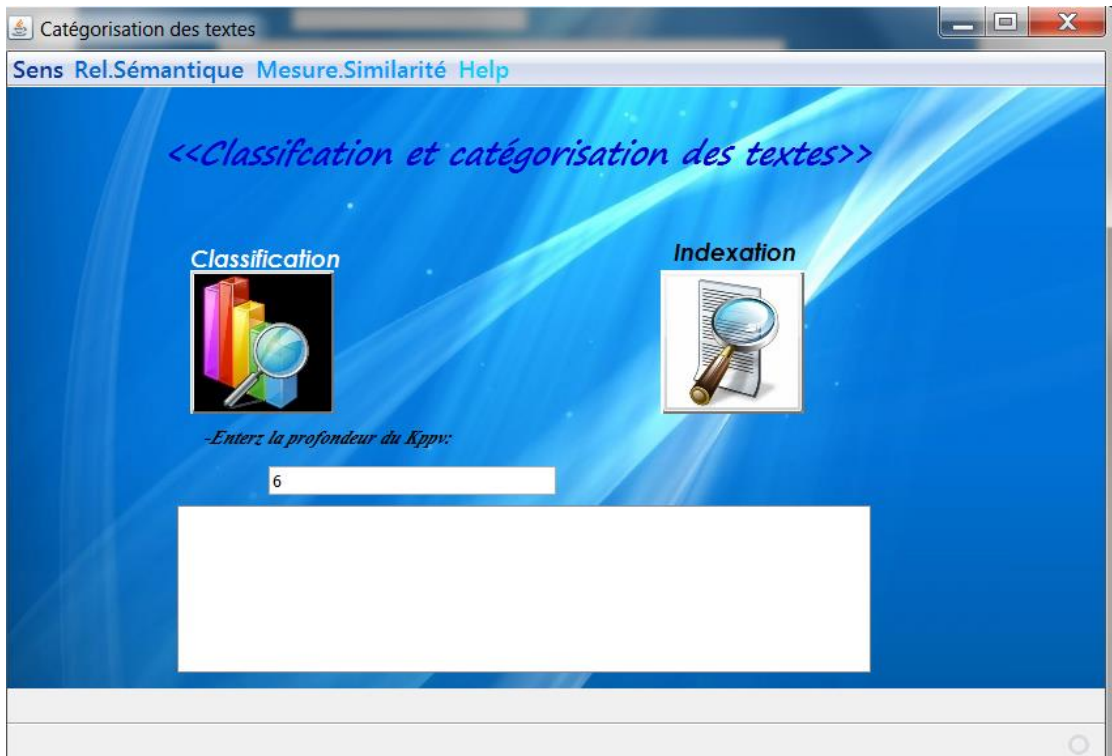
**Figure II.15** : document à classer



Cette étape nécessite de cliquer sur le bouton  pour classer le document de la **Figure II.15** en donnant la possibilité de choisir une des mesures de similarités sémantiques proposées sur notre interface et elle nous permet aussi de choisir la profondeur du Kppv.



**Figure II.16** : Sélection de la mesure de similarité sémantique



**Figure II.17** : La profondeur de Kppv

Voici le vecteur du document a classé comme illustré dans la **Figure II.18**

```
le vecteur du document à classé:
3.0 5.0 3.0 5.0 3.0 5.0 3.0 5.0 3.0 6.0 4.0 3.0 1.0 3.0 1.0 3.0 1.0 3.0 5.0 7.0 5.0 7.0 5.0 7.0 6.0 5.0 3.0 5.0 :
```

**Figure II.18** : Fréquence des termes dans le document a classé

```
le sens: 7637708n'exsiste pas dans la matrice des documents classés:
```

**Figure II.19** : Exemple montrant l'absence du sens dans les documents classés

Comme illustré dans la **Figure II.19** on voit clairement que le sens 7637708 du document non classé n'existe pas dans l'ensemble des sens des documents classés.

C'est pour cette raison qu'il faut lui calculer la mesure de similarité avec tous les sens des documents classés.

