.

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

Catégorisation automatique des textes avec des mesures de similarité sémantiques

Réalisé par :

- ***M$_{elle}$ BENACHOUR Ikram.***
- ***M$_{elle}$ CHIKHAOUI Hadjer.***

Présenté le 03 Juillet 2019 devant le jury composé de :

- ***Mr MERZOUG.M*** *(Président)*
- ***Mr BENTAALLAH Mohammed Amin*** *(Encadreur)*
- ***Mr HADJILA.F*** *(Examinateur)*

Année universitaire: 2018-2019

Remerciements

Avant tout nous remercions le bon DIEU de nous avoir aidés à accomplir ce modeste projet et nous à données la patience et le courage durant cette année.

Nous remercions très sincèrement notre encadreur Monsieur A.BENTAALLAH qui nous a permis de bénéficier de son encadrement. Les conseils qu'il nous a prodigué, ses remarques pertinentes, la patience et la confiance qu'il nous a témoignés ont été déterminants dans la réalisation de notre travail de recherche.

Nos 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.

Nos remerciements s'étendent également à tous nos enseignants durant les années des études.

Merci à Tous et à Toutes.

Dédicace

Je dédie ce travail :

A mes très chers parents qui ont toujours répondu présents dans les moments les plus difficiles pour leurs irremplaçables et inconditionnels soutiens, leurs confiances et leurs sacrifices qui ont contribués à ma réussite.

A mon cher frère Lotfi ;

A mes chères sœurs Manel et Lamia ;

A mes cousines warda et Nawel ;

A mes Copines Soumia, Imen, nadia, khaiyra khawla, fatima et warda ;

Pour la gentillesse, la générosité, la joie de vivre, la patience et la volonté dont vous m'avez toujours entourée et que vous m'avez transmise.

A mon binôme Mlle Chikhaoui Hadjer et toute sa famille

Et ainsi qu'à tous mes amis.

B.Ikram

Dédicace

Je dédie ce travail :

A mes très chers parents, je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours. Bien que je ne vous en acquitterai jamais assez. Puisse Dieu, le Très Haut, vous accorde santé, bonheur et une longue vie.

A mes grands-mères qu'Allah vous protège.

A mes très chers et adorable frères Sid Ahmed et Zouhir.

A mes cousines et cousins et toute ma famille.

A ma chère amie Fatima Zohra .

Ames chères voisines Imene , Asma .

A mon binôme Mlle BENACHOUR Ikram et toute sa famille

Et ainsi qu'à tous mes amis.

CH.Hadjer

Table de matières

Table de matière……1
Liste des figures ……...3
Liste des tableaux ……4
Introduction générale 5

Chapitre I : Classification et Catégorisation du texte

l.1 Introduction :..... 7
I.2 Historique : 7
l.3 Définition de la Catégorisation de texte (C.T) : 7
I.4 Processus de la catégorisation de textes : 8
1.4.1 La représentation de textes : 9

- Représentation en « sac de mots » : 9
- Représentation par lemmes ou racines lexicales : 9
- Représentation des textes par des phrases :..... 10
- Représentation N_gramme : 10
- Représentation par concept : 11

I.4.2 Pondération : 11
I.4.3 Réduction de dimensionnalité : 12
I.4.4 Choix de classifieurs :..... 14

- La méthode de Rocchio..... 14
- La méthode SVM((Support Vector Machin)…….....14
- Les arbres de décision : 14
- Les k plus proches voisins (k-PPV) :..... 15

I.4.5 Evaluation de la qualité des classifieurs :..... 15

- Précision 15
- Rappel..... 16

I.5 Les applications de la catégorisation des textes : 17
I.6 Les problèmes de la C.T 17
I.7 conclusion ……19

Chapitre II : les mesures de similarités sémantiques

II.1 Introduction : 21
II.2 C'est quoi une mesure de similarité ? 21
II.3 Les mesures de similarités statistiques : 21
II.4 Les mesures de similarité sémantiques: 23
II.4.1 Approche basée sur les arcs : 24
II.4.2 Approche basé sur les nœuds : 25
II.4.3 Approches hybrides : 27
II.5 Conclusion : 28

Chapitre III : implémentation de l'application et l'évaluation des résultats

III.1Introduction : 30
III.2 Etapes de notre système : 30
III.2.1 Représentation de documents : 32
- Tockénisation (analyse lexicale) : 32
- Elimination des mots vides 33
- Pondération : 34
- Maping des termes en concept(Synset) : 35

III.2.2 Classification : 36
- III.2.1 classification via des mesures de similarité statistiques 40
- III.2.1 classification via des mesures de similarité sémantiques40
- III.3.Description des outils et les technologies utilisées : 44

III.4. conclusion ……..……46
. conclusion générale ……..……..……..……47
Références bibliographiques……..……48

Liste des figures

Chapitre I :

Figure I.1: Les étapes de catégorisation de texte. ……8
Figure I.2 : Exemple de N-grammes.……..10
Figure I.3 : Principe de sélection…….13
Figure I.4: Principe d'extraction……..13
Figure I.5 : Exemples de séparateurs ……..14
Figure I.6 : Exemple d'arbre de décision……...….15
Figure I.7 : Principe de Précision et Rappel ……..16
Figure 1.8 : Exemple de système de classification d'emails……..17

Chapitre II :

Figure II.1 : Les approches de mesure de similarité sémantique…….23
Figure II.2 : Exemple d'un extrait d'ontologie..……..24

Chapitre III :

Figure III.1 : Etapes de notre système…….....31
Figure III.2 : Exemple de corpus…….....32
Figure III.3 : La liste de séparateur……...……...32
Figure III.4 : La liste des mots vides……..……..……...33
Figure III.5 : Exemple d'un groupe de synset…….....35
La figure III.6 : Représentation conceptuelle……...……...36
Figure III.7 : Représentation conceptuelle de l'exemple III..2…...……..……..36
Figure III.8 : Algorithme de Knn……...…….37
Figure III.9 : Exemple montre l'inconvenant de produit scalaire...…...…….38
Figure III.10 : Algorithme de produit scalaire enrichi…….39
Figure III.11 le document non classé……..40
Figure III.12 : le résultat de classification statistique ……...40
Figure III.13 **:** exemple de calcule la similarité avec la mesure de wu palmer……...41
Figure III.14 les valeurs de produit scalaire enrichi totale avec la méthodes tous sens..42
Figure III.15 les valeurs de produit scalaire enrichi totale avec la méthode un seul sens…….42
Figure III.16 application d'algorithme knn sur l'exemple III.1 avec tous sens…….....43
Figure III.17 application d'algorithme knn sur l'exemple III.1 avec un seul sens..…..43

Liste des tableaux

Tableau I.1 : les méthodes de pondération…….12

Tableau III.1 : le résultat d'appliqué la Tockénisation sur l'exemple II.....32

Tableau III.2 : les résultats d'élimination des mots vides sur l'exemple …….34

Tableau III.3 : La matrice documents*termes résultante de l'exemple III.2…….…...34

Tableau III.4 : représentation du document D' non classé …….…....40

Tableau III.5 : calcule des mesures de similarité.....…….41

Tableau III.6 : Caractéristiques du nombre de mots et de concepts dans WordNet…...46

Introduction générale

Aujourd'hui, nous vivons dans un monde où l'information est disponible en grande quantité, chaque jour les êtres humains publient sur internet des informations et des documents et créent des sites web, les statistiques montrent une augmentation énorme de la quantité d'information disponible. Un meilleur exemple de cette augmentation est celui de la base documentaire de Google qui est passe de 25 millions de pages durant ces premiers jours en novembre 1998 pour atteindre environ 30 trillions de pages en aout 2012 (voir http ://www.google.fr). Face à cette augmentation, il est devenu quasi impossible de traiter manuellement cette gigantesque base documentaire. Pour cela, il est indispensable de concevoir des systèmes permettant de chercher, classer, conserver, mettre à jour et analyser et accéder rapidement à l'information désirée et minimisant l'intervention humaine. Ces outils proviennent de deux domaines à savoir la recherche d'information et la catégorisation de textes.
La catégorisation automatique des textes consiste à associer une catégorie à un document pouvant être une phrase, un paragraphe, un texte, etc.

Cette étude conduite à améliorer les performances des systèmes de catégorisation automatique des documents via une catégorisation sémantique qui se base sur l'utilisation des mesures de similarité sémantique, pour éviter les soucies de catégorisation statistique.

Le mémoire est structuré en trois chapitres. Le premier chapitre vise à focaliser le processus de la catégorisation des textes et les principales phases de ce dernier, ainsi, les applications et les problèmes liées à la catégorisation des textes , le deuxième chapitre présente un état d'art sur les mesures de similarité sémantiques et leurs approches. Finalement, le dernier chapitre expose la description de l'approche implémentée ainsi que les outils et ressources utilisés.

Chapitre I : classification et catégorisation des textes

l.1 Introduction :

De nos jours, les besoins de catégorisation automatique de documents se font ressentir de plus en plus. En effet, en raison de l'augmentation constante du volume d'informations accessibles électroniquement, la conception et la mise en œuvre d'outils efficaces, permettant notamment à l'utilisateur de n'avoir accès qu'à l'information qu'il juge pertinente, deviennent une nécessité absolue. Comme la plupart de ces outils sont destinés à être utilisés dans un cadre professionnel, les exigences de fiabilité et de convivialité sont très importantes ; les problèmes à résoudre pour satisfaire ces exigences sont nombreux et difficiles.

Dans ce chapitre, nous allons nous intéresser au domaine de la catégorisation de textes en présentant en premier lieu une définition formelle puis nous décrivons le processus général de la catégorisation en détaillant ces étapes, à savoir, la représentation de textes, la réduction de dimensionnalité et la classification. Puis, nous abordons l'évaluation de la qualité des classifieurs.

I.2 Historique :

La classification des textes remonte au début des années 60 et à partir des années 90, les chercheures réfléchissent aux moyens techniques permettant d'améliorer et d'enrichir ce domaine surtout avec l'apparition des algorithmes beaucoup plus performants qu'avant.

Les termes 'classification' et 'catégorisation' ont des histoires et des origines très différentes. Le terme classification est apparu pour la première fois dans la cinquième édition du dictionnaire de l'Académie Française en 1798 sous la définition : « distribution en classes et suivant un certain ordre » et dans la dernière édition par« l'Action de classifier et le résultat de cette action». Le terme 'catégorisation' n'existe pas dans le dictionnaire de l'Académie Française, contrairement au mot 'catégorie' qui est définit dans tous les éditions du dictionnaire comme étant une classe dans laquelle on range plusieurs choses qui sont des espèces différentes, mais qui appartiennent à un même genre **[1].**

l.3 Définition de la Catégorisation de texte (C.T) :

La catégorisation de texte (C.T) est de pouvoir associer automatiquement des documents à des catégories (classes, groupes, index) prédéfinies. Mr. Sébastian définit la C.T dans **[2]** comme étant le processus qui consiste à associer une valeur booléenne à

chaque paire (dj, ci) ∈ D × 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 dj appartient à la classe ci tandis que la valeur F (Faux) lui sera associée dans le cas contraire.

