Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

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 : Modèle Intelligent et Décision (M.I.D)

Evaluation de l'utilisation des mesures de similarité sémantiques dans le cadre de la catégorisation de textes

Enrichissement de la représentation conceptuelle

Réalisé par:

- BETAOUAF Mohammed Abd El-Aziz
- BENAISSA Boumediene

Présenté le 24 Septembre 2020 devant le jury composé de :

- M.BENAISSA Mohammed (Président)
- M.BENTAALLAH Mohamed Amine (Encadreur)
- M.BELABED Amine (Examinateur)

2019-2020

Dédicace

“

Je dédie ce mémoire :

À ma mère pour tout l'amour dont vous m'avez entouré, pour vos attachements, j'apporte à vous beaucoup d'affection et de reconnaissance.

Je ferai de mon mieux pour rester un sujet de fierté à vos yeux avec l'espoir de ne jamais vous décevoir.

À mes très chers sœurs et frères que je vénère comme des talismans.

À la mémoire de mon père qui est parti vers le monde de l'au-delà avant que je le voie de mes propres yeux. À toute ma famille : oncles, tantes, cousins et cousines paternels et maternels.

À tous mes amis qui me sont chers, à tous ceux que j'aime et qui m'aiment : qu'ils trouvent ici l'expression de mes sentiments les plus dévoués et mes vœux les plus sincères. Que dieu vous préserve tous et, vous procure sagesse et bonheur.

”

- Boumediene

DEDICACE

À la lumière de mes yeux et le bonheur de mon existence les plus chères et les plus idéaux hommes et femmes dans ma vie ***« Mon père et Ma mère »*** *pour l'amour qu'ils m'ont porté et pour leur soutient et conseils, m'ont donné confiance, courage et sécurité. Qu'ils trouvent ici le témoignage de ma grande affection et amour.*

À mes chères sœurs en témoignage de l'affection qui nous unit, je leur souhaite la réussite dans la vie et Beaucoup de bonheur.

À toute ma famille ***BETAOUAF*** *et* ***SEKRAN*** *sans exception : oncles, tantes, cousins et cousines.*

À tous mes amis qui m'ont toujours encouragé et à qui je souhaite plus de succès

À tous les étudiants de la promotion de ***Master informatique***

À toute personne qui a participé de près ou de loin dans la réalisation de ce travail et dont je n'ai pas mentionné les noms à travers ces lignes ; je vous dis tous ***MERCI***

MOHAMMED ABD EL-AZIZ.

Remerciements

Tout d'abord, nous remercions Allah, notre créateur de nous avoir donné la force, la volonté et le courage d'accomplir ce modeste travail.

Nous adressons les plus grands remerciements à notre encadreur **M. BENTALLAH**, pour ses conseils et le temps qu'il nous a consacré pour arriver à ce résultat.

Nous tenons également à remercier messieurs les membres du jury pour l'honneur qu'ils nous ont fait en acceptant de juger ce modeste travail.

Finalement, nous tenons à exprimer notre profonde gratitude à nos familles qui nous ont toujours soutenues et à tout ce qui a participé à la réalisation de ce travail. Ainsi que l'ensemble des enseignants du département d'informatique de l'université Abou bekr Belkaid pour l'effort fourni lors de notre formation.

Table des matières

Introduction générale **10**

1 Catégorisation des textes **11**

1.1 Introduction 12

1.2 Pourquoi automatiser la classification 12

1.3 Définitions 13

1.4 Processus de la catégorisation des textes 13

1.4.1 La représentation des textes 15

1.4.2 La pondération des termes 19

1.4.3 La réduction de la taille de vocabulaire 20

1.4.4 Les classificateurs 22

1.5 Évaluation du processus de catégorisation 29

1.6 Domaine d'applications de la catégorisation des textes 30

1.7 Conclusion 30

2 Les mesures de similarités syntaxiques et sémantiques **31**

2.1 Introduction 32

2.2 Similarité syntaxique 32

2.2.1 Similarité Cosinus 32

2.2.2 Distance euclidienne 33

2.2.3 Indice de Dice 33
2.2.4 Coefficient de Jaccard 34
2.3 Similarité sémantique 34
2.3.1 Approches basées sur les arcs 34
2.3.2 Approches basées sur les nœuds 38
2.4 Conclusions 39

3 Enrichissement de la représentation conceptuelle 40
3.1 Introduction 41
3.2 Architecture générale 41
3.2.1 Phase de représentation 44
3.2.2 La phase d'enrichissement 53
3.2.3 Application pratique sur les trois approches 58
3.2.4 La phase de classification 61
3.3 Déroulement de notre programme 62
3.3.1 La représentation 62
3.3.2 L'enrichissement 64
3.3.3 Résultat 68
3.4 Les ressources utilisées 69
3.4.1 Description de corpus utilisé 69
3.4.2 Environnement de travail : 70
3.4.3 Python 70
3.4.4 Bibliothèque pywsd 70
3.4.5 WordNet 71
3.4.6 QT Designer 71
3.4.7 Scikit learn 71

3.5 Expérimentations 71
3.6 Conclusions 72

Conclusion générale **73**

Références **74**

Table des figures

1.1 Processus de la catégorisation des textes [1]. 14
1.2 Exemple de la représentation en sac de mot 15
1.3 Exemple de la représentation par phrases 16
1.4 Exemple de N-grammes de mots et de caractères. 18
1.5 La représentation conceptuelle du mot « Vivre ». 19
1.6 Sélection de termes 21
1.7 Extraction de termes 22
1.8 Exemple de SVM 23
1.9 Exemple des k-plus proches voisins 24
1.10 Exemple d'arbre de décision 26
1.11 Schéma d'un neurone 27
1.12 Schéma d'un neurone 28

2.1 Exemple de la Distance euclidienne. 33
2.2 Exemple de Wu et Palmer. 36

3.1 Architecture générale. 43
3.2 Exemple d'un document. 44
3.3 Application de la Tokenisation sur notre document. 44
3.4 Document sans majuscule. 45

3.5 Liste des mots vides. 47
3.6 L'étape d'élimination des mots vides. 47
3.7 Catégories grammaticales pour la langue anglaise. 48
3.8 Les dix concepts du mot « cat ». 49
3.9 Exemple d'un deuxième document. 50
3.10 le vecteur des couples (Terme,Concept). 50
3.11 vecteur conceptuel. 51
3.12 La matrice (document, concept) et le vecteur conceptuel de document à catégoriser. 52
3.13 Algorithme de l'approche sans enrichissement. 53
3.14 Algorithme de l'approche d'enrichissement globale. 55
3.15 Algorithme de l'approche d'enrichissement local. 57
3.16 le corpus et le document à catégoriser. 58
3.17 la matrice et le vecteur conceptuelle. 59
3.18 Résultat de l'approche sans enrichissent. 60
3.19 Résultat de l'approche d'enrichissement globale. 60
3.20 Résultat de l'approche d'enrichissement locale. 61
3.21 Le chargement de corpus et le document à classer. 62
3.22 Le choix de l'approche. 64
3.23 L'interface de l'approche sans enrichissement. 65
3.24 L'interface de l'approche d'enrichissement (partie 1). 67
3.25 L'interface de l'approche d'enrichissement (partie 2). 68
3.26 Exemple d'un résultat. 69

Introduction générale

À notre époque, la catégorisation des textes est devenu une phase très importante pour la navigation en raison des bases de données énorme de sorte qu'il est difficile pour le navigateur de connaître les informations importantes dans sa recherche parmi toutes ces information, mais de nouvelles technologies ont été développées dans ce domaine pour faciliter le processus de la catégorisation des documents.

L'augmentation du volume d'informations textuel sous forme numérique nécessite une représentation efficace pour les organises, d'après plusieurs chercheurs qui ont été préférer d'utiliser la représentation conceptuelle qui se base sur le web sémantique. Cette représentation a eu comme inconvénient l'absence du distance sémantique entre les documents pour cela nous avons propose deux approches d'enrichissements.

Notre mémoire est composé de trois chapitres, dans le premier chapitre nous avons expliqué le processus de la catégorisation des textes avec ces traitements principaux. Dans le deuxième chapitre nous avons abordé les différentes approches de mesure de similarité sémantique avec une comparaison entre eux et aussi quelques mesures syntaxiques. Le dernier chapitre concernant à exposer l'architecture générale et les deux approches proposées, ainsi que le déroulement de notre programme et on termine par les expérimentations de notre travail.

Chapitre 1

Catégorisation des textes

1.1 Introduction

Vu la forte croissance du nombre de documents numériques accessible sur le web, les utilisateurs trouvent une difficulté pour localiser le contenu qui les intéresse. Pour cela les recherches se sont focalisées dans la découverte de moyens permettant d'automatiser la catégorisation de documents dans cette grande collection documentaire.