Les résultats interprétés dans la **Figure II.20** ont été calculé par la mesure de Wu&Palme

```
le sens: 7637708n'exsiste pas dans la matrice des documents classés:
Le sens du doc classé:7637708   Le sens du doc a classé:5952891   La similarité entre ces deux sens:0.2962962962962963
Le sens du doc classé:7637708   Le sens du doc a classé:5950505   La similarité entre ces deux sens:0.3333333333333333
Le sens du doc classé:7637708   Le sens du doc a classé:5953359   La similarité entre ces deux sens:0.2857142857142857
Le sens du doc classé:7637708   Le sens du doc a classé:13157003  La similarité entre ces deux sens:0.3333333333333333
Le sens du doc classé:7637708   Le sens du doc a classé:6141850   La similarité entre ces deux sens:0.32
Le sens du doc classé:7637708   Le sens du doc a classé:5948608   La similarité entre ces deux sens:0.32
Le sens du doc classé:7637708   Le sens du doc a classé:4887553   La similarité entre ces deux sens:0.2857142857142857
Le sens du doc classé:7637708   Le sens du doc a classé:5953118   La similarité entre ces deux sens:0.34782608695652173
Le sens du doc classé:7637708   Le sens du doc a classé:5950120   La similarité entre ces deux sens:0.32
Le sens du doc classé:7637708   Le sens du doc a classé:4882286   La similarité entre ces deux sens:0.2962962962962963
Le sens du doc classé:7637708   Le sens du doc a classé:6899621   La similarité entre ces deux sens:0.3333333333333333
Le sens du doc classé:7637708   Le sens du doc a classé:1051650   La similarité entre ces deux sens:0.2962962962962963
Le sens du doc classé:7637708   Le sens du doc a classé:12818586  La similarité entre ces deux sens:0.3636363636363636
Le sens du doc classé:7637708   Le sens du doc a classé:13029645  La similarité entre ces deux sens:0.3636363636363636
Le sens du doc classé:7637708   Le sens du doc a classé:7692741   La similarité entre ces deux sens:0.6956521739130435
Le sens du doc classé:7637708   Le sens du doc a classé:5528185   La similarité entre ces deux sens:0.3076923076923077
Le sens du doc classé:7637708   Le sens du doc a classé:8869095   La similarité entre ces deux sens:0.2
Le sens du doc classé:7637708   Le sens du doc a classé:2645     La similarité entre ces deux sens:0.2
```

**Figure II.20** : Calcul de la mesure de similarité

Dans le cas de l'existence du concept a classé dans les documents classés, notre programme calcule directement le produit scalaire entre ces deux derniers comme illustré dans la **Figure II.21**

```
la somme de la distance de similarité:22.0
le sens13157003 existe dans la matrice des doc classés:
le produit scalaire :32.0
la somme de la distance de similarité:32.0
le sens6141850 existe dans la matrice des doc classés:
le produit scalaire :38.0
la somme de la distance de similarité:38.0
le sens5948608 existe dans la matrice des doc classés:
le produit scalaire :48.0
la somme de la distance de similarité:48.0
le sens4887553 existe dans la matrice des doc classés:
le produit scalaire :54.0
la somme de la distance de similarité:54.0
le sens5953118 existe dans la matrice des doc classés:
le produit scalaire :64.0
la somme de la distance de similarité:64.0
le sens5950120 existe dans la matrice des doc classés:
le produit scalaire :70.0
la somme de la distance de similarité:70.0
le sens4882286 existe dans la matrice des doc classés:
le produit scalaire :82.0
la somme de la distance de similarité:82.0
```

**Figure II.21** : calcul du produit scalaire

Une fois la mesure de similarité et le produit scalaire calculés, le programme nous affiche le K- plus proche document et à quelle classe il appartient comme montré dans la **Figure II.22**.

```
bravo, votre document est bien classé
c0est la classe du document a classé
c0est la plus proche classe du document
- - -
```

**Figure II.22** : le classement du document

## V. Environnement et outils de développement :

Les outils et les langages utilisés pour la manipulation des données ainsi que l'implémentation sont décrits comme suit :

### V.1. Language JAVA :

C'est un langage de programmation orienté objet simple ; ce qui réduit les risques d'incohérence, développé par Sun Microsystems, inspiré de C++.il permet de



créer des logiciels compatible avec de nombreux systèmes d'exploitations (Windows, Linux, Macintosh, Solaris).