La catégorisation de texte consiste à chercher une liaison fonctionnelle entre un ensemble de textes et un ensemble de catégories (étiquettes, classes). Cette liaison fonctionnelle, que l'on appelle également modèle de prédiction, est estimée par un apprentissage automatique. Pour ce faire, il est nécessaire de disposer d'un ensemble de textes préalablement étiquetés, dit ensemble d'apprentissage, à partir duquel nous estimons les paramètres du modèle de prédiction le plus performant possible, c'est-à-dire le modèle qui produit le moins d'erreur en prédiction. [2]

I.4 Processus de la catégorisation de textes :

La C.T comporte quatre phases importantes comme illustre la **figure I.1**:

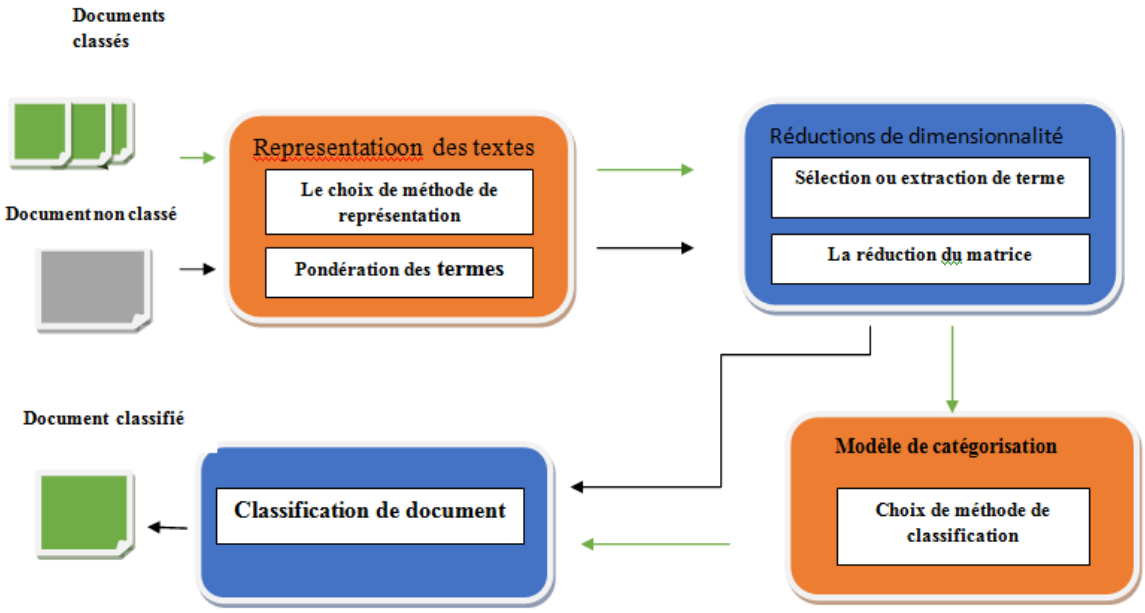


Figure I.1: Les étapes de catégorisation de texte.

1.4.1 La représentation de textes :

C'est une phase importante dans le processus de la C.T dont l'objectif consiste à présenter les documents textuel sous une forme exploitable par la machine. Pour cela il faut transformer le texte en un tableau à deux dimensions où les lignes représentent les documents et les colonnes représentent les descripteurs (termes). Le croisement entre la $i^{ème}$ ligne et la $j^{ème}$ colonne correspond au poids du $j^{ème}$ terme dans le $i^{ème}$ document.

Cette étape s'intéresse aussi au choix de la nature des termes. En effet plusieurs méthodes existent pour la représentation des textes :

- **Représentation en « sac de mots » :**

C'est la représentation de textes la plus simple, l'idée est de transformer les textes en vecteurs dont chaque composante représente un mot. Les mots ont l'avantage de posséder un sens explicite. Néanmoins, il est difficile de pouvoir délimiter les mots. Mr **R.JALAM** considère dans **[3]** le mot comme étant une suite de caractères appartenant à un dictionnaire, ou bien, de façon plus pratique, comme étant une séquence de caractères non-délimiteurs encadrés par des caractères délimiteurs (caractères de ponctuation).

Cette méthode présente plusieurs difficultés dont les plus reconnues sont :

- Absence de délimiteurs dans certaines langues comme par exemple la langue chinoise (人人生而自由, 在尊嚴和權利上一律平等)
- Certains délimiteurs peuvent faire partie des termes eux-mêmes. Ainsi, leur suppression peuvent détruire la sémantique des textes (exemples : pomme de terre, 172.16.28.147 et M. durant).

- **Représentation par lemmes ou racines lexicales :**

La représentation par lemmes ou racines lexicales consiste à remplacer les verbes par leur forme infinitive et les noms par leur forme singulière, il s'agit de regrouper les différentes formes graphiques que peut avoir un mot en une seule forme appelée forme canonique. Par exemple : les mots «déménageur, déménageurs, déménagement, déménagements, déménagé, déménagent, déménagera, etc.» sont considérés comme des descripteurs différents alors qu'il s'agit de la même racine « déménage ».

Ce regroupement permettra de réduire la dimensionnalité de l'espace de représentation ainsi que l'augmentation des occurrences des descripteurs. Le seul inconvénient de cette technique réside dans le fait qu'elle est indépendante de la langue du document. De ce fait, il est primordial de connaitre la langue du document.

- **Représentation des textes par des phrases :**

Etant donné que les phrases sont plus informatives que les mots seuls. Plusieurs travaux de recherche proposent d'utiliser les phrases pour représenter les textes. En effet, malgré la simplicité de l'utilisation de mots comme unité de représentation, il est parfois utile de conserver l'information relative à la position du mot dans la phrase.

Logiquement, une telle représentation doit obtenir de meilleurs résultats que ceux obtenus via les mots. Mais les expériences présentées ne sont pas concluantes car, si les qualités sémantiques sont conservées, les qualités statistiques sont largement dégradées.

- **Représentation N_gramme :**

Cette méthode consiste à représenter le texte en plusieurs séquences de n caractères consécutifs avec un déplacement d'un caractère. Ainsi, à chaque séquence qui suit, on doit commencer par le deuxième caractère de la séquence précédente.

La figure I.2 illustre un exemple de la méthode N-grammes.

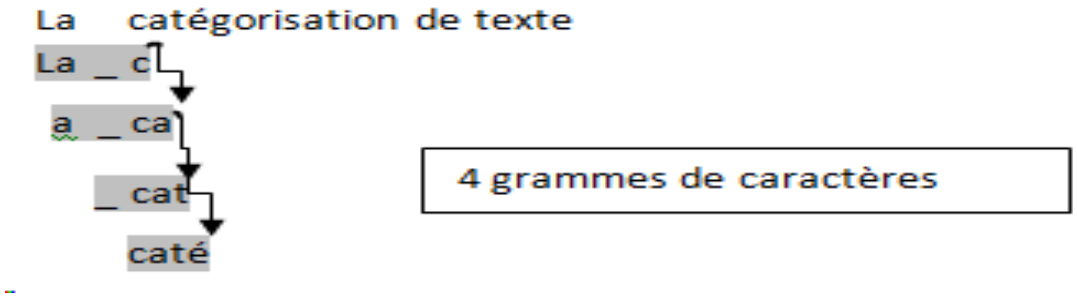


Figure I.2 : Exemple de N-grammes.

Cette technique a pour avantage la détection automatique des racines sans passer par les algorithmes linguistiques d'une part et son indépendance aux langues d'autre part. Néanmoins, il n'est plus possible de garder la sémantique du texte.

❖ **Représentation par concept :**

La représentation conceptuelle consiste à utiliser les ontologies et thésaurus dont le but de représenter le texte sous forme d'un ensemble de concepts. En effet, chaque concept représente un sens unique qui peut être exprimé par plusieurs mots synonymes. De même, un mot ayant plusieurs sens se retrouve mappé dans plusieurs concepts. Par exemple, les mots : voiture, véhicule, automobile peuvent être regroupés dans un seul concept.

Cette méthode présente l'avantage de réduire :

- le problème d'ambiguïté et de synonymie dans le vocabulaire et de la construction syntaxique (restituer l'ordre des termes).
- La dimensionnalité de l'espace de représentation. **[4]**

Néanmoins, les résultats de cette représentation dépendront de la richesse de la ressource sémantique utilisée.

I.4.2 Pondération :

Une fois les descripteurs sont choisis, cette étape consiste à calculer le poids reflétant l'importance du terme dans un document.

La pondération doit prendre en considération l'aspect local, l'aspect global ainsi que l'aspect de normalisation. Généralement la pondération d'un descripteur *i* dans un document *j* implique le calcul de : $W_{i,j=}L_{i,j}G_iN_j$ où :

- $L_{i,j}$ est la pondération locale. Cette pondération est généralement basée sur le nombre d'occurrences du descripteur dans le document (TF).

- G_i est la pondération globale du descripteur *i* dans la collection des documents. Cette pondération défavorise les descripteurs les plus communs de la collection (**IDF (Inverse of Document Frequency)**. **[5]**

- N_j est le facteur de normalisation des poids du document *j*. Ce facteur a pour but d'éliminer l'influence de la taille des documents.

Plusieurs méthodes de pondération existent, **Le tableau I.1** résume les méthodes de pondérations les plus utilisées.

Type de model	Model booléen	Model vectoriel
Pondération	$W_{kj} = \begin{cases} 1 & si \quad (t_k, d_j) \geq 1 \\ 0 & sinon \end{cases}$ W_{kj} le nombre d'occurence du terme t_k le document d_j	**La mesure IDF :** $IDF = \log\left(\frac{N}{DF(t_i)}\right)$ avec N: le nombre total de document et DF le nombre de documents contenant le terme La mesure TF*IDF : c'est la combinaison entre TF et IDF Avec TF la fréquence du terme dans le document (pondération local) la mesure TFC : $TFC = \frac{TF \times IDF(t_k, d_j)}{\sqrt{\sum_{s=1}^{\lvert t \rvert}(TF \times IDF(t_k, d_j))^2}}$

Tableau I.1 : les méthodes de pondération

I.4.3 Réduction de dimensionnalité :

Le problème majeur de la C.T réside dans la grande dimension de l'espace de représentation, un corpus de taille raisonnable contient des dizaines milliers de descripteurs. Le traitement de ces grandes quantités nécessite beaucoup d'espace mémoire et de temps de calcul.