Dans ce chapitre, nous abordons en premier lieu les besoins qui se cachent derrière l'automatisation de la catégorisation tout en listant quelques définitions de la catégorisation de textes (C.T). Nous détaillons par la suite, les étapes du processus de la C.T à savoir la représentation des textes, la pondération des termes, la réduction de la taille du vocabulaire et les classificateurs. Nous exposons aussi les mesures utilisées pour l'évaluation ainsi que les applications de la C.T et en dernier lieu les principales difficultés liées à la C.T.

1.2 Pourquoi automatiser la classification

Aujourd'hui, la quantité d'information est faramineuse au point que les serveurs, pages et les corpus utilisés ont atteint leur point culminant. Reuters a recensé ses nouvelles et articles à catégoriser dépassant les 800000 et 5,5 millions respectivement. De plus elle employait maintes personnes pour l'étiquetage de ses documents. La question suivante se pose : combien de temps a besoin un humain pour associer des textes à une catégorie ? Plusieurs variables influencent ce phénomène à savoir le temps de lecture et relecture, la vitesse de lecture et la longueur du texte. De plus, le temps nécessaire pour consulter la description des classes.

La réalisation manuelle de la classification est sujette à ces contraintes majeures :

– La réalisation manuelle de cette tâche par un expert est coûteuse en matière de temps et personnel.[2] [3]

–Les traitements manuels sont peu flexibles et leur généralisation à d'autres

domaines est quasi-impossible.[2] [3]

–L'opération est subjective : deux experts peuvent classer différemment un même document. [4] [5]

Ainsi, plusieurs travaux de recherche se concentrent sur la perspective de l'automatisation de classification de textes.

1.3 Définitions

Dans la littérature, plusieurs définitions de la C.T ont été proposées, SEBASTIANI définit dans [2] la catégorisation de textes comme étant le processus qui consiste à associer une valeur booléenne à chaque paire $(d_j, c_i) \in D$ x C, ouD est l'ensemble des textes et C 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) lui sera associée dans le cas contraire.

Le but de la catégorisation de texte est de construire une procédure (modèle, classifieur)$\phi : D$ x $C \rightarrow \{V, F\}$ qui associe une ou plusieurs étiquettes (catégories) à un document dj telle que la décision donnée par cette procédure « coïncide le plus possible » avec la fonction $\phi :' D$ x $C \rightarrow \{V, F\}$, la vraie fonction qui retourne pour chaque vecteur d_j une valeur c_i. Selon R.JALAM, 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 (traduction de machine Learning méthode). [1]

1.4 Processus de la catégorisation des textes

Comme montré dans (Figure 1.1), le processus de catégorisation de textes se compose essentiellement de deux phases à savoir l'apprentissage et le classement.

La phase d'apprentissage consiste à utiliser les algorithmes d'apprentissage automatique afin de construire un modèle de prédiction à partir d'un ensemble de textes étiquetés.

La phase du classement d'un nouveau texte d_x se focalise ensuite sur l'application du modèle construit sur ce dernier afin de prédire sa catégorie .[1]

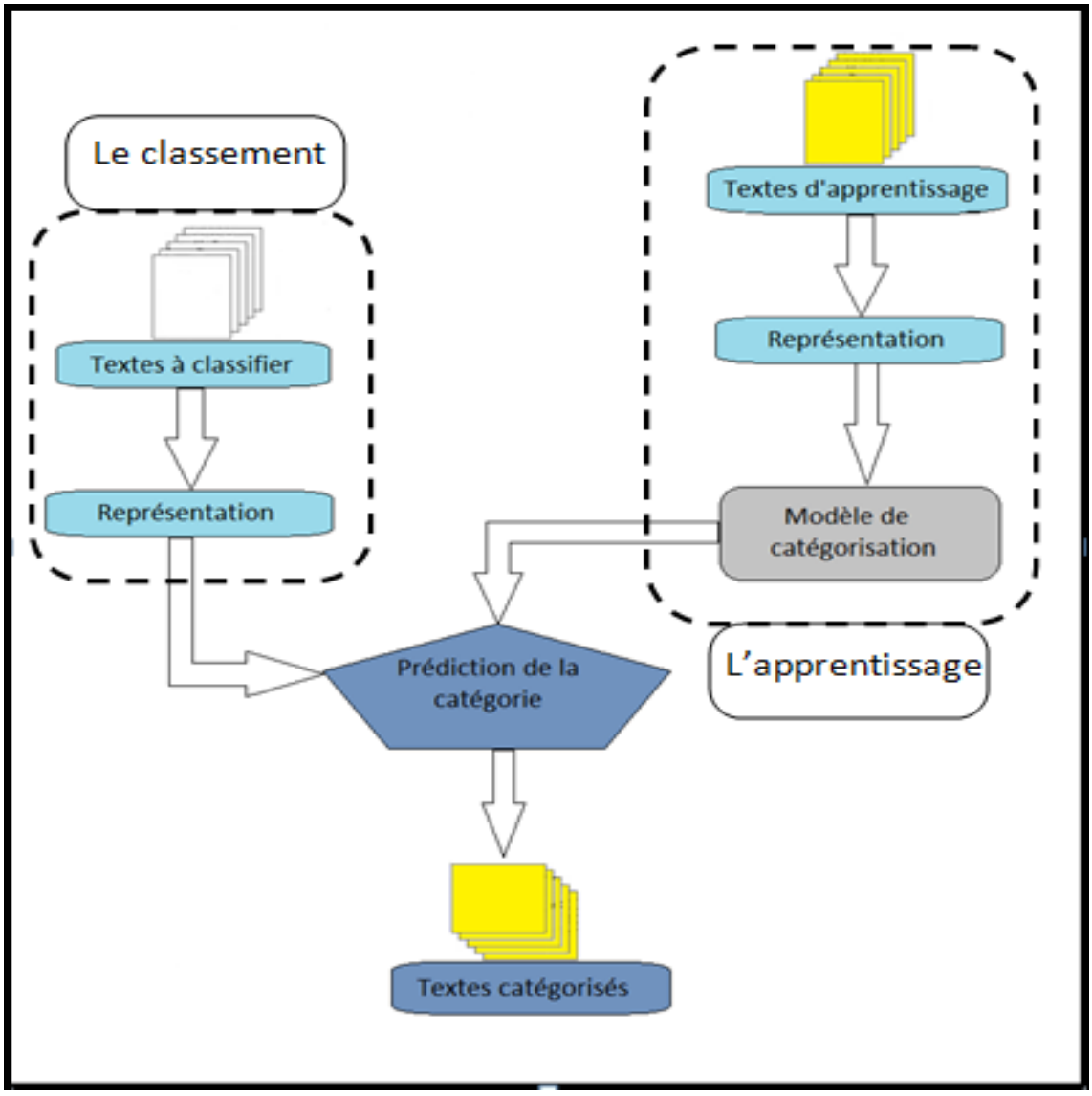

FIGURE 1.1: Processus de la catégorisation des textes [1].

1.4.1 La représentation des textes

La représentation de texte consiste à représenter les documents textuels sous une forme exploitable par la machine. Plusieurs méthodes ont été proposées. Elles se distinguent selon la nature du descripteur utilisé (mot, phrase, lemme, ...etc.).Le choix de la méthode utilisée est primordial. En effet, la qualité de la représentation influence directement sur les performances du catégorisation, ces méthodes sont :

Représentation en sac de mot

Cette méthode se base sur l'utilisation de « mots » comme unité élémentaire pour représenter les textes. Un document est ainsi représenté sous forme d'un ensemble de mots.

Cette méthode présente l'avantage d'être simple dans son implémentation, Néanmoins les délimiteurs diffèrent d'une langue à une autre. De plus, certains délimiteurs peuvent faire partie des unités comme les adresse IP (172.16.29.1) , les dates(15/03/2021) ou les mots composé (pomme de terre) ,d'un autre côté elle exclut toute analyse grammaticale et toute notion de distance entre les mots ce qui conduit à une perte dans la sémantique du texte.

Nous présentons ci-dessous dans (Figure 1.2) un exemple de la représentation en sac de mot.