JAVA donne la possibilité de développer des programmes pour téléphone portable et assistants personnels. Il est caractérisé aussi par la réutilisation de son code ainsi que la simplicité de sa mise en œuvre. Ce langage peut être utilisé sur internet pour des petites applications intégrées à la page Web ou encore comme langage serveur (jsp). Il possède une riche bibliothèque de classes.

#### V.2. Environnement de développement :

L'environnement de développement utilisé, est le NetBeans 6.8, il possède de nombreux avantages qui sont les suivants :

- un environnement de développement intégré (EDI)
- en plus de JAVA, NetBeans permet également 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, ...

#### V.3. WordNet :

Afin d'implémenter notre travail, nous avons utilisé WordNet de version **2.1** 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 tableau ci-dessous montre la structure de WordNet d'anglais en nombre de mots, nombre de synsets et nombre de sens, globalement et par catégorie grammaticale :

Position	Mots	Synsets	Total paires Mots-Sens
Nom	117097	81426	145104
Verbe	11488	13650	24890
Adjectif	22141	18877	31302
Adverbe	4601	3644	5720
Total	155327	177597	207016

**Tableau II.2** : Caractéristiques du nombre de mots et de concepts dans WordNet

#### V.4.JWNL :

JWNL(Java WordNet Library) est une API Java pour avoir accès au dictionnaire relationnel WordNet dans des formats multiples, aussi bien que la découverte des relations hiérarchiques et de traitement morphologique. Elle est compatible avec des versions WordNet 2.0 à 3.0 et est une mise en œuvre Java complète. L'API courant est JWNL 1.3. JWNL 1.4 est dans le développement.

#### V.I.Conclusion :

Dans ce chapitre nous avons présenté la description et la mise en œuvre des étapes implémentées pour notre approche, qui avaient pour intérêt d'utiliser les mesures de similarité sémantiques pour la catégorisation et la classification des textes en utilisant une représentation conceptuelle basée sur une base lexicographique WordNet ainsi qu'un corpus Reuters, ensuite nous avons utilisé la méthode Kppv pour attribuer à chaque document sa catégorie.

# Conclusion générale

# Conclusion générale

Notre travail présenté dans ce mémoire s'inscrit dans le cadre de la représentation conceptuelle pour la catégorisation des textes. Sans oublier que le but de la catégorisation est d'apprendre à une machine à classer un texte dans la bonne catégorie en se basant sur son contenu.

Notre mémoire se décompose en deux chapitres. Le premier chapitre vise à présenter le processus de la catégorisation des textes et les principales phases de ce dernier, ainsi, les applications liées à la catégorisation des textes, le deuxième chapitre présente un état d'art sur les mesures de similarité sémantique et leurs approches, aussi une exposition de la description des approches implémentées ainsi que les résultats obtenus.

La représentation conceptuelle dans laquelle l'unité de vecteur serait un concept (groupes des synonymes appelé synsets), nous a permis de voir comment l'intégration d'une base lexicale Wordnet a permis l'amélioration de la performance de notre classificateur. Les éléments de cette représentation ne sont plus associés directement à des simples mots mais plutôt à des concepts.

Malheureusement, le temps est court et il a été nécessaire d'ajouter d'autres mesures de similarité sémantique, de fixer certains paramètres pour en étudier d'autres plus en profondeur ainsi que plusieurs seuils. Evidemment, il aurait été intéressant d'observer le comportement de nos approches implémentées sur d'autres corpus plus riches, ainsi que sur d'autres classificateurs. Notre perspective dans un premier temps est de consolider la démarche implémentée en évaluant sur d'autres collections, puis élargir notre domaine en ajoutant d'autres mesures de similarité sémantique et aussi de travailler avec la dernière version de WordNet 3.1.