Pour résoudre ce problème, on utilise les techniques de réduction de nombre des descripteurs, F.Sébastiani classe ces techniques dans **[2]** de deux façons :

a) selon qu'elles agissent localement ou globalement : Pour la réduction locale, il s'agit de proposer un nouveau ensemble de termes T' pour chaque catégorie C_i tel que |T'| <<|T| avec T les anciens termes. Chaque document d_j sera représenté par un ensemble de vecteurs d_j différents selon la catégorie. Au contraire, pour la dimension globale, il s'agit de choisir un nouvel ensemble de termes T'en fonction de toutes les catégories, et chaque document dj sera représenté par un seul vecteur.

b) La nature des résultats de la sélection (s'agit-il d'une sélection de termes ou d'une extraction de termes) : comme montre dans la figure **I.3,** les techniques de sélection de termes consistent à sélectionner un nouvel ensemble de descripteurs T' le plus pertinent parmi les descripteurs existant T, avec une taille plus petite que l'ancien. Parmi les techniques de sélection figures : MI (Mutuel Information), la méthode CHI-2, Gain d'Information (IG).

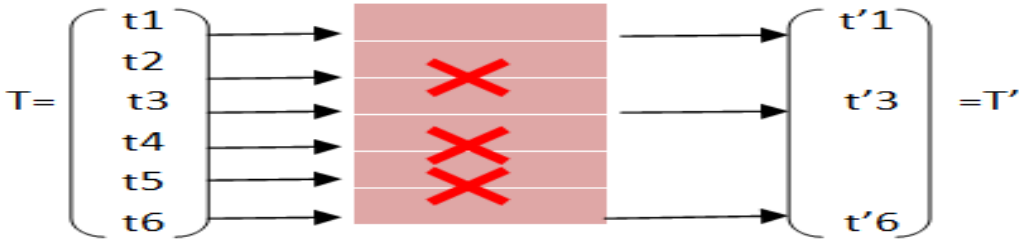


Figure I.3 : principe de sélection

Les techniques d'extraction ont l'objectif de proposer un sous-ensemble T avec |T'|<< |T | mais, à la différence des techniques de sélection, le sous-ensemble T'est une synthèse (combinaison linéaire des descripteurs) qui devrait maximiser la performance. La figure **I.4** illustre le principe d'extraction. Parmi les techniques de sélection figures : Clustering, LSA.

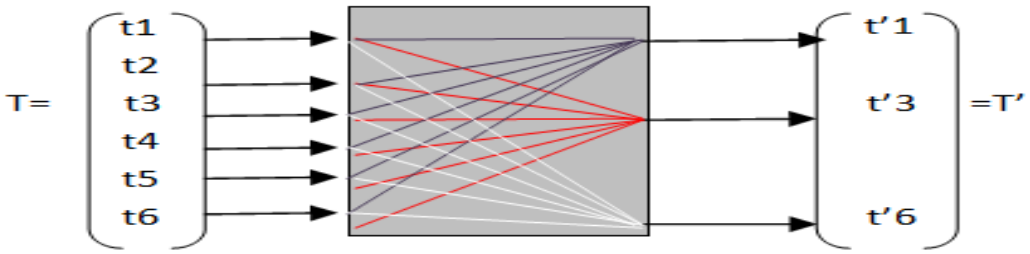


Figure I.4 : principe d'extraction

I.4.4 Choix de classifieurs :

Cette étape consiste à construire un classifieur autonome à la base des méthodes d'apprentissage. Dans ce qui suit, nous allons aborder les méthodes les plus utilisées :

- **La méthode de Rocchio** : son but est de réaliser une reformulation de la requête par son optimisation initiale de l'utilisateur, à travers la distance entre les documents pertinents et les documents non pertinents **[5].**
- **La méthode SVM((Support Vector Machin)** : Il s'agit de déterminer le meilleur séparateur entre classes dans l'espace de représentation. Par exemple dans **la figure I.5**, on remarque que le vecteur A est le meilleur séparateur parmi les trois séparateurs A, B et C parce qu'il est le plus distant de tous les éléments offrant **[5]** .

Théoriquement, on peut calculer la marge qui représente la plus petite distance entre les différentes classes et la surface séparatrice par la formule **[5]** suivante :

$$\boldsymbol{marge(s) = \sum_{c_j \in E} min_{x_i \in Cj}\ (d(x_i, s)} \quad \textbf{(I.1)}$$

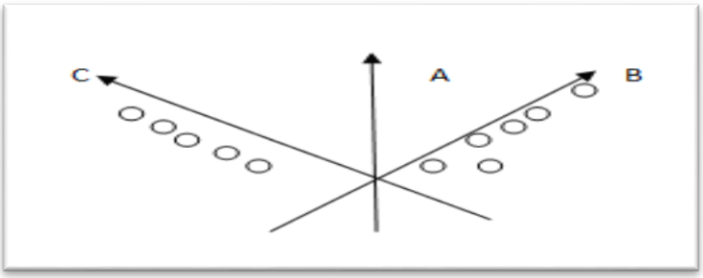


Figure I.5 Exemples de séparateurs

- **Les arbres de décision :**

Les arbres de décision sont des structures hiérarchiques en forme d'arbres, dont les nœuds portent une question, les arcs des réponses et les feuilles des conclusions ou des classes terminales.

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 pour effectuer la partition.**[5]**

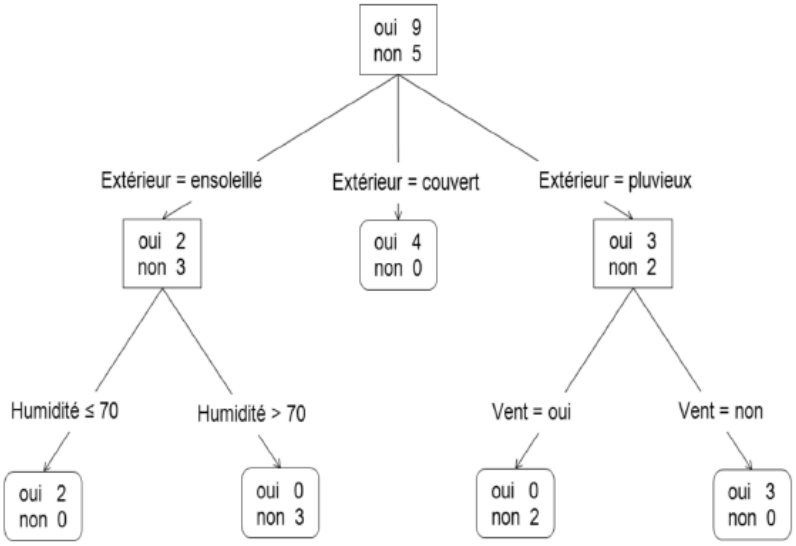


Figure I.6 Exemples d'arbre de décisions

- **Les k plus proches voisins (k-PPV) :**

C'est une des méthodes d'apprentissage supervisé les plus utilisées, elle consiste à représenter chaque document dans un espace vectoriel, et de déterminer les k documents les plus proches au document à classer en utilisant une mesure de similarité. Le document sera assigné à la catégorie la plus représentée dans les k plus proches documents. Elle nécessite beaucoup d'espace mémoire pour stocker tous les k documents et beaucoup temps de calcule.

I.4.5 Evaluation de la qualité des classifieurs :

Afin d'assurer que le classifieur construit est généralisable à d'autres textes. Il est nécessaire d'évaluer la performance du classifieur. L'évaluation consiste à mesurer la différence entre un résultat attendu et un résultat obtenu.

Les performances en termes de classification sont généralement mesurées à partir de deux indicateurs :

- **Précision** : Elle mesure la proportion de documents pertinents relativement à l'ensemble des documents restitués par le système.

- **Rappel** : Il mesure la proportion de documents pertinents restitués par le système relativement à l'ensemble des documents pertinents contenus dans la base documentaire

La figure ci-dessous illustre le concept de ces critères.

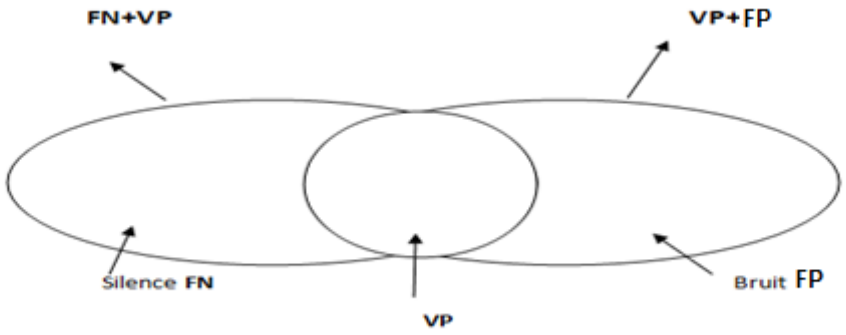


Figure I.7 principe de Précision et Rappel

Les expressions ci-dessous représentent les formules de précision et rappel :

$$P(C_i) = \frac{VP_i}{VP_i + FP_i} \qquad \textbf{(I.2)}$$

$$R(C_i) = \frac{VP_i}{VP_i + FN_i} \qquad \textbf{(I.3)}$$

Où :

$\boldsymbol{VP_i}$: représente le nombre de documents correctement attribués à la catégorie.

$\boldsymbol{FP_i}$: représente le nombre de documents incorrectement attribués à la catégorie.

$\boldsymbol{FN_i}$: représente le nombre de documents qui auraient dû lui être attribués mais qui ne l'ont pas été.

La mesure F combine les deux mesures complémentaires (précision et rappel) selon la formule suivante :

$$\boldsymbol{F_\beta(c_i) = \frac{(\beta^2+1)\times P(c_i)\times R(c_i)}{\beta^2 \times P(c_i) + R(c_i)}} \qquad \textbf{(I.4)}$$

La valeur du facteur β permettra de favoriser une mesure par rapport à l'autre, une valeur β=1 donnera la même importance au deux mesures.

I.5 Les applications de la catégorisation des textes :

Pratiquement la C.T est appliquée dans la plupart des domaines du monde réel par exemple :

- Les applications de classification des documents en fonction de leurs sujets et les applications de filtrage des spams (spam /ou non spam). La **figure I.8** illustre un système de classification d'emails où les classes peuvent être de différentes natures (thèmes, messages provenant de certaines personnes, messages d'un certain type, courrier urgent, spam)[4].
- L'identification de la langue : il s'agit d'utiliser un ensemble de documents étiquetés dans plusieurs langues afin de catégoriser des documents de différentes langues.
- La reconnaissance d'écrivains dans la recherche documentaire.
- La catégorisation de documents multimédia.