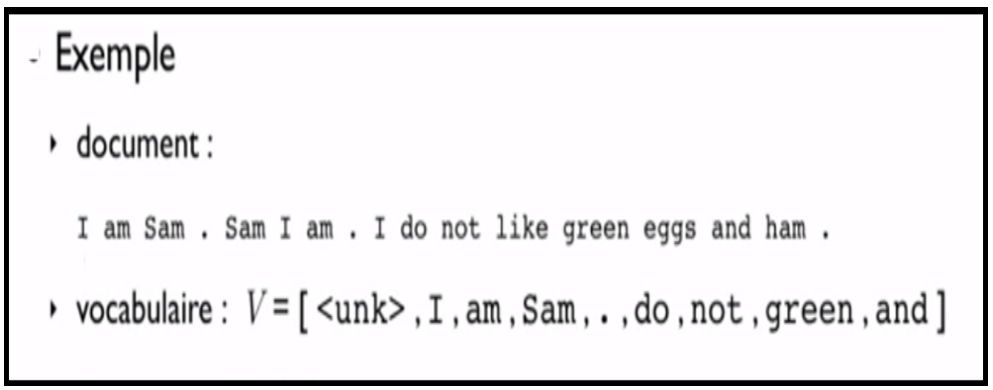

FIGURE 1.2: Exemple de la représentation en sac de mot

Représentation par phrases

A la différence de la représentation « sac de mots », Cette technique utilise les phrases comme unité de représentation au lieu des mots, plusieurs chercheurs ont préféré utiliser les phrases en raison de leur richesse sémantique.

Cette méthode présente l'avantage de conserver l'information relative à la position du mot dans la phrase («dark web», «génie logiciel»). Néanmoins, les qualités statistiques ne sont pas fiables car le grand nombre de combinaisons possibles entraîne des fréquences faibles et trop aléatoire [6].

Nous présentons ci-dessous dans (Figure 1.3) un exemple de la représentation par phrases.

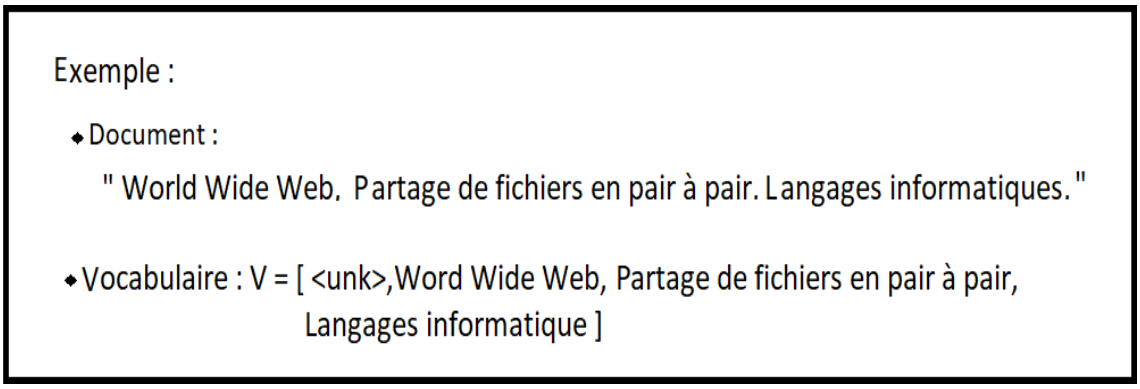

FIGURE 1.3: Exemple de la représentation par phrases

Représentation avec racines lexicales

Cette représentation consiste à convertir le mot en racinece qui conduit au regroupement de mots qui partagent la même racine en une seule forme canonique, par exemple il est nécessaire de faire une liaison entre « petit » « petite », « petites ».

Il existe des nombreux algorithmes pour réalises cette représentation, le plus célèbre d'entre eux est l'algorithme de Porter [7] qui se base sur les techniques de désuffixation. L'inconvénient majeur réside dans la possibilité de générer la même racine pour des mots ayant des significations, par exemple : «

university » et « universe » ont la même racine «univers» mais ils ne partagent pas la même signification.

Représentation avec lemme

Cette représentation consiste à remplacer les verbes par leur forme infinitive, et les noms par leur forme au singulier. Pour cela, la lemmatisation nécessite une analyse grammaticale des textes afin de pouvoir détecter la catégorie grammaticale des mots (verbe, nom, adjectif ou adverbe) que nous pouvons réaliser par un algorithme efficace TreeTagger1. [8]

Néanmoins L'ambiguïté peut être causée aussi par le simple remplacement de la forme plurielle d'un mot par sa forme singulière par exemple le mot « actions » qui est représenté par le descripteur « action ». Dans un contexte économique, le mot « actions » se réfère couramment à des actions d'entreprises et n'a rien à voir avec la notion « action » employée par exemple dans la phrase : « Le plan d'action de l'état ».[8]

Représentation avec les N-grammes

Cette méthode consiste à représenter le document sous forme d'un ensemble de sous-séquences de n caractères consécutifs appelées « N-grammes ». Elle consiste a découpé le texte en plusieurs séquence de n caractère en se déplaçant avec une fenêtre de N caractères ce déplacement, s'effectue caractère par caractère, à chaque déplacement la séquence de n caractères est enregistrée, l'ensemble de ces séquences constitue l'ensemble des n-grammes représentant le texte [9]. Quelque auteur comme [10] considèrent un n-grammes comme une suite de n mot. 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 [11]. (Figure 1.4) illustre un exemple de découpage selon les deux types de n-grammes, caractères et mots.

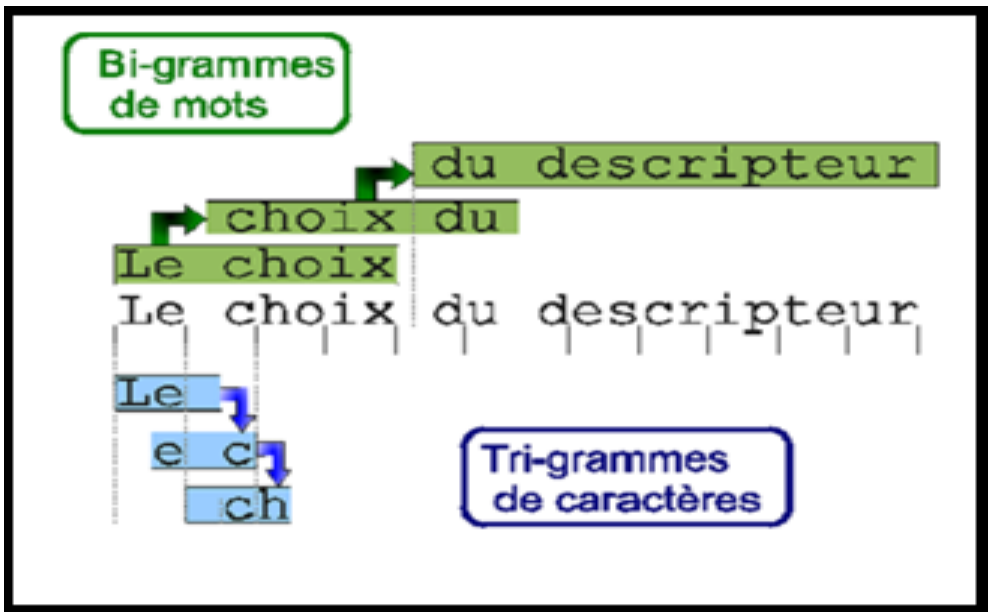

FIGURE 1.4: Exemple de N-grammes de mots et de caractères.

Représentation basée sur les concepts

Cette technique est complètement différente car elle n'utilise pas la comparaison morphologique comme les autres techniques.

On peut la définir comme une représentation des documents sous forme d'un ensemble de concepts via l'utilisation des réseaux sémantiques ou les sous arbres pour capturer les concepts.

Cette méthode a comme avantage selon REHEL dans [12] de réduire l'espace de travail car les mots qui sont synonymes partagerons le même concept. Cependant, l'inconvénient majeur de cette représentation est qu'il n'existe pas des bases lexicales pour toutes les langues.

Selon l'exemple de (Figure 1.5), on peut regrouper les quatre termes (exister, subsister, habiter, expérimenter) du vecteur dans le concept vivre.

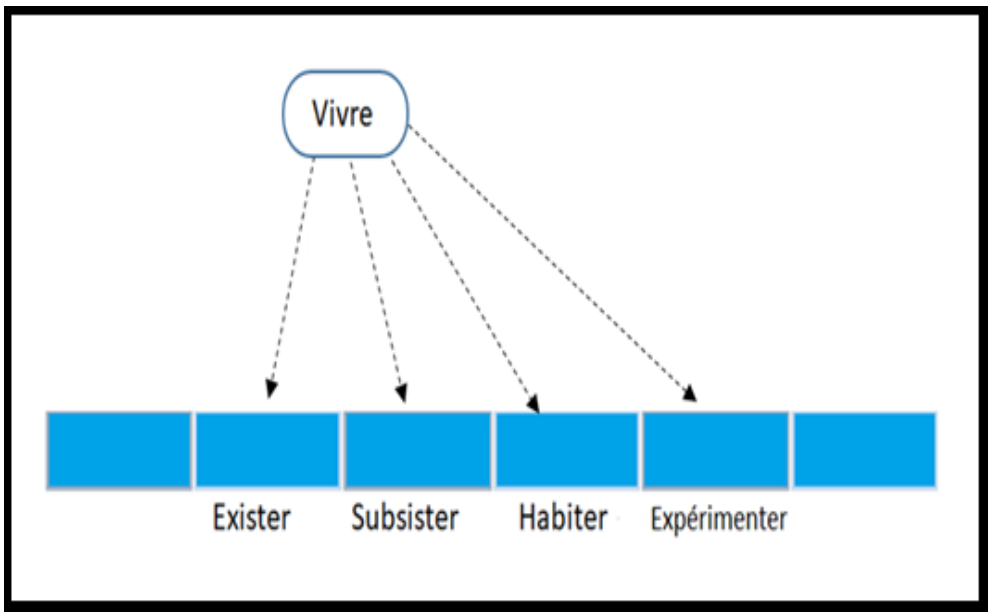

FIGURE 1.5: La représentation conceptuelle du mot « Vivre ».