# Listes des figures

## **Chapitre I** : Classification des textes

Figure I.1 : Processus de la catégorisation des textes

Figure I.2 : Exemple de N-grammes de mots et de caractères

Figure I.3 : La représentation conceptuelle du mot « pic »

Figure I.4 : Sélection des attributs

Figure I.5 : Extraction des attributs

Figure I.6 : La séparation de l'hyper plan par les SVM

Figure I.7 : Les vecteurs de support

Figure I.8 : Exemple d'arbre de décision

Figure I.9 : Architecture générale d'un réseau de neurone artificielle

## **Chapitre II** : Notre approche

Figure II.1 : Exemple d'extrait d'ontologie

Figure II.2 : Exemple montrant l'inconvénient majeur de la mesure Resnik

Figure II.3 Exemple montrant l'inconvénient des MS statistiques

Figure II.4 : Processus de représentation

Figure II.5 : Exemple d'un document

Figure II.6 : Texte sans ponctuation

Figure II.7 : Document sans majuscules

Figure II.8 : Liste des mots vides

Figure II.9 : Elimination des mots vides

# Listes des figures

Figure II.10 : Exemple d'un groupe de synset.

Figure II.11 : Représentation matricielle d'un corpus

Figure II.12 : La sélection des sens

Figure II.13 : La sélection des relations sémantiques

Figure II.14 : matrice (concept, document)

Figure II.15 : document a classé

Figure II.16 : Sélection de la mesure de similarité sémantique

Figure II.17 : La profondeur de Kppv

Figure II.18 : Fréquence des termes dans le document a classé

Figure II.19 : Exemple montrant l'absence du sens dans les documents classés

Figure II.20 : Calcule de la mesure de similarité

Figure II.21 : calcule du produit scalaire

Figure II.22 : le classement du document

# Listes des tableaux

Tableau II.1 : Fréquence des mots dans les documents

Tableau II.2 : Caractéristiques du nombre de mots et de concepts dans WordNet

# Listes des abréviations

CT : Catégorisation des textes.

TF: Term Frequency.

TFidf: Term Frequency Inverse Document Frequency.

TFC: Term Frequency Collection.

SVM : Machine a Vecteur Support.

K-ppv : le K plus proche voisin.

IC : Contenne Informationnelle.

PS : Produit Scalaire.

MS : Mesure de similarité.



# Listes des références

[1] : Radwan JALAM, « Apprentissage automatique et catégorisation de textes multilingues », Thèse de doctorat, Université Lumière Lyon 2, France, Juin 2003

[2] : FABRIZIO SEBASTIANI, «Machine Learning in Automated Text Catégorisation», Conseil recherché National, Italie, Mars 2002

[3] : Salton 1968, Université de CORNELL, pionnier de l'information retrieval modèle vectorielle

[4] : M. F. Porter, «An algorithm for suffix stripping », Program, pp 130–137, Morgan Kaufmann Publishers Inc, 1980.

[5] : Camelia Ignat, « Représentation de textes a l'aide d'étiquettes sémantiques dans le cadre de la classification automatique », European Commission, IPSC, Strasbourg, France, 2007.

[6] : Jalam, R. and Chauchat, J.-H. « *Pourquoi les n-grammes permettent de classer des textes? Recherche de mots-clefs pertinents à l'aide des n-grammes caractéristiques* ». In Morin, A. and Sébillot, P., editors, 6èmes Journées internationales d'Analyse statistique des Données Textuelles, St. Malo France. 2002

[7] : Caropreso Maria Fernanda, Stan Matwin , Fabrizio Sebastiani, 2000 « *Statistical Phrases in Automated Text Categorization* » Department of Computer Science of the University of URL :

<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.5902>

[8] : Furnkranz Johannes, Tom Mitchell, Ellen Riló 1998, « *A Case Study in Using Linguistic Phrases for Text Categorization on the WWW* » School of Computer