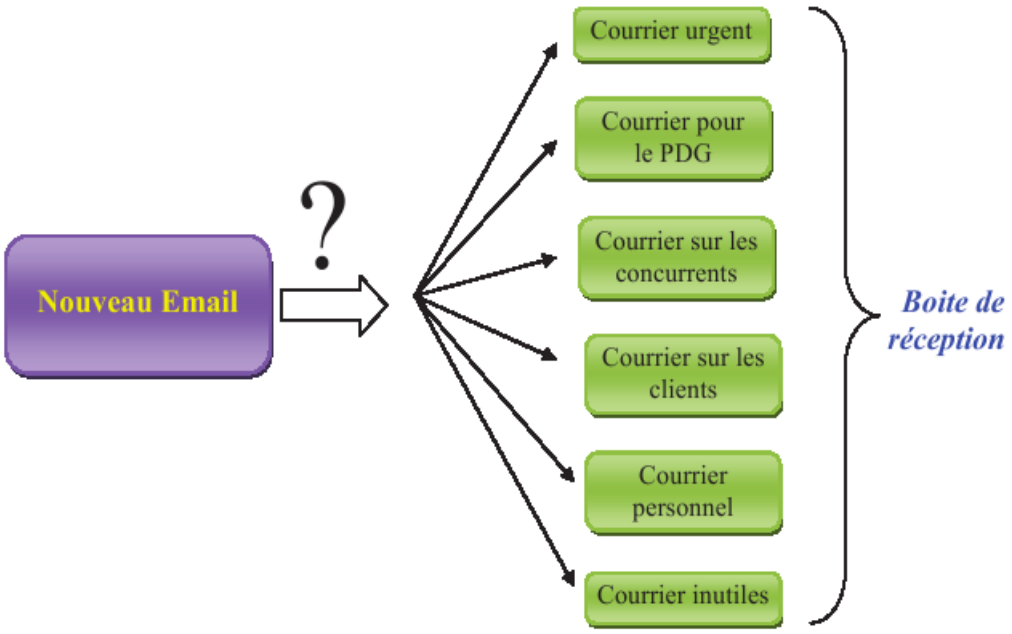


Figure 1.8 : Exemple de système de classification d'emails

I.6 Les problèmes de la C.T :

Le processus de C.T provoque plusieurs problèmes, certains problèmes peuvent être originaires de **l'apprentissage automatique supervisé** tel que le problème de la subjectivité de la décision prise par les experts ou le problème de sur-apprentissage.

D'autres problèmes sont relatifs **à la nature des données traitées** à savoir les données textuelles. Parmi les problèmes les plus répondus figurent :

- **Ambigüité :**

L'ambiguïté peut être lexique dans le cas des termes ayant la même forme mais des sens différents comme elle peut être syntaxique suite à une déduction erronée de l'agencement des mots dans la phrase. L'ambiguïté peut être aussi de niveau sémantique où un mot peut avoir plusieurs sens possibles selon son contexte d'utilisation. Il existe aussi de multiples manières d'exprimer la même réalité, avec des nuances diverses. Deux mots ou expressions seront dits synonymes s'ils ont le même sens ; exemples : mon chat mange un oiseau, mon gros matou croque un piaf et mon félin préféré dévore une petite bête à plumes. On voit bien qu'il s'agit d'un chat qui mange un oiseau mais pourtant les trois textes ne partagent aucun mot autre que des mots-outils (mon, un)**[5].**

- **Redondance(Synonymie)** *:*

Il s'agit d'exprimer le même concept par des expressions différentes, plusieurs façons d'exprimer la même chose contrairement aux données numériques. Le problème imposé est lié à la nature des documents traités exprimée en langage nature.

- **L'homographie :**

L'homographie est une sorte d'ambigüité supplémentaire, signifie que deux mots ayant la même écriture sans forcément avoir la même prononciation. (Ex : avocat en tant que fruit est avocat en tant que juriste).

- **La graphie :**

Signifiée dans **[4]** qu'un terme peut être écrit de plusieurs manières : il peut comporter des fautes d'orthographe, de frappe ou écrit avec une majuscule, ce qui va influencer les résultats de la recherche puisque les différentes graphies vont être traitées séparément.

- **Les variations morphologiques :**

Les conjugaisons, pluriels, influent négativement sur la qualité des résultats puisque les différentes variations morphologiques vont êtes considérées séparément et chacune va être prise comme un élément à part comme par exemple les trois termes : maître, maîtresse, maîtriser est traités indépendamment quoique en réalité ça pivote sur la même idée **[4].**

- **Les mots composés :**

le manque de traitement des mots composés par exemple Arc-en-ciel, peut-être, sauve-qui-peut, etc.. Dont le nombre est très important dans toutes les langues, où traiter le mot Arc- en-ciel en étant 3 termes séparés réduits considérablement la performance d'un système de classification. Néanmoins l'utilisation de la technique des n -grammes pour le codage des textes atténuent considérablement ce problème des mots composés**. [4]**

I.7 Conclusion :

La catégorisation de textes est actuellement un sujet à la pointe de la recherche, en particulier dans des domaines tels que la recherche d'informations, la recommandation et la personnalisation des profils d'utilisateurs.

Nous avons présenté dans ce chapitre la définition de base de la C.T puis nous avons détaillé les étapes du processus de catégorisation ainsi que les différentes méthodes utilisées pour l'évaluation des performances des systèmes de classification. Finalement, nous avons cité quelques problèmes rencontrés dans les systèmes de catégorisation.

Chapitre II : les mesures de similarité sémantique

II.1 Introduction :

Plusieurs disciplines telles que le web sémantique et l'intelligence artificielle adoptent le domaine de classification de texte. En effet, ce dernier repose largement sur des mesures pour l'identification de la similarité entre les documents. La majorité des mesures sont fondées sur des aspects statistiques ne prenant pas en considération les relations sémantiques existant entre les mots de la langue. Durant ces dernières années, plusieurs travaux de recherches relatives au domaine du web sémantique ont proposé de nouvelles mesures de similarité basées sur les relations sémantiques entre mots de la langue.

Dans ce chapitre, nous allons aborder en premier lieu les inconvénients des mesures de similarité statistiques ; puis nous détaillerons les mesures de similarités sémantiques proposées dans la littérature.

II.2 C'est quoi une mesure de similarité ?

Selon Maria dans **[6]** Une mesure de similarité S est une fonction X *X → R qui satisfait les propriétés suivantes :

1. Positivité : ∀x, y ∈ X, S(x, y) ≥ 0.
2. Symétrie : ∀x, y ∈ X, S(x, y) = S(y, x).
3. Maximalité : ∀x, y ∈ X, S(x, x) ≥ S(x, y).

II.3 Les mesures de similarités statistiques :

Parmi les modèles de la CT qui utilisent les mesures statistiques, on trouve le modèle booléen qui se base sur la présence / absence de terme dans un document, et le modèle vectoriel qui utilise les fonctions de similarité, on peut distingue les meures suivants :

- **Produit scalaire :** se calcule par la formule suivante

$$\sum_{t \in T} W_{t,A} * W_{t,B} \qquad \textbf{(II.1)}$$

Où :

T : représente l'ensemble des attributs.

A et B : deux documents (A le document a classé) B le document classé.

$W_{t,A}$: représente le poids de terme dans le document A.

$W_{t,B}$: représente le poids de terme dans le document B.

Le seul inconvénient de produit scalaire c'est que les résultats obtenus sont non normalisés.

- **Cosinus :** afin de pallier l'inconvénient de produit scalaire, les chercheurs proposent la mesure de cosinus qui donne des résultats normalisés qui est entre 0 et 1, elle est définie par la formule suivante :

$$\sum_{t\in T} \frac{p_t(a).p_t(b)}{\sqrt{\sum_{t\in T} p_t(a)^2}.\sum_{t\in T} p_t(b)^2} \qquad \textbf{(II.2)}$$

Avec :

T : représente l'ensemble des attributs.

$p_t(a)$: représente le poids du terme t dans le document a.

$p_t(b)$: représente le poids du terme t dans le document b.

- D'autres fonctions de similarité ont été proposées dans la littérature, parmi lesquelles on peut citer la mesure de Dice qui est défini par la formule suivante :

$$\text{Dice (A, B)} = \frac{2N_c}{N1+N2} \qquad \textbf{(II.3)}$$

Avec

- A et B deux documents
- N_c est le nombre de termes communs entre A et B, et N1 (resp. N2) est le nombre de termes dans le document A et (resp. B).

Il ya aussi la mesure de **JACCARD** qui définit par **:**

$$JACCARD(A,B) = \frac{2*\sum_{i=1}^{t} A_i*B_j}{Min(\sum_{i=1}^{t} A_i^2 \sum_{i=1}^{t} B_i^2)} \qquad \textbf{(II.4)}$$

Tel que :

- A et B deux documents (A le document a classé)
  B le document classé

II.4 Les mesures de similarité sémantiques:

Les mesures de similarité sémantique se basent sur les relations entre les mots de la langue, de ce fait, plus les documents contiennent des termes similaires sémantiquement plus les documents sont proches.

Pratiquement pour utiliser ces mesures, il est nécessaire de définir des bases de connaissances qui regroupent les termes et les relations et les règles qu'elles permettent de combiner les termes et les relations comme les ontologies. Le seul problème de ces mesures c'est la dépendance vis-à-vis de l'outil linguistique, plus l'outil est riche plus les résultats seront efficaces.

Afin de trouver la similarité sémantique entre les termes, plusieurs mesures ont été proposées. Ces mesures se classent en trois catégories selon qui utilisent les arcs de l'outil linguistique, ou des nœuds, ou la combinaison entre les deux, comme illustre **la figure II.1.**

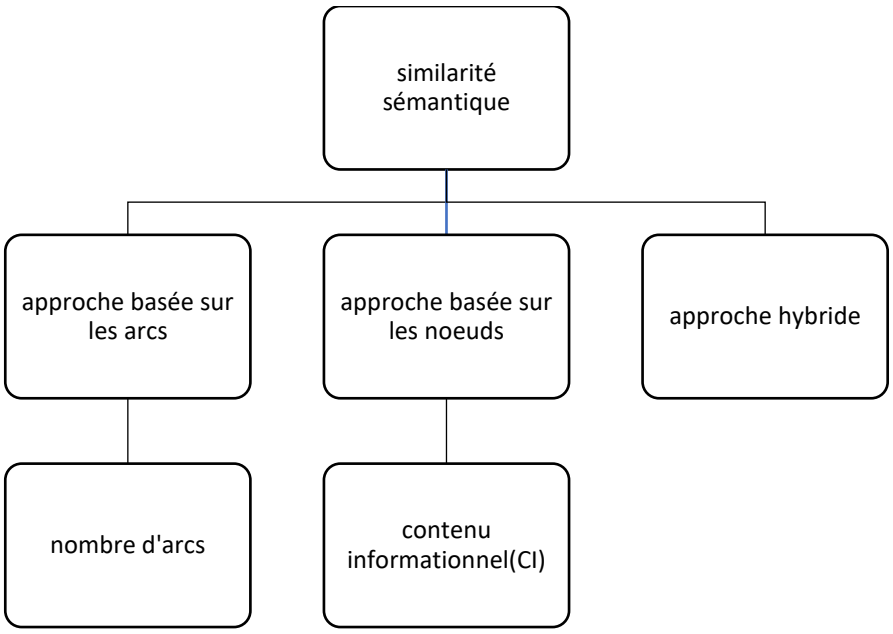


Figure II.1 : Les approches de mesure de similarité sémantique.