1.4.2 La pondération des termes

La pondération est utilisée dans le calcul de poids des termes, Ce poids doit modéliser l'importance d'un terme dans un document mais aussi au sein du document et de la collection, parmi les méthodes les plus utilisées :

La mesure TF

TF(term frequency)détermine la fréquence relative d'un terme dans un document. Cette fréquence du terme sera comparée à la survenance de tous les autres termes restants du document [13]. Selon la pondération TF, plus le terme est récurrent dans le document plus il est important. L'inconvénient majeur de cette pondération réside dans le fait qu'elle mesure l'importance du terme dans le document sans prendre en considération sa présence dans les autres documents.

La mesure IDF

A l'inverse de la méthode TF, L'IDF(inverse document frequency)[13] évalue l'importance d'un terme globalement en se basant sur le nombre de documents contenant ce dernier. Ainsi, plus le terme apparait dans plusieurs documents moins il est important. Formellement, l'IDF se définie par :

$$\text{idf} = \log\left({}^{N}/_{Df}\right)$$

Avec :

Df : Le nombre de documents contenant le terme.

N : Le nombre total de documents de la base documentaire

La mesure TF*IDF

Cette mesure combine les deux mesures précédentes (TF, IDF) dans le but de pondérer le terme localement et globalement. C'est une bonne approximation de l'importance du terme dans le document ; Ainsi, pour qu'un terme soit important dans un document, il doit se répéter fréquemment dans ce document et rarement dans les autres documents. Néanmoins, cette mesure ne prend pas en considération la longueur du document. La mesure TFC corrige ce problème via une normalisation selon la longueur du document :

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

1.4.3 La réduction de la taille de vocabulaire

Dans le cadre de la catégorisation de texte, la phase de réduction du vocabulaire est importante parce que elle traite un problème central c'est la

grande dimension de l'espace de représentation. Ce problème influence sur l'algorithme d'apprentissage par le coût du traitement car le nombre des termes intervient dans l'expression de la complexité de l'algorithme, plus ce nombre est élevé, plus le volume de calcul est important, ainsi que la faible fréquence de certains termes : on ne peut pas construire des règles fiables à partir de quelques occurrences dans l'ensemble d'apprentissage. Parmi les solutions proposées, nous citons :

Sélection de termes

Cette phase consiste à réduire la taille de l'espace d'apprentissage par diminuer le nombre des termes pour construire un sous-ensemble de taille moins que le premier ensemble , Parmi ces techniques figurent le calcul de l'information mutuelle [6] et [14], la méthode du 2 max [15], ou des méthodes plus simples utilisant uniquement les fréquences d'apparitions [16] .

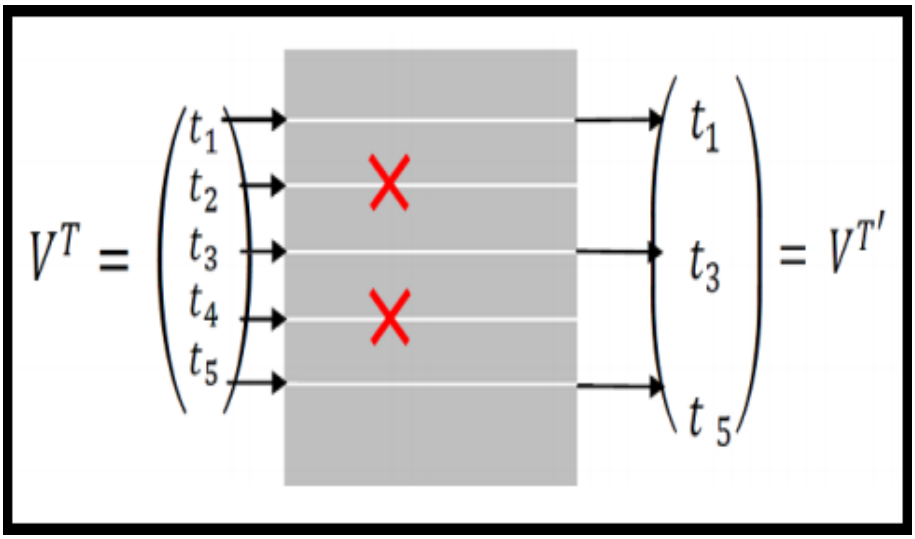

FIGURE 1.6: Sélection de termes

Considérons (Figure 1.6) sélection des attributs où on dispose d'un ensemble d'attributs $V^T = \{t1, t2, t3, t4, t5\}$ réduit en $|T'|$ 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 de termes

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. [2]

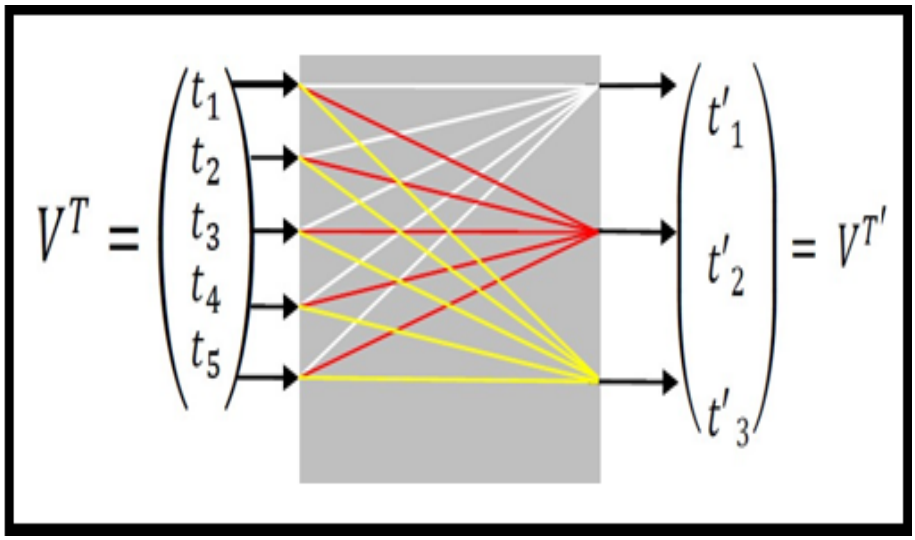

FIGURE 1.7: Extraction de termes

1.4.4 Les classificateurs

Les classificateurs ou les algorithmes d'apprentissage influencent directement sur les résultats de la CT, pour cela il faut bien choisir la technique d'apprentissage qui s'adapte le plus afin d'obtenir des bons résultats, parmi les méthodes d'apprentissage : l'analyse factorielle discriminante, la régression logistique, les réseaux de neurones, les réseaux bayésiens, les plus proches voisins, les arbres de décision, les machines à vecteurs supports et les méthodes de boosting.

Les machines à vecteurs de support

Les machines à vecteurs de support (SVM) sont une généralisation des classifieurs linéaires. Le principe consiste à rechercher l'hyperplan permettant de maximiser la marge entre les classes garantissant ainsi une meilleure séparation entre les classes, La marge est la distance entre la frontière de séparation et les échantillons les plus proches.

Comme montré dans (Figure 1.8), l'hyperplan sépare les deux ensembles d'échantillons des deux classes. Les points les plus proches sont utilisés pour la détermination d'hyperplan, et sont appelés vecteurs de support.