# Listes des références

Science Carnegie Mellon University URL :[www.cs.utah.edu/~riloff/pdfs/final-webslog-paper.pdf](http://www.cs.utah.edu/~riloff/pdfs/final-webslog-paper.pdf)

[9] : Simon RÉHEL, « Catégorisation automatique de textes et Cooccurrence de mots provenant de documents non étiquetés », Mémoire, Université Laval Québec, Canada, Janvier 2005

[10] : Simon JAILLET, Maguelonne TEISSEIRE, Jacques CHAUCHE, Violaine PRINCE, « Classification automatique de documents, Le coefficient des deux écarts », Université Montpellier2, France, 2005

[11] : Terkia Amel La Représentation Conceptuelle pour la Catégorisation des Textes Multilingue .Mémoire de magistère. Algérie 2011-2012.

[12] : Karima ABIDI, « La catégorisation de texte Multilingue », Mémoire de magistère, Ecole supérieur d'Informatique, Algérie, 2010-2011.

[13] : Mataalah Hocine « classification automatique de textes Orienté Agent » faculté des sciences –algerie2010-2011

[14] : R. Rada, H. Mili, E. Bichnell et M. Blettner, Development and application of a metric on semantic nets. IEEE Transaction on Systems, Man, and Cybernetics: pp 17-30. 1989.

[15] : Z. Wu et M. Palmer. Verb semantics and lexical selection. In Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics, pp 133-138. 1994.

# Listes des références

[16]: D. Lin. An Information-Theoretic Definition of similarity. In Proceedings of the Fifteenth International Conference on Machine Learning (ICML'98). Morgan-Kaufmann: Madison, WI, 1998.

[17]: P. Resnik. Using information content to evaluate semantic similarity in taxonomy. In Proceedings of 14<sup>th</sup> International Joint Conference on Artificial Intelligence, Montreal, 1995.

[18]: G.Hirst et D.St Onge. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum (editor), WordNet: An electronic lexical database, Cambridge, MA:The MIT Press .1998.

[19]: J. Jiang et D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings of International Conference on Research in Computational Linguistics, Taiwan, 1997.

[20]: C. Leacock et M. Chodorow. Combining Local Context and WordNet Similarity for Word Sense Identification. In WordNet: An Electronic Lexical Database, C. Fellbaum, MIT Press, 1998.

[21]: <http://theses.ulaval.ca/archimede/fichiers/22376/ch02.html> 21/04/2015

[22]: <https://fr.wikipedia.org/wiki/Hyponymie> 20/09/2015

[23]: <https://fr.wikipedia.org/wiki/hyponymie> 20/09/2015

[24]: <https://fr.wikipedia.org/wiki/méronymie> 20/09/2015

[25]: <https://fr.wikipedia.org/wiki/Holonymie> 20/09/2015

## Résumé

Dans le but de présenter notre projet de master nous avons proposé une mesure de similarité sémantique dans le cadre de la catégorisation des textes afin de trouver à quelle classe appartient le document et le classer, à l'aide de la base de données lexicales WordNet.

L'implémentation et la conception sont faites à l'aide du langage Java en utilisant l'IDE NetBeans6.8.

Mots-clés : JAVA ,WordNet

## Abstract

In order to present our master project we proposed a semantic similarity measure through the categorization of texts to find at which class the document belongs and classify it, with the help of the WordNet lexical database.

The implementation and design are made using the Java language using the IDE NetBeans6.8.

Keywords: JAVA, WordNet

## ملخص

من أجل تقديم مشروع ماجستير اقترحنا مقياس التشابه الدلالي من خلال تصنيف النصوص للعثور على الوثيقة WordNet التي ينتمي إليها الملف و القيام بتصنيفه بمساعدة قاعدة بيانات معجمية

تم التصميم و التنفيذ عن طريق لغة java وباستخدام netbeans.

كلمات البحث: JAVA ،WordNet، mesure de similarité، classification، le K-ppv