II.4.1 Approche basée sur les arcs :

Cette approche se base sur la structure hiérarchique de l'ontologie qui est représentée par un graphe tel que les nœuds sont des concepts, et les arcs sont les liens entre ces concepts, afin de compter le nombre d'arcs séparant deux concepts.

La **figure II.2** montre une ontologie Ω formée par un ensemble de nœuds et un nœud racine R, soit c1 et c2 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 c1 et c2 du nœud racine et la distance qui sépare le concept subsumant2 (CS) de c1 et de c2 du nœud **[7]**

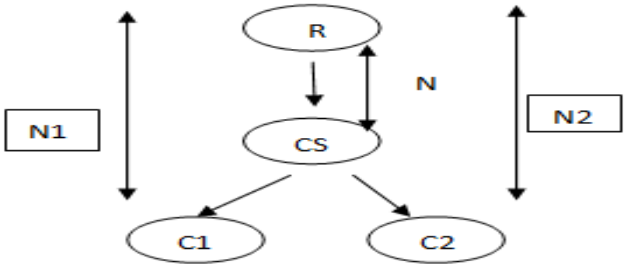


Figure II.2 : exemple d'un extrait d'ontologie

Parmi les mesures de cette approche on peut citer :

- **Mesure de Rada et al**

Cette mesure est utilisée dans un réseau sémantique. Elle permet de calculer la similarité en se basant sur les liens hiérarchiques « is -a» à travers le calcul de la distance entre les nœuds par le plus cours chemin **[8],** elle est définie par la formule suivante**:**

$$sim_{rada}(c_1, c_2) = \frac{1}{1+dist(c_1,c_2)} \quad \textbf{(II.5)}$$

Avec

Dist (c1; c2) : indique le nombre d'arcs minimum à parcourir pour aller d'un concept c1 à un concept c2.

Elle présente un moyen des plus évidents pour évaluer la similarité sémantique dans une ontologie hiérarchique .Mais elle ne prend pas en compte la profondeur des concepts.

- **Mesure de Wu & Palmer**

La mesure de Wu et Palmer est définie par la formule suivante **[9]** :

$$Simc(c_1, c_2) = \frac{2 \times prof(c)}{prof(c_1) + prof(c_2)} \qquad \textbf{(II.6)}$$

Avec :

$prof(c)$: La profondeur du concept ci, c'est-à-dire la distance à la racine de ci ;

c le plus petit ancêtre commun à c1 et c2. Certaines autres prennent en compte la Profondeur de la hiérarchie.

Cette mesure simple et facile à implémenter et prend en compte la profondeur des concepts. Par contre, elle ne donne pas une bonne similarité entre concepts voisins et concepts de la même hiérarchie.

II.4.2 Approche basé sur les nœuds :

Cette approche a pour objectif de surmonter les limites de l'approche basée sur les chemins, ceci en complétant la structure taxinomique d'une ontologie avec la distribution de l'information de concepts évalués dans des corpus d'apport. Elle exploite la notion du contenu de l'information (IC), en associant des probabilités P d'apparence à chaque concept c dans la taxinomie, calculée à partir de leur présence dans un corpus donné. L'IC associé à c'est tel qu'elle est définie dans **[7]** par la formule suivante :

$$IC(c) = -\log\big(P(c)\big) \qquad \textbf{(II.7)}$$

Elle améliore les mesures précédentes en augmentant les concepts avec leur contenu informatif (Information Content, IC)**.**

- **Mesure de Resnik :**

Resnik **[10]** a été le premier qui a fusionné l'utilisation d''ontologie avec le corpus, elle permet de mesurer la quantité d'informations partagées entre deux concepts qui égale au contenu informationnel du plus petit généralisant (PPG) c'est-à-dire le concept le plus spécifique qui subsume les deux concepts dans l'ontologie dont elle est définie par la formule suivante.

$$Sim_{Res}(C_1, C_2) = IC(PPG(C_1, C_2)) \qquad \textbf{(II.8)}$$

Cette mesure offre une très bonne performance, par contre elle est un peu sommaire car elle ne dépend que du concept le plus spécifique**.**

- **Mesure de Lin :**

Lin a utilisé une approche hybride qui permet de combiner deux sources de connaissances différentes (Thesaurus, corpus) **[11].** Elle est légèrement différente de celle de Resnik, dont elle représente la similarité comme degré probabiliste de chevauchement des concepts descendants de X et Y. Cette mesure a été évaluer par Les travaux de Mil **[12]**, qui utilise des sujets humains pour évaluer la similarité entre 30 paires de noms, il en ressort que cette méthode offre une amélioration significative**,** elle est représentée par la formule suivante :

$$sim(c_1, c_2)_{lin} = \frac{2 * CI\left(LCT(c_1, c_2)\right)}{GI(c_1) + GI(c2)} \qquad \textbf{(II.9)}$$

- **Mesure de Hirst et Onge**

L'idée de cette mesure **[13]** 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 :

$$\mathrm{sim}_{\mathrm{hirst}}(c_1, c_2) = c - \mathrm{len}(c_1, c_2) - k * t(c_1, c_2) \qquad \textbf{(II.10)}$$

Avec :

C et k sont des constant.

Len : le plus court chemin

T : changement de direction**.**

II.4.3 Approches hybrides :

L'idée de cette approche est avoir une possibilité de combiner 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. Parmi ces mesures :

- **La mesure de Jiang-Conrath**

La mesure de Jiang-Conrath **[14]** pallie les limites de la mesure de Resnik en combinant le contenu informationnel du PPG à ceux des concepts. Elle prend en considération aussi le nombre d'arcs. Ainsi une distance est définie par :

$$dis(c_1, c_2) = \mathrm{CI}(c_1) + \mathrm{CI}(c_1) - (2 * \mathrm{CI}(\mathrm{ppg}(c_1, c_2))) \qquad \textbf{(II.11)}$$

La mesure de similarité devient donc :

$$sim(c_1, c_2) = \frac{1}{dis(c_1, c_2)} \qquad \textbf{(II.12)}$$

- **Mesure de Leacock et Chodorow**

Cette mesure **[15]** est basée sur la longueur du plus court chemin entre deux sens. 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 taxinomie. La formule est définie par:

$$\mathrm{Sim}(x, y) = -\log\left(\frac{cd(x,y)}{2*M}\right) \qquad \textbf{(II.13)}$$

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.

II.5 Conclusion :

Afin d'améliorer l'efficacité de la CT et leur résultat, nous avons présenté dans ce chapitre une extension de mesure de similarité avec leurs différents types.

Premièrement nous avons commencé par le type statistique avec leurs inconvénients, puis nous avons entamé les mesures de similarité sémantique et leurs trois grandes familles d'approche telle que nous avons cité les approches basées sur les nœuds utilisant des mesures du contenu informationnel, puis L'autre approche se base sur les distances des arcs et prendre en considération le plus court chemin entre les nœuds. Finalement l'approche hybride qui combine entre les deux premières approches.

.

Chapitre III : Implémentations de l'application et l'évaluation des résultats

III.1Introduction :

Avec l'arrivé de web sémantique ainsi que les ontologies, un autre axe de recherche focalise l'utilisation de web sémantique dans la CT, en effet ;

L'utilisation de la sémantique peut être sollicité au niveau de la représentation conceptuelle comme elle peut être dans le calcule de similarité. Afin d'améliorer la performance de classification et pallier aux inconvénients des mesures de similarité statistiques, nous avons opté pour un système de catégorisation de textes intégrant la sémantique au niveau de la représentation des documents ainsi qu'au niveau de la classification pas le biais des mesures de similarité sémantique.

Ce chapitre est essentiellement consacré aux grandes lignes qui visent à réaliser l'objectif de ce thème, à savoir les étapes de notre application, le choix du langage de programmation, l'environnement de programmation ainsi que les ressources exploitées pour le développement d'application**.**

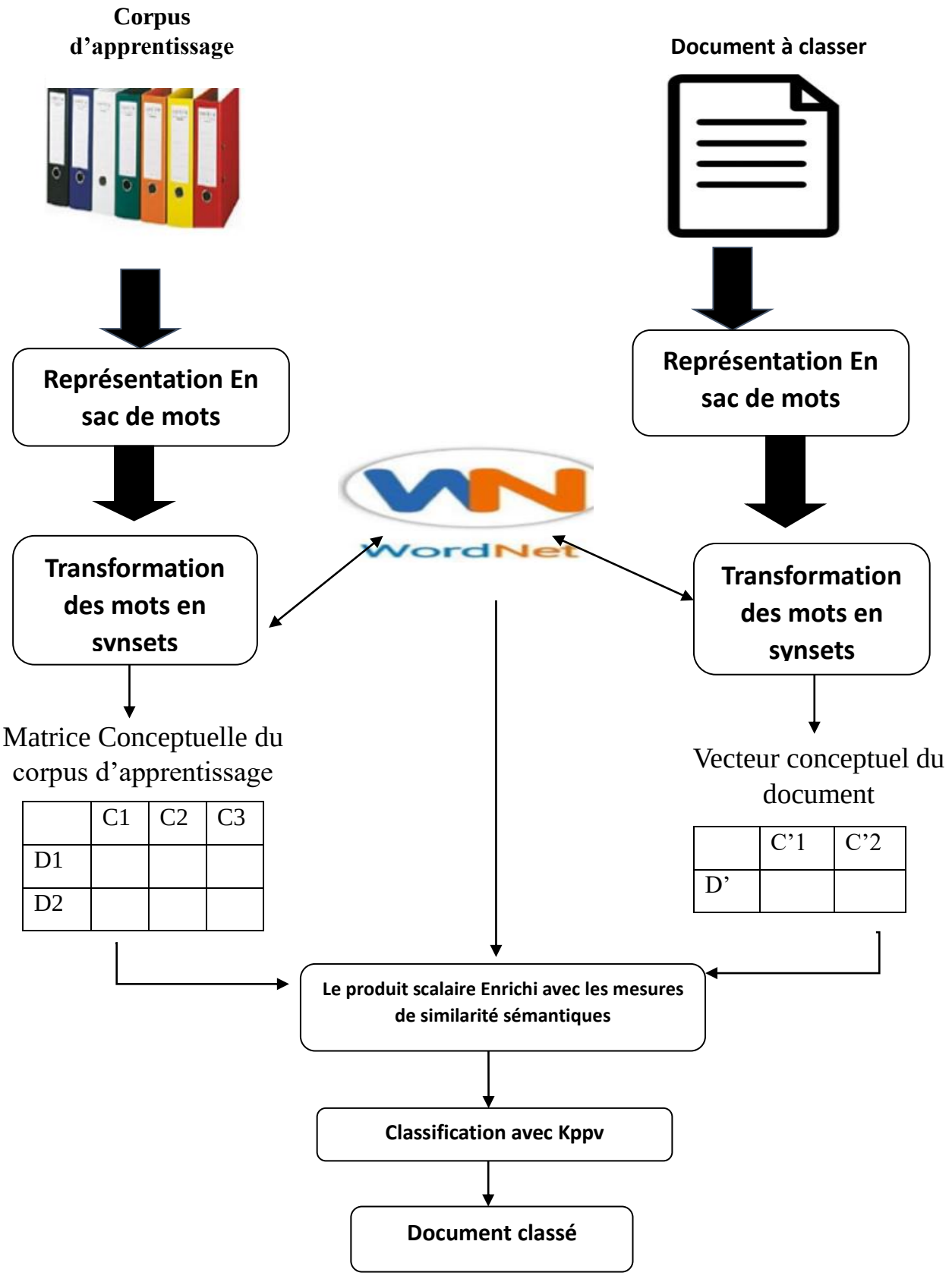
III.2 Etapes de notre système :