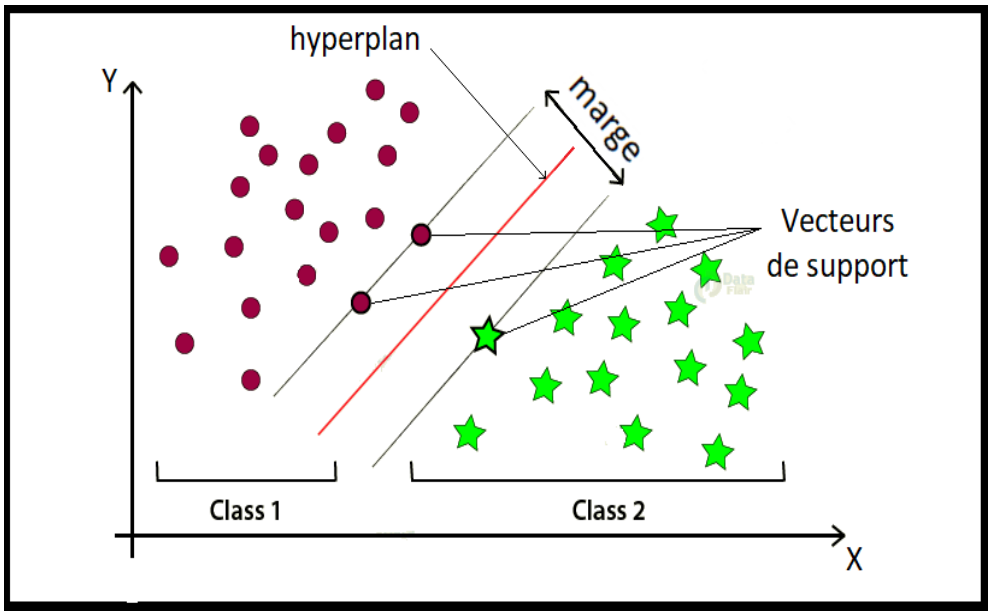

FIGURE 1.8: Exemple de SVM

Les k plus proches voisins

Faisant partie des algorithmes d'apprentissage supervisée, Les KPPV associent le texte à la catégorie la plus représentée dans l'ensemble des K textes les plus proches. Ainsi, il est nécessaire de calculer d'abord le degré de similarité du texte à catégoriser avec tous les textes de la base d'apprentissage et recherche les k textes d'apprentissage les plus similaires. Finalement, le texte

sera affecté à la catégorie ayant le plus grand nombre de textes parmi les k textes les plus proches (Figure 1.9).

Le mérite de cet algorithme est qu'il est simple. Cependant, le problème est qu'il doit stocker tous les échantillons dans l'ordinateur et comparer la distance entre l'échantillon en cours d'identification et tous les autres échantillons d'apprentissage.

Le choix du paramètre K est primordial pour le bon fonctionnement de cette méthode. [17]

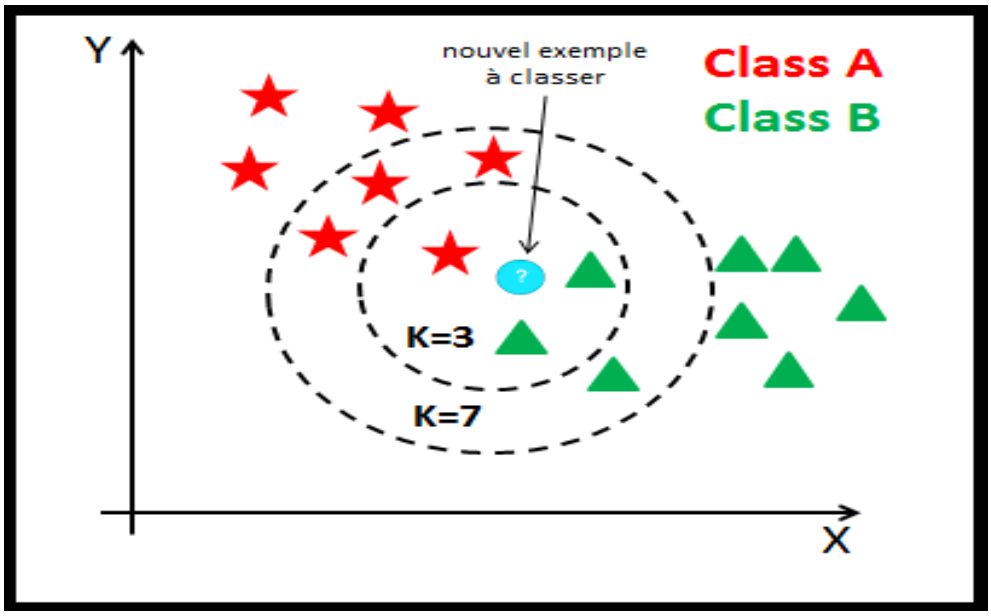

FIGURE 1.9: Exemple des k-plus proches voisins

Méthode de Rocchio

Initialement proposée pour la recherche d'information, Cette méthode a comme principe la construction d'un profil prototypique (pour chaque classe) permettant la discrimination de la classe par rapport aux autres classes. 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. [18]

$$w_{ki} = \alpha . \sum_{t_{j \epsilon POS_i}} \frac{w_{kj}}{|POS_i|} - \beta . \sum_{t_{j \epsilon NEG_i}} \frac{w_{ki}}{|NEG_i|}$$

Avec POS_i l'ensemble des documents de Tr appartenant à la catégorie C_i et POS_i l'ensemble des documents de Tr n'appartenant pas à la catégorie Ci. Les valeurs réelles et sont fixées arbitrairement. En général $\alpha > \beta$.

La méthode est efficace pour des catégorisations multi-classes ou un texte ne peut appartenir qu'à une seule catégorie. Mais elle n'est pas très efficace quand un texte peut appartenir à plusieurs.

Naïve bayes

Cette méthode se base sur le calcul de la probabilité d'appartenance d'un document x_i à une classe C_j en utilisant l'équation suivante :

$$P(c_j|x_i) = \frac{P(c_j)P(x_i|c_j)}{P(x_i)}$$

Le document est associé à la classe maximisant 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

L'avantage de cette méthode réside dans sa simplicité. En effet, elle se base sur un simple calcul de co-occurrences sans aucune pondération de descripteurs.

Les arbres de décisions

Les arbres de décision sont des méthodes supervisées dont le principe consiste à effectuer une décomposition hiérarchique de la base d'apprentis-

sage en se basant sur la présence/absence des termes. L'arbre de décision est une structure similaire à un organigramme, chaque branche représente le résultat d'un test et chaque feuille ou nœud terminal contient une étiquette de classe.

La construction de l'arbre se base sur le partitionnement récursif des textes de la base d'apprentissage. A chaque itération, il s'agit de choisir le meilleur partitionnement parmi les partitionnements possibles. Chaque sous ensemble de la partition représente un nœud et sera par la suite partitionné. 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. Il existe plusieurs variantes des arbres de décision qui se différencient essentiellement par le critère de partitionnement.

(Figure 1.10) représente un arbre de décision d'une maladie, les nœuds sont : douleur, fièvre, toux. La classe maladie contient les étiquettes : rhume, mal de gorge, refroidissement, rien.

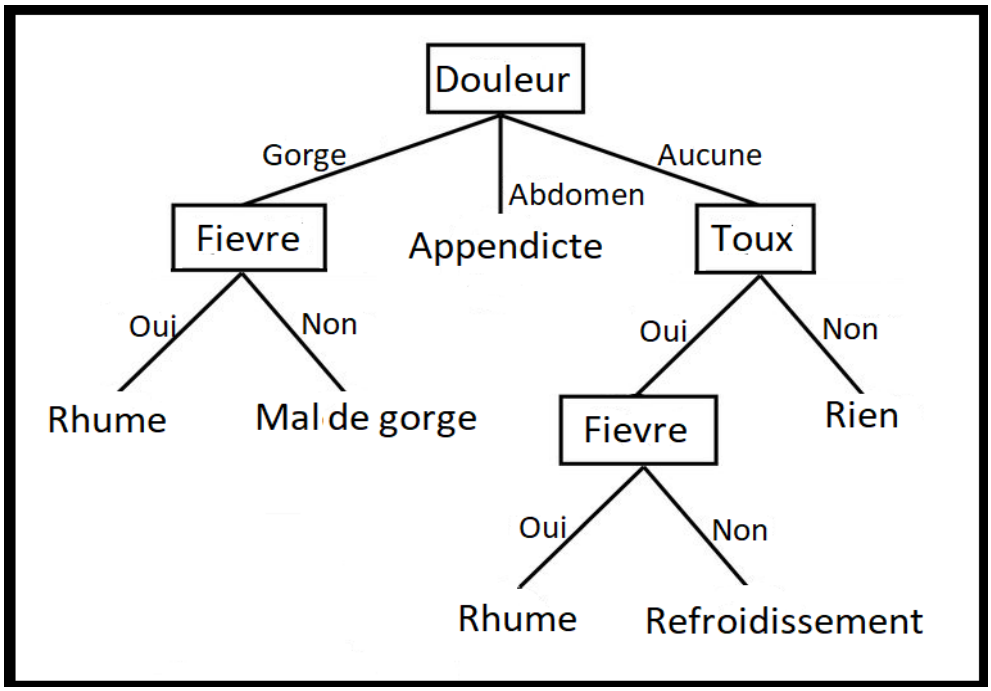

FIGURE 1.10: Exemple d'arbre de décision

Les réseaux de neurone

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau.[19]

Si nous voulons comprendre le fonctionnement de Les réseaux de neurones artificiels nous devons revenir à la biologie humaine plus spécifique le cerveau, a l'intérieur de cerveau on a des milliards de neuronique transmettent de l'information sous forme d'impulsions électrique (Figure 1.11). En schématiser le fonctionnement d'un neurone on 3 étapes :

-des dendrites qui reçoivent l'impulsion.

-le neurone retransmet l'impulsion à travers in long câble qu'on appelle l'axone.

-les synapses qui envoi l'impulsion a un autre neurone.

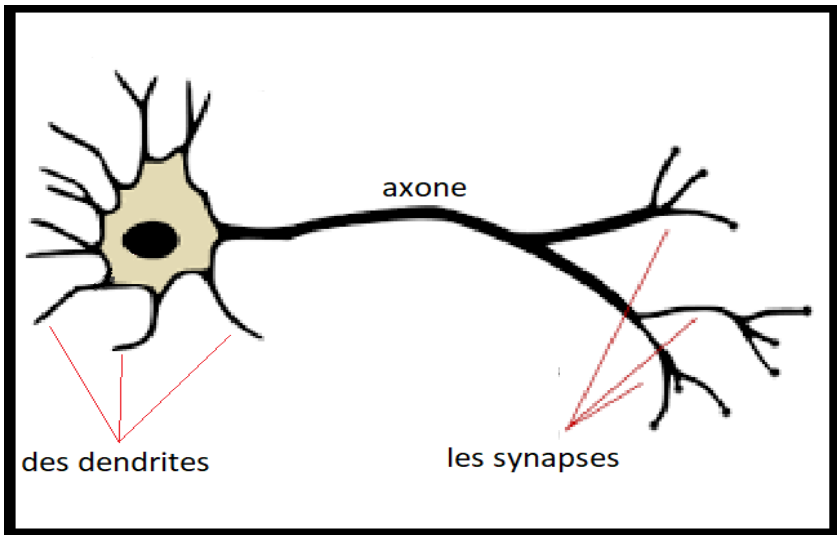

FIGURE 1.11: Schéma d'un neurone

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.[8]

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 (Figure 1.12).

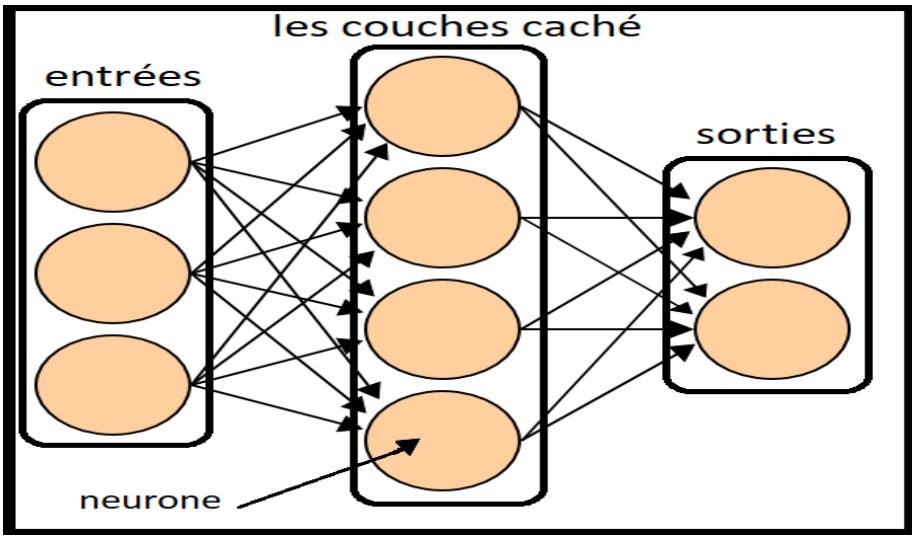

FIGURE 1.12: Schéma d'un neurone

1.5 Évaluation du processus de catégorisation

Afin de s'assurer de la capacité du classifieur à bien classer de nouveaux textes, il est nécessaire d'évaluer les performances du classifieur. Plusieurs mesures originaires du domaine de la Recherche d'Information ont été utilisées pour évaluer les performances d'une C.T.

La précision :[20] définit la précision en apprentissage comme la probabilité conditionnelle qu'un exemple choisi aléatoirement soit bien classé par le système et généralement la formule utilisée est :

Avec :

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.

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

Le rappel : est la proportion des solutions pertinentes qui sont trouvées autrement dit la mesure de capacité du système à donner toutes les solutions pertinentes [21], généralement on peut la calculer par la formule suivante :

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

La macro-moyenne (traduction de macro-averaging) évalue d'abord indépendamment chaque catégorie. Ensuite, la performance globale du classifieur est calculée en faisant la moyenne des mesures individuelles. Les différentes catégories ont alors la même importance [21].

Il existe tout de même d'autres critères pour mesurer les performances comme : la capacité d'apprentissage, l'efficacité d'apprentissage, l'efficacité de classement. [21]

1.6 Domaine d'applications de la catégorisation des textes

La catégorisation de textes peut être utilisée dans différentes domaines, parmi lesquels :

- **L'étiquetage de documents :** consiste à classifier les documents dans des catégories pour par exemple facilite la recherche.
- **Le filtrage :** c'est un domaine très large pour parmi l'utilisation de classification c'est le filtrage de spam ou en général déterminer si un document est pertinent ou non pour l'utilisateur.
- **Le routage :** consistant à affecter un document à une ou plusieurs catégories parmi n et beaucoup d'autres applications. [8]
- **L'identification de la langue :** consiste à la détection automatique du langage d'un texte pour garantir que les résultats de la recherche être dans la même langage.

1.7 Conclusion

Nous avons tenté dans ce chapitre de définir les techniques de la catégorisation automatique des textes et aussi leurs avantages et leurs inconvénients, puis nous avons cité les différents moyens d'évaluation d'un classificateur. A la fin ce domaine de la catégorisation devient dans ces dix dernières années très essentielles dans plusieurs domaines à l'aide des techniques d'intelligence ou d'apprentissage automatique qui ont donné des résultats meilleurs par rapport au technique précédente.

Dans le chapitre suivant nous présentons les différentes mesures de similarité syntaxique et sémantique

Chapitre 2

Les mesures de similarités syntaxiques et sémantiques

2.1 Introduction

La similarité entre documents textuels est une des problématiques importantes de plusieurs disciplines comme la catégorisation des textes, l'analyse de données textuelles, la recherche d'information ou l'extraction de connaissances à partir de données textuelles (Text Mining).

Nous allons nous concentrer sur la catégorisation des textes (l'évaluation des similarités entre documents et document pour identifier la catégorie du document).

Dans ce chapitre, Nous présenterons quelques définitions sur les mesures de similarités, en premier lieu commençant par les mesures syntaxiques les plus connues ensuite les mesures de similarités sémantiques. Parmi les approches citées dans la littérature on peut citer :

2.2 Similarité syntaxique

C'est une fonction mathématique utilisée en informatique pour le but de comparaison des documents textuels autrement dit consiste à comparer des valeurs statistiques ne changent pas en fonction de leurs points de ressemblance et de dissemblance.

2.2.1 Similarité Cosinus

La similarité cosinus est basée sur la représentation vectorielle. Selon « Baeza et Ribeiro »[21] cette approche est fréquemment utilisée en tant que mesure de ressemblance entre deux documents d1 et d2. Il s'agit de calculer le cosinus de l'angle entre les représentations vectorielles des documents à comparer. Cette approche est calculée par la loi suivante :

$$sim_{cosinus}(d_1, d_2) = \frac{\vec{d_1}.\vec{d_2}}{\|\vec{d_1}\|\|\vec{d_2}\|}$$

2.2.2 Distance euclidienne

La distance euclidienne mesure la similarité entre deux documents d_1 et d_2 comme illustré dans la figure 2.1.La mesure s'exprime par la loi suivante :

$$sim_{euclidienne}(d_1, d_2) = \| \vec{d_1} - \vec{d_2} \| = \sqrt{\sum_{i=1}^{n} (d_{1_i} - d_{2_i})^2}$$

Où n est le nombre total de termes représentés, i.e. la taille des vecteurs.