L'objectif de notre travail est de réaliser un système permettant de catégoriser un document textuel en se basant sur des documents préalablement classés. Cela nécessite de créer une liaison fonctionnelle entre l'ensemble de documents et l'ensemble de classes (catégories). Afin d'avoir une meilleur classification, on a essayé d'apporter une nouvelle mesure de similarité entre documents qui se base sur l'utilisation des mesures de similarité sémantiques entre concepts afin d'aboutir à une extension sémantique du produit scalaire. Cette nouvelle mesure de similarité et par la suite utilisé dans l'étape de classification pour calculer le rapprochement entre le document à classer et les documents préalablement classés dans le but de lui assigner une classe L'utilisation des mesures de similarité sémantiques entre concept exige de passer par une représentation conceptuelle des documents textuels.

Comme tout processus de la C.T, notre système comporte deux étapes :

- Une étape de représentation permettant de transformer les documents textuels sous une forme exploitable par la machine.
- Une étape de classification permettant d'effectuer une liaison fonctionnelle entre documents et classes afin de pouvoir catégoriser le nouveau document.

La **figure III.3** illustre les étapes de notre système.



	C1	C2	C3
D1			
D2			

	C'1	C'2
D'		

Figure III.1 : Etapes de notre système

III.2.1 Représentation de documents :

Cette étape consiste à transformer chaque document (les documents classés ainsi que le document à classer) en vecteur dont chaque composante représente un mot. Cette transformation nécessite l'exécution des traitements suivants :

Catégorie1 : pays
Document 1 (you , visit Algeria ?. africa goal).
Document 2 (My name ,is Algeria!) .
Document 3 (my country, is Egypt..).
Catégorie2 : sport
Document1 (kicking a ball ,to score a goal sport !.)
Document2(athletics,physical)

Figure III.2 : Exemple de corpus

- **Tockénisation (analyse lexicale) :** cette étape permet de convertir les textes en un ensemble de mots appelé « dictionnaire ». En considérant qu'un mot est une suite de lettres comprise entre deux séparateurs, il est indispensable d'utiliser une liste stockant les séparateurs de la langue dont laquelle les documents sont exprimés. La figure **III.3** présente la liste de séparateurs utilisée dans notre système.

```
",;:(){}\\-_0123456789 '.?!$'\"<>\\r\\n\\t&/#"
```

Figure III.3 : la liste de séparateur

Le tableau ci-dessous montre le résultat de la Tockénisation sur l'exemple illustré dans la **figure III.2.**

Africa	You	Visit	Algeria	My	Name	Is	algeria	My	country	is
egypt	kiking	A	ball	To	score	A	goal	Athletics	physical	sport

Tableau III.1 : le résultat d'appliqué la Tockénisation sur l'exemple II.2

- **Elimination des mots vides :** Cette étape permet de supprimer les mots qui se répètent fréquemment dans tout les documents et qui n'ont aucun pouvoir discriminent lors du processus de la catégorisation de textes. La **figure III.4** illustre la liste des mots vides utilisés dans notre système.

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 III.4 : la liste des mots vides

Le tableau III.2 représente le résultat de l'élimination des mots vides sur l'exemple III.1.

africa	Visit	algeria	name	Algeria	country	egypt	kiking	ball	score
goal	athletics	physical	sport						

Tableau III.2 : les résultats d'élimination des mots vides sur l'exemple III.1

- **Pondération :** Dans cette étape, il s'agit de mesurer l'importance de chaque mot dans chaque document afin de construire une matrice (documents*termes) où les lignes représentent les documents et les mots représentant les colonnes. L'intersection entre la ligne i et la colonne j représente le poids du $j^{ième}$ mot dans le $i^{ième}$ document. Dans notre travail nous avons utilisé la mesure de pondération TF IDF (voir chapitre I).

  Le tableau ci-dessous représente la matrice documents*termes résultante de l'exemple III.2.

terme / doc	Africa	Algeria	Country	Egypt	Goal	Name	Visit	Athletics	Ball	Kicking	Physical	Score	sport
Doc 1	1.12	0.63	0	0	0.63	0	1.12	0	0	0	0	0	0
Doc 2	0	0.63	0	0	0	1.11	0	0	0	0	0	0	0
Doc 3	0	0	1.12	1.12	0	0	0	0	0	0	0	0	0
Doc 4	0	0	0	0	0.63	0	0	0	0.63	1.12	0	1.12	1.12
Doc5	0	0	0	0	0	0	0	0.63	0	0	0.63	0	0

. **Tableau III.3** : La matrice documents*termes résultante de l'exemple III.2

- **Maping des termes en concept(Synset) :**

Cette étape consiste à remplacer chaque mot par son concept adéquat selon l'ontologie utilisée dans le but d'aboutir à une représentation conceptuelle de documents. Dans notre travail, nous avons utilisé le thésaurus WordNet, qui regroupe tous les mots ayant le même sens sous une forme appelé « synset » (Synonyme Set). Sachant qu'un mot de la langue peut avoir plusieurs sens (voir l'exemple de la **figure III.5**), la base lexicographique WordNet renvoie une liste ordonnée de synsets pour chaque mot selon sa nature grammaticale (nom, verbe, adverbe ou adjectif). Afin de pouvoir sélectionner le sens adéquat du mot dans le texte, plusieurs méthodes de désambiguïsation existent. Dans notre travail, nous avons utilisé les deux méthodes de désambiguïsation suivantes :

- ✓ **Tous les sens** : Il s'agit de prendre en considération tous les sens proposés.
- ✓ **Premier sens** : Il s'agit de prendre seulement le premier sens en considération.

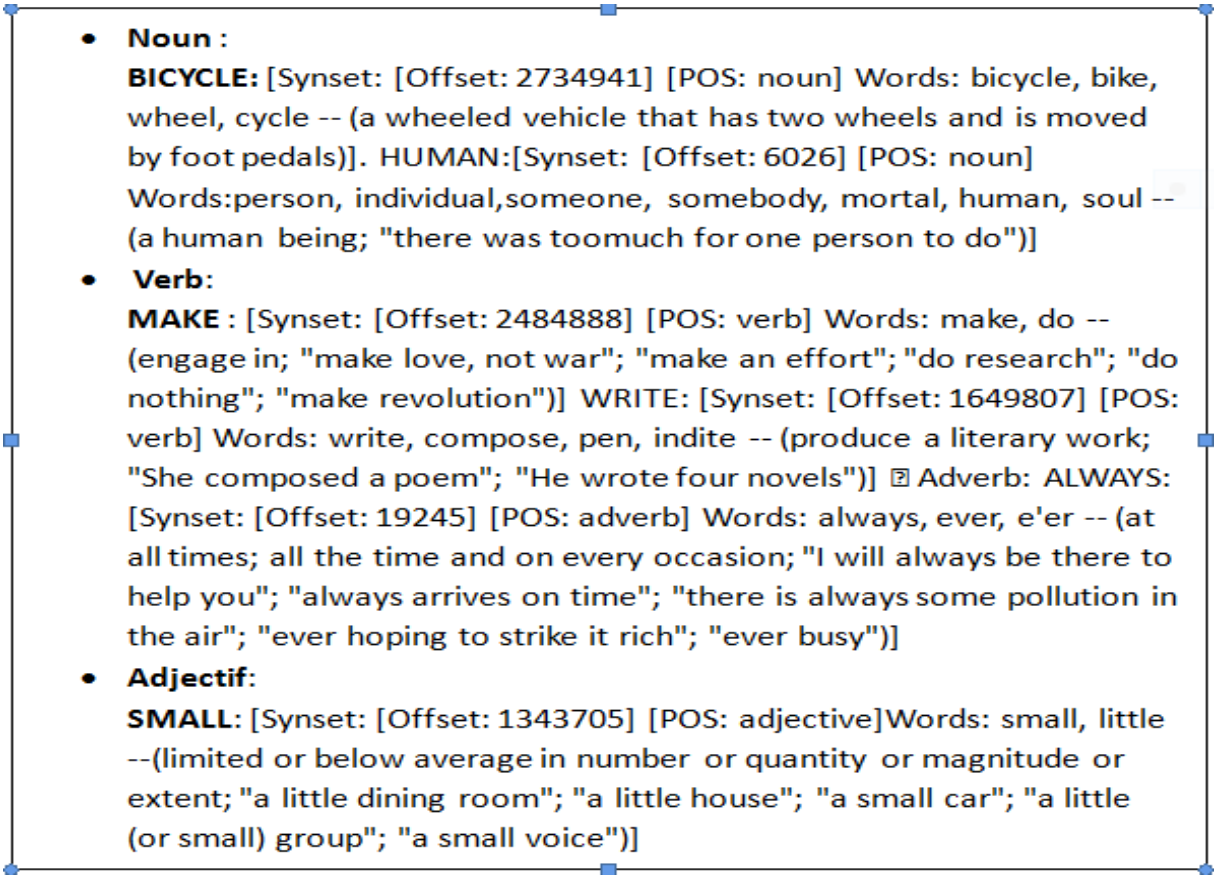
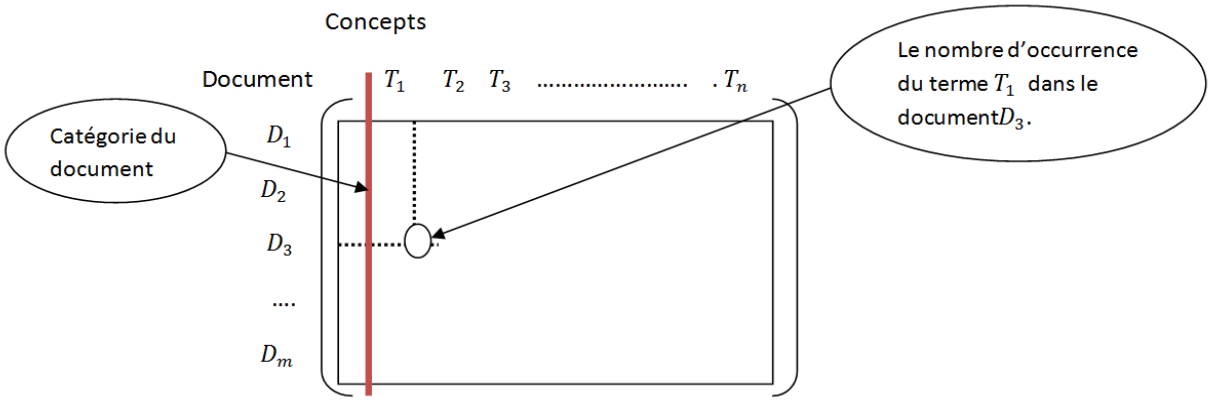