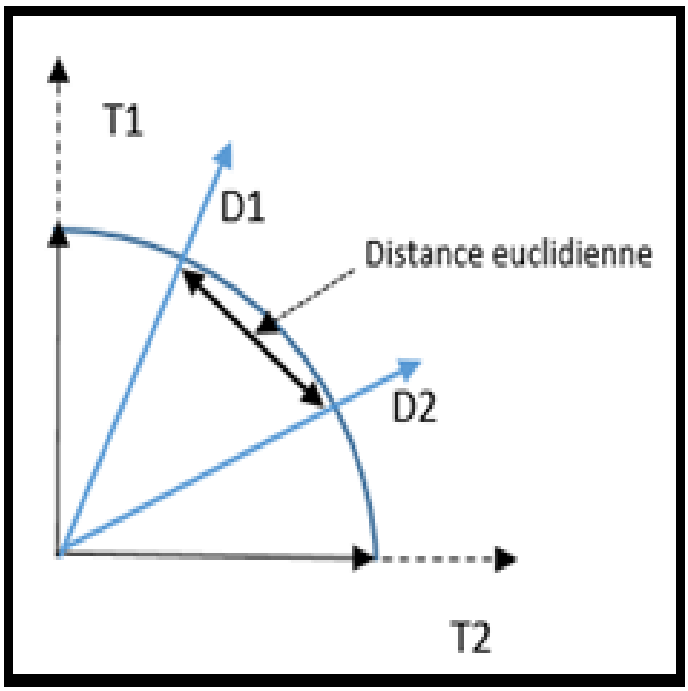

FIGURE 2.1: Exemple de la Distance euclidienne.
[1]

2.2.3 Indice de Dice

L'indice de Dice est basé sur le nombre des mots(concept) communs entre deux document. La mesure s'exprime par la loi suivante :

$$\text{dice}(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|}.$$

Où X et Y deux ensembles. On note |Z| le nombre d'éléments d'un ensemble Z.

2.2.4 Coefficient de Jaccard

Selon « Jaccard » [22] L'indice de Jaccard est le rapport entre la cardinalité (la taille) de l'intersection des ensembles considérés et la cardinalité de l'union des ensembles. La similarité obtenue appartient à l'intervalle [0, 1]. La mesure s'exprime par la loi suivante :

$$sim_{jaccard}(d_1, d_2) = \frac{\|d_1 \cap d_2\|}{\|d_1 \cup d_2\|}$$

2.3 Similarité sémantique

Les inconvénients des mesures syntaxiques sont souvent reliés à l'absence de la sémantique, par exemple dans le cas de deux mots différents en forme et similaire en sens est considéré comme dissemblable, ce qui peut conduire à la dégradation des performances.

2.3.1 Approches basées sur les arcs

Cette approche est très simple Il s'agit d'estimer la distance entre les nœuds correspondants aux concepts, la distance conceptuelle peut facilement être mesurée par la distance géométrique entre les nœuds représentants les concepts. Évidemment, plus le chemin d'un nœud à l'autre est court, plus ils sont similaires.

Leacock et Chodorow

Selon « Leacock Chodorow » [23] cette mesure est basée sur la longueur du plus court chemin entre deux concepts de l'ontologie. Cette mesure est représentée par la formule suivante :

$$sim_{lch}(c_1, c_2) = -log\frac{longueur}{2D}$$

Où :

Longueur : est la longueur du plus court chemin entre c_1 et c_2 (en terme de nombres de noeuds)

D : est la profondeur/hauteur maximale de la taxonomie.

Wu et Palmer

Selon Wu et Palmer [24],cette mesure consiste à mesurer la profondeur de deux concepts donnés dans la taxonomie. Pour trouver la similarité entre deux éléments C1 et C2 de l'ontologie, on calcul les distances (N1 et N2) qui séparent les nœuds C1 et C2 du nœud racine et la distance qui sépare le concept subsumant (CS) de C1 et C2 du nœud R. La mesure de Wu et Palmer est définie par la formule suivante :

$$sim_{wup}(c1, c2) = \frac{2 * N}{N1 + N2}$$

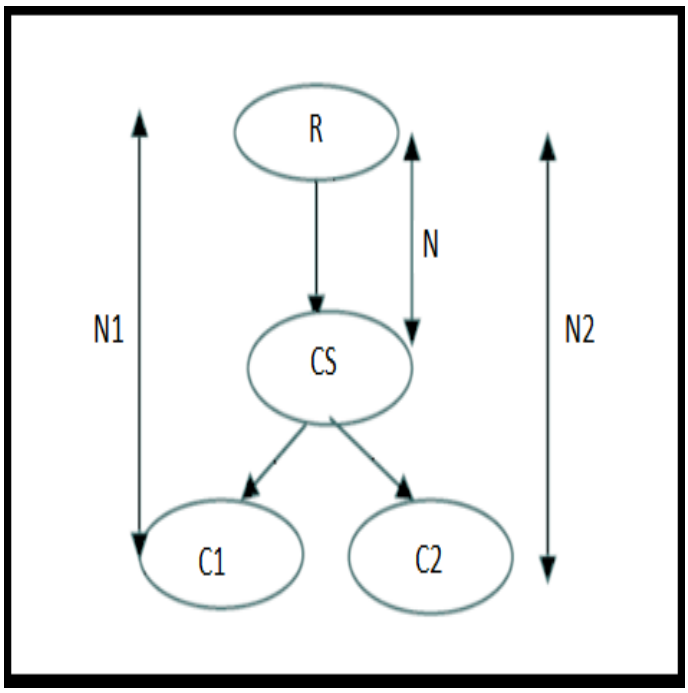

FIGURE 2.2: Exemple de Wu et Palmer.
[2]

La mesure de Hirst-St.Onge

Cette mesure a été crée par G.Hirst et D.St-Onge qui prend en considération toutes les relations dans WordNet. La similarité est calculée entre les concepts par les poids du chemin le plus court qui mène d'un concept à un autre avec l'idée de deux concepts sont proches sémantiquement si leurs concepts sont connectés par un chemin court et qui ne change pas souvent de direction. [25]

Il est calculé en fonctions des changements de direction par la loi suivante :

$$Sim(c1, c2) = T - chemin - K \times d$$

Tels que :

T et K : constantes.

Chemin : la longueur du chemin le plus court en nombre d'arcs.

d : le nombre de changements de direction.

Mesure de Slimani

Cette mesure a été adoptée pour apporter des améliorations à certains problèmes de la mesure de Wu Palmer liées à sa structure hiérarchique [26]. La mesure s'exprime par la loi :

$$Sim_{tbk}(c_1, c_2) = \frac{2 \times P_3}{P_1 + P_2} \times fp(c_1, c_2)$$

avec

$$fp(c_1, c_2) = \begin{cases} \frac{1}{|P_1 - P_2| + 1} & \text{Si } c_1 \text{ et } c_2 \text{ sont inclus dans le même chemin ;} \\ 1 & \text{sinon.} \end{cases}$$

les avantages et les inconvénients

« Aly Ngoné Ngom » résume dans [27] les avantages et les inconvénients de ces mesures.

Mesures	Année	Avantages	Inconvénients
Wu-Palmer	1994	Simple et facile a implémenter , prend en compte la profondeur des concepts.	Ne donne pas une bonne similarité entre concepts voisins et concepts de la même hiérarchie.
Leacock-Chodorow	1998	Simple a implémenter	Ne prend en compte que la relation is-a , moins performante que Wu-Palmer sur WordNet.
Hirst-ST Onge	1998	Permet d'évaluer la similarité entre nom et verbe sur WordNet.	Limitée par des restrictions sur le nombre de chemins.
Slimani	2007	Simple et facile a implémenter ,prend en compte la profondeur des concepts et la similarité entre concepts voisins et concepts de la même hiérarchie	Trop dépendante a l'organisation des concepts dans la Taxonomie.

TABLE 2.1: avantages et inconvénients des mesures de similarité sémantiques [27]

2.3.2 Approches basées sur les nœuds

Une approche basée sur les nœuds pour le but de déterminer la similarité conceptuelle et aussi appelée une approche de contenu informationnel (CI), parmi les mesures connues basées sur les nœuds citons :

Mesure de Resnik

Cette mesure est basée sur La notion du contenu informationnel (CI) qui a été introduite par [28] dans le but de calculer la similarité entre deux concepts en utilisant la quantité d'information commune entre eux. La mesure s'exprime par la loi :

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

La mesure de Jiang-Conrath

« Jiang-Conrath » [29] ont proposé une nouvelle mesure pour remédier Le problème de mesure de Resnik. Cette mesure est une combinaison entre les approchs basées sur les arcs et les approches basées sur les nœuds. Cette mesure est représentée par la formule suivante :

$$distance(c1, c2) = CI(c1) + CI(c2) - (2.CI(ppg(c1, c2)))$$

$$Sim(c1, c2) = 1/distance(c1, c2)$$

2.4 Conclusions

Dans ce chapitre nous avons abordé les différentes mesures de similarités syntaxiques et sémantiques avec quelques avantages et inconvénients.