Figure III.5 : Exemple d'un groupe de synsets

Une fois le Maping de mots en synsets effectué, la matrice documents*termes sera transformée en une matrice documents*concepts. La figure III.6 illustre la représentation matricielle d'un corpus ou chaque ligne i représente le document D_i et chaque colonne j représente le concept (terme) T_j. L'intersection entre un document D_j et un terme T_j représente le nombre d'occurrences du concept T_j dans le document D_i.



La figure III.6 : représentation conceptuelle

La figure III.7 présente la matrice documents*concepts du corpus de l'exemple III.2. La première colonne représente la classe (0 la classe pays et 1 la classe sport) du document, si on prend la valeur d'intersection le colonne 2 avec la ligne 1 qui est 1,115577 c'est la TFIDF de sens 9057648 dans le doc1

```
@data
pays,1.115577,0.635124,0,0,0.635124,0,1.115577,0,0,0,0
pays,0,0.635124,0,0,0,1.115577,0,0,0,0,0
pays,0,0,1.115577,1.115577,0,0,0,0,0,0,0
sport,0,0,0,0,0.635124,0,0,0.635124,1.115577,1.115577,1.115577
sport,0,0,0,0,0,0,0,1.750702,0,0,0
```

Figure II.7 : Représentation conceptuelle de l'exemple III.1

III.2.2 Classification :

Cette étape consiste à utiliser la représentation conceptuelle du corpus d'apprentissage ainsi que le vecteur conceptuel du document à classer afin de pouvoir prédire la classe de ce dernier. Plusieurs méthodes de classification existent (voir chapitre 1). Dans notre

travail, nous avons utilisé la méthode de classification Knn (K Nearest Neighbor). Comme indiqué dans l'algorithme de la figure III.8. Le principe consiste à assigner le document à classer à la classe la plus représentée parmi ces K documents les plus proches.

Entrées : La matrice documents*concepts

Le vecteur conceptuel du document à classer

Le nombre K de voisin

Le Dictionnaire de concepts

Début :

Pour chaque document classé d_j **faire**

Calculer la mesure de similarité *entre le vecteur conceptuel du document d_j et le vecteur conceptuel du document à classer ;*

Fin pour

Trier les similarités calculées par ordre décroissant ;

Assigner le document à la classe *c* la plus représentée dans les k documents ayant les plus grandes valeurs de similarité ;

Fin

Figure III.8 : algorithme de Knn.

Afin de pouvoir trouver les k plus proches documents, il est nécessaire de calculer la similarité entre le vecteur conceptuel du document à classer avec chaque document du corpus d'apprentissage. Plusieurs mesures de similarité existent (voir chapitre II). Dans notre travail, nous avons opté pour une nouvelle mesure de similarité basée sur l'enrichissement du produit scalaire sémantiquement via l'utilisation des mesures de similarité sémantiques. En effet, le produit scalaire comme toute mesure de similarité statistique, ne prend pas en considération la notion de synonymie entre mots ou concepts. Si on prend l'exemple de **figure II.9**, qui montre deux documents qui sont proche l'un à l'autre et tous les deux sont relatifs au domaine de sport, mais ils ne partagent aucun mot commun et par conséquence la mesure du produit scalaire ne détectera pas le rapprochement entre ces deux documents (Produit scalaire (Document1, document2)=0).

$$PS(doc1, doc2) = \sum_{t=0}^{n} W_{t,1} * W_{t,2}$$

Avec $W_{t,1}$ Représente Le poids de terme dans le document 1

Et $W_{t,2}$ le poids de terme dans le document 2

Figure II.9 : exemple montrant l'inconvenant du produit scalaire

Comme montré dans l'algorithme de la **figure III.10.** Pour tout concept « c » du document à classer, deux cas de figures se présentent :

1- Si le concept existe dans les documents classés, la similarité est mise à jour selon la formule suivante :

$$PS_N = PS_N + W_{c,d1} * W_{c\prime,d2}$$

2- Si le concept n'existe pas dans les documents classés, nous utilisons les mesures de similarité sémantique afin de trouver le concept « sp » le plus proche (ayant la plus grande valeur de similarité $MSmax$) parmi les concepts figurant dans les documents classés. Ainsi, la similarité est mise à jour selon la formule suivante :

$$PS_N = Ps_N + W_{s,d1} * W_{sp,d2} * MSmax$$

Les entrées : vecteur du doc1 classé, vecteur du doc2 à classer,

S1 synsets du doc1 classé,

S2 synsets du doc2 non classé

Les paramètres : MS sémantique, , PS_N , $W_{c,d1}$, $W_{c\prime,d2}$,maxMS.

Début:

PS_N=0 ;

Pour chaque sens s de S1 faire

Si le sens s existe dans le doc2 alors

$$PS_N = PS_N + W_{s,d1} * W_{s,d2}$$

Sinon

MSmax=0 ;

Pour chaque sens s' du doc2 faire

M=Calculer MS(s , s')

Si M > MSmax alors

MSmax=M ;

Sp=s' ;

Fin pour

$$PS_N = Ps_N + W_{s,d1} * W_{sp,d2} * MSmax$$

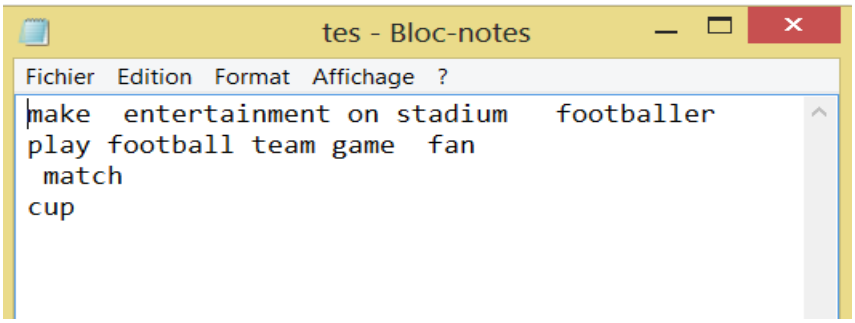
Fin si .

Fin.

Sortie : PS_N

Figure III.10: algorithme de produit scalaire enrichi.

Pour bien comprendre l'algorithme précédent, nous allons utiliser l'exemple de la **Figure III.2** afin de pouvoir catégoriser le document D' présenté dans la figure **III.11.**



La figure III.11 le document non classé.

Après les étapes de représentation, le document D' sera représenté par le tableau suivant:

make	enteraiment	stadium	footballer	play	football	team	game	fan	match	cup

1. Classification via des mesures de similarité statistiques :

Dans ce cas, il s'agit de classer des documents selon la présence /absence de mots. Dans l'exemple précédent, on applique la mesure du produit scalaire, les résultats obtenus sont présentés dans la figure **III.12**. Puisque les documents d'apprentissage et le document D' ne partagent aucun mot commun, le PS aura la valeur 0.

```
indmax 0
ps [0] = 0.0
ps [1] = 0.0
ps [2] = 0.0
ps [3] = 0.0
ps [4] = 0.0

  csVote[ 0 ]= 0
  csVote[ 1 ]= 0
```

Figure III.12 : résultat de classification statistique.

2. Classification via des mesures de similarité sémantiques :

Dans ce cas, le document D'est classé en utilisant les mesures de similarité sémantiques. On reprend le même exemple précédent et on le classé avec la mesure sémantique « wu palmer» et la base de données lexicographique wordNet sachant que nous avons utilisé les deux méthodes de désambiguïsions qui sont :

- **« tous les sens »** cette méthode consiste a prendre tous les synsets proposé par le wordnet, et comparé les synsets de chaque terme de D'avec les synsets des

documents d'apprentissage. La **figure III.13** représente un extrait d'exécution qui fait le calcule de similarité entre le sens 4244644 et 9057648.

```
--------nouveau------------------------
ossfettttt 4244644
s1  nom  [Synset: [Offset: 4244644] [POS: noun] Words:
stadium, bowl, arena, sports_stadium -- (a large structure for
open-air sports or entertainments)]
s1 stadium
s1 a large structure for open-air sports or entertainments
   attribut   4244644
zzzz
x 9057648
Africas2   nom
[Synset: [Offset: 9057648] [POS: noun] Words: Africa -- (the
second largest continent; located south of Europe and bordered
to the west by the South Atlantic and to the east by the
Indian Ocean)]
s2 the second largest continent; located south of Europe and
bordered to the west by the South Atlantic and to the east by
the Indian Ocean
sont de meme type
stadium:1:Africa:1:n
distance :   0.5
Max valuuuu    0.5   j     0
------------------------------------------------------------
```

Figure III.13 : exemple de calcule la similarité avec la mesure de wu palmer.

Les résultats interprétés dans le **tableau III.5**, ont été calculé par la mesure Wu&Palmer.

	Africa 9057648	Algerai 8583593	Egypt 8769547	Goal 5904683	Name 6248892	Visit 1217529
Stadium 4244641	0.5	0.4210	0.4210	0.25	0.25	0.2222

	Africa 9057648	Algerai 8583593	Egypt 8769547	Goal 5904683	Name 6248892	Visit 1217529
Stadium 4244641	0.5	0.4210	0.4210	0.25	0.25	0.2222

Tableau III.5 : calcule des mesures de similarité

Une fois la mesure de similarité est calculée pour tous les concepts du document D', on doit calculer le produit scalaire enrichi avec tout les documents, la **figure III.14** illustre les valeurs obtenues sur l'exemple précédent avec « tous les sens » comme méthode de désambigüisation

```
clacluler le produit scalaire erichi:
ps [0] = 0.9125729052631579
ps [1] = 0.0
ps [2] = 0.9394332631578948
ps [3] = 6.10436245142837
ps [4] = 3.009804209090909
```

figure III.14 : les valeurs de produit scalaire totale enrichi avec la méthode « tous les sens ».

On reprend notre exemple et on applique l'algorithme de k-nn avec k=3. D'après la **figure III.15,** on remarque que parmi les 3 plus proches documents deux documents appartiennent à la classe « sport »et un document appartient à la classe « pays ». Par conséquent, le document sera assigné à la classe « sport » (la plus représentée dans les 3 plus proche documents).

```
avec le kpp choisi est 3
 ------------------- pour k = 0--------------
maximum de ps  est    6.10436245142837
  doc  3
class trouvé est sport
--------------------- fin k = 0 ---------------
--------------------- pour k = 1 ---------------

maximum de ps est 3.009804209090909
  doc  4
class trouvé est sport

--------------------- fin k = 1 ---------------
--------------------- pour k = 2 ---------------

maximum de ps est 0.9394332631578948
  doc  2
class trouvé est pays
--------------------- fin k = 2 ---------------

 csVote[ 0 ]= 1
 csVote[ 1 ]= 2

 bravo ,le document sera classé dans la catégorisation sport
```

Figure III.15 : Application d'algorithme knn sur l'**exemple III.1** avec la méthode tous les sens

- **Méthode un seul sens** : cette méthode permet de prendre un seul synset qui est le premier de chaque concept est le compare avec le premier synset de concept d'apprentissage le produit scalaire totale est illustré dans la figure suivante :

```
produit scalaire enrichi:
ps [0] = 2.655713436532510
ps [1] = 0.0
ps [2] = 0.0
ps [3] = 5.229990610638897
ps [4] = 2.018180209090909
```

Figure III.16 : les valeurs de produit scalaire totale enrichi avec la méthode « un seul sens ».

On appliquant le knn, on trouve les résultats dans la figure ci-dessous.

```
avec le kpp choisi est 3
 ------------------- pour k = 0--------------
maximum de ps  est    5.229990610638897
  doc  3
class trouvé est sport
--------------------- fin k = 0 ---------------
--------------------- pour k = 1 ---------------

maximum de ps est 2.655713436532510

  doc  1
class trouvé est pays

--------------------- fin k = 1 ---------------
--------------------- pour k = 2 ---------------

maximum de ps est 2.018180209090909
 doc  4
class trouvé est sport
--------------------- fin k = 2 ---------------

 csVote[ 0 ]= 1
 csVote[ 1 ]= 2

 bravo ,le document sera classé dans la catégorisation sport
```

Figure III.17 : Application d'algorithme knn sur l'**exemple III.1** avec la méthode un seul sens.

III.3 Description des outils et les technologies utilisées :

Cette application offre de multiples fonctionnalités qui exigent l'utilisation de différents outils pour les implémenter, certains de ces outils sont communs sur tous les projets java mais d'autres sont particuliers à ce type de projets.

III.4.1 Java :

Java est un langage de programmation orienté objet simple crée par Sun Microsystems en 1995. qui réduit les risques d'incohérence .il est rapide, sécurisé et fiable, il permet de crée des logiciels compatible avec des nombreux systèmes d'exploitations (Windows, Linux, Macintosh, Solaris), Il permet d'accéder d'une manière simple aux fichiers et aux réseaux. Beaucoup d'applications et de sites Web ne fonctionnent pas si Java n'est pas installé, et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable. Des ordinateurs portables aux centres de données, des consoles de jeux aux superordinateurs scientifiques, des téléphones portables à Internet, la technologie Java est présente sur tous les fronts.

En a choisi java avec sa version 8 comme langage de programmation du projet.

III.4.2 NetBeans :

L'environnement de développement utilisé, est NetBeans 8.0.2, 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, **…**

III.4.3 Bibliothèque Weka (Waikato environment for knowledge analysis):

Weka est une suite de logiciels d'apprentissage automatique écrite en Java et développée à l'université de Waikato en Nouvelle-Zélande. L'espace de travail Weka contient une collection d'outils de visualisation et d'algorithmes pour l'analyse des

données et la modélisation prédictive, allié à une interface graphique. Pour un accès facile de ses fonctionnalités. Les principaux points forts de Weka sont qu'il :

- est libre et gratuit, distribué selon les termes de la licence publique générale GNU ;
- est portable car il est entièrement implémenté en Java et donc fonctionne sur quasiment toutes les plateformes modernes, et en particulier sur quasiment tous les systèmes d'exploitation actuels ;
- contient une collection complète de préprocesseurs de données et de techniques de modélisation ;
- est facile à utiliser par un novice en raison de l'interface graphique qu'il contient.
- Weka supporte plusieurs outils d'exploration de données standards, et en particulier, des préprocesseurs de données, des agrégateurs de données (data clustering), des classificateurs statistiques, des analyseurs de régression, des outils de visualisation, et des outils d'analyse discriminante.

III.4.4 WordNet :

Afin de réaliser notre travail, nous avons utilisé WordNet de version 2.0 qui est une base de données lexicographique pour les raisons suivantes :

- 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. Déversions de ce dernier existent 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 en nombre de mots, nombre de synsets et nombre de sens, globalement et par catégorie grammaticale.

Position	Mots	Synset	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 III.6 : Caractéristiques du nombre de mots et de concepts dans WordNet

III.4.5 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.

III.4 conclusion :

Dans ce chapitre nous avons présenté le déroulement des phases les plus importants de la classification de texte tel que la représentation conceptuelle et le classement de document.

A propos de classement, nous avons classé les documents sémantiquement et statistiquement et on a interprété les résultats obtenus, puis nous avons cité les outils nécessaires pour implémenter notre approche. Finalement nous avons présenté les interfaces graphiques de notre application.

Conclusion générale

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

Nous nous sommes intéressés dans cette thèse à développer un travail dans le cadre d'enrichir et d'améliorer la représentation conceptuelle des documents dans la catégorisation des textes, via l'utilisation des mesures de similarité sémantique. Cette amélioration consiste à calculer et évaluer la proximité sémantique entre les concepts afin d'améliorer les performances de classification automatique.

Avec l'utilisation de la base lexicographique Wordnet 2.0 qui fournit un ensemble des sens, nous avons effectué le Maping des mots en concepts. Sachant que ce mapping introduira un problème d'ambiguïté, nous avons utilisé deux méthodes de désambiguïsions. La première consiste à prendre en considération tous les sens associés au mot, par contre la deuxième méthode consiste à ne prendre que le premier sens (le sens le plus répondu).

A l'issue de ce travail, de nombreuses pistes restent à explorer :

- Evaluer notre travail, avec des corpus standards comme Reuters10 .
- Enrichir la représentation conceptuelle des documents via les relations sémantiques (hyponymie, meronymie… ect),
- Prendre en considération d'autres méthodes de désambiguïsation afin de localiser le sens adéquat du mot dans le document.
- Elargir notre implémentation en utilisant la dernière version de Wordnet (WordNet3.0) afin de pouvoir évaluer l'influence de la richesse de la ressource sémantique sur les performances de la classification.

Références bibliographiques

[1] : Dictionnaire de l'Académie française, 8e édition ,1992

[2] : Sebastianni, F. *«Machine learning in automated text categorization*». ACM Computing Surveys, pages 02:16, 2002.

[3] : JALAM, R *« Apprentissage automatique et catégorisation de textes multilingues* » Université Lumière Lyon2, Juin 2003.

[4] MATALLAH, H. « *Classification Automatique de Textes Approche Orientée Agent* »,Thèse de Magister, Université ABOUBEKR BELKAID-TlEMCEN,2011 pages 13-35.

[5] :Bentaallah, M. « *Utilisation des Ontologies dans la Catégorisation de Textes Multilingues* »,Thèse de doctorat, Université DjillaliLiabes de Sidi Bel Abbes,2011.pages 21-50

[6] : RIFQI, M. «*Mesures de similarité, raisonnement et modélisation de l'utilisateur*», Article de recherche, Laboratoire d'Informatique de Paris

[7] : Slimani, T., Ben Yaghlane, B., Mellouli, k. *« Une extension de mesure de similarité entre les concepts d'une ontologie* » .2016 .p03

[8] Rada, R., Mili, H., Bichnell, E., al. *«Development and application of ametric on semantic nets*». IEEE Transaction on Systems, 1989.

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

[10] Resnik , P. *«* Using information content to evaluate semantic similarity in taxonomy ». In Proceedings of 14 th International Joint Conference on Artificial Intelligence , Montreal, 1995.

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

[12] Miller, G., Beckwith, R., Fellbaum, C., al. *« Introduction to WordNet: An On-line Lexical Database*» . Cognitive Science Laboratory, Princeton University, Princeton, Technical Report 1993.

[13] Hirst, G., St Onge, D. *« 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.

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

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

Résumé :

Dans notre projet de master, nous avons traité une problématique liée au domaine de catégorisation du texte qui consiste à associer chaque document non classé à sa catégorie en utilisant un ensemble des documents préalablement classés.

Notre but est d'évaluer l'utilisation des mesures de similarités sémantiques et leurs impacts sur la classification automatique de textes à l'aide de la base de donnée lexical Wordnet.

L'implémentation de notre projet est faite à l'aide de langage java en utilisent la bibliothèque Weka.

Mots-clés : wordnet, java, Weka, classification du texte, mesures de similarités sémantiques**.**

Abstract :

In this project, we dealt with a problem related to the text categorization domain, which involves associating each non-classified document with its category using a Set of previously classified documents.

Our goal is to evaluate the use of semantic similarity measures and their impact on automatic classification of texts using the Wordnet lexical database.

The implementation of our project is done using java language and Weka library.

Keywords: wordnet, java, Weka, text classification, semantic similarity measures.

ملخص:

في مشروعنا الرئيسي ، تعاملنا مع مشكلة تتعلق بمجال تصنيف النص ، والذي يتضمن ربط كل وثيقة غير مصنفة باستخدام مجموعة من الوثائق المصنفة سابقًا بفئتها.

هدفنا هو تقييم استخدام مقاييس التشابه الدلالي وتأثيرها على التصنيف التلقائي للنصوص باستخدام قاعدة بيانات المعجمية Wordnet.

يتم تنفيذ مشروعنا باستخدام لغة جافا باستخدام مكتبةWeka .

الكلمات المفتاحية: تصنيف النص ، مقاييس التشابه الدلالي، Weka، java،