Le chapitre suivant donne un aperçu sur notre travail qui est une évaluation des mesures de similarité sémantiques dans le cadre de la catégorisation des textes.

Chapitre 3

Enrichissement de la représentation conceptuelle

3.1 Introduction

L'objectif de la catégorisation de texte est de classifier des documents qui se rapprochent dans la même catégorie. Pour faire une bonne catégorisation, l'étape de représentation est très importante, pour cela il faut commencer par choisir la méthode de représentation. La représentation vectorielle non sémantique est l'une des méthodes les plus utilisées et elle est très populaires mais elle représente un inconvénient majeur qui est l'indépendance des termes, puisque les termes sont traités indépendamment impliquera un mauvais résultat de catégorisation car ne prend pas la notion de sens. Comme solution à ce problème, une autre méthode a été proposée qui est la représentation conceptuelle, cette méthode prend en charge la sémantique du document et ne donne pas beaucoup d'importance aux termes, néanmoins deux représentations conceptuelles peuvent être très similaires alors qu'ils ne partagent aucun concept impliquera des résultats de catégorisation non satisfaisable. Ainsi, il est nécessaire d'appliquer un enrichissement à cette représentation conceptuelle afin de résoudre ce problème.

Nous allons présenter une application qui a pour but d'enrichir la représentation conceptuelle des documents. Ainsi, deux approches d'enrichissement ont été proposées, un enrichissement global et local.

Dans ce chapitre, nous allons présenter l'architecture générale de notre catégorisation de textes avec ces deux phases apprentissage et classification, puis nous allons détailler les deux approches d'enrichissement proposées. La dernière section sera consacrée au déroulement de notre programme, les ressources utilisées, ainsi que les expérimentations effectuées.

3.2 Architecture générale

Comme montré dans la figure 3.1, l'architecture de notre approche se compose de trois phases. La première phase est la phase de représentation, elle consiste à générer une représentation conceptuelle des documents étiquetés ainsi que le document à catégoriser. La deuxième phase est la phase d'enri-

chissement dont laquelle réside l'essentielle de notre travail à savoir la proposition de deux approches pour l'enrichissement de la représentation générée lors de la première phase. La troisième phase est la phase de classification, elle consiste à construire un modèle de prédiction à partir de la matrice conceptuelle enrichie lors de la deuxième phase dans le but de pouvoir catégoriser le nouveau document.

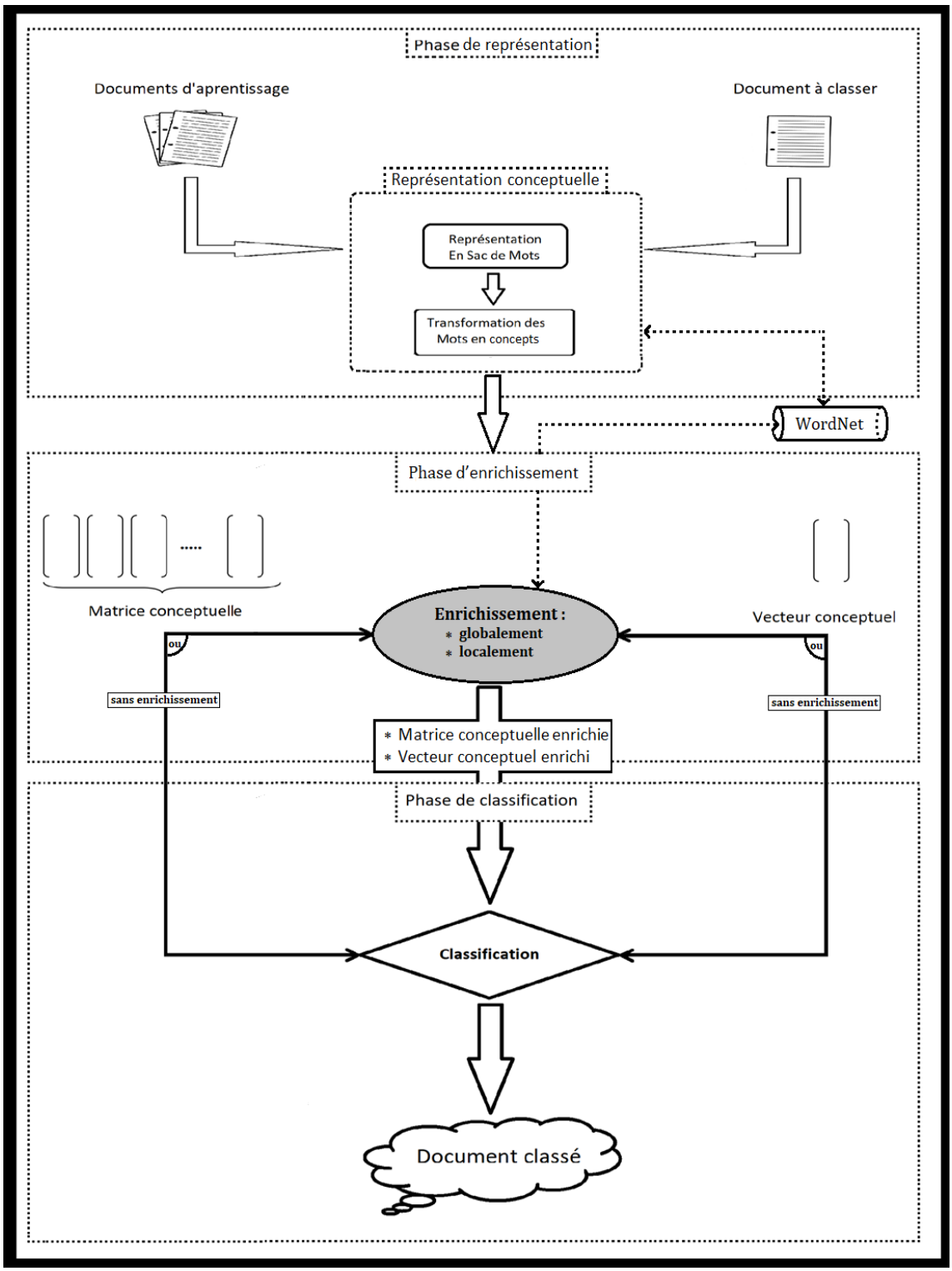

FIGURE 3.1: Architecture générale.

3.2.1 Phase de représentation

Cette phase est importante pour le bien déroulement des phases suivantes, en effet, il est important de représenter les documents sous une forme exploitable par les algorithmes d'apprentissage tout en préservant le plus possible la sémantique des documents.

Dans notre travail nous avons choisi la représentation conceptuelle qui consiste à transformer chaque document en un ensemble de concepts. Pour cela, il est nécessaire de passer par les étapes suivantes :

Représentation en sac de mot

Cette représentation consiste à transformer un texte sous forme d'un vecteur dont chaque composante représente un mot. Afin d'obtenir de telle représentation, il est nécessaire d'appliquer les prétraitements suivants sur les documents classés ainsi que les documents non classés.

L'exemple de la figure suivante illustre la série de prétraitements pour obtenir à la fin un vecteur qu'est composé par des mots :

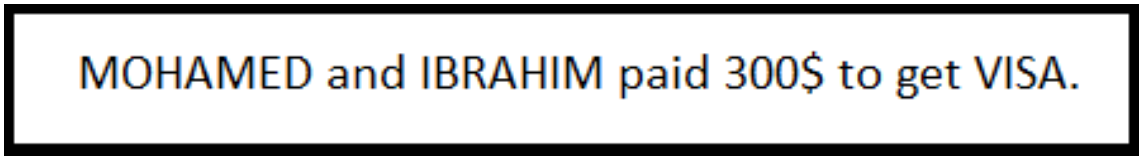

FIGURE 3.2: Exemple d'un document.
[2]

— **Tokenisation :** Consiste à faire une élimination des signes de ponctuation (+-*/ ; :.() !,' ?<>) et les caractères spéciaux, la figure 3.3 donne une aperçu sur l'application de la Tokenisation.

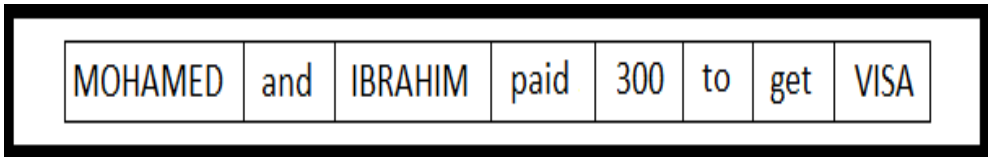

FIGURE 3.3: Application de la Tokenisation sur notre document.
[3]

— **L'élimination des majuscules :** Consiste à rendre les caractères majuscules en minuscules, le résultat est dans la figure 3.4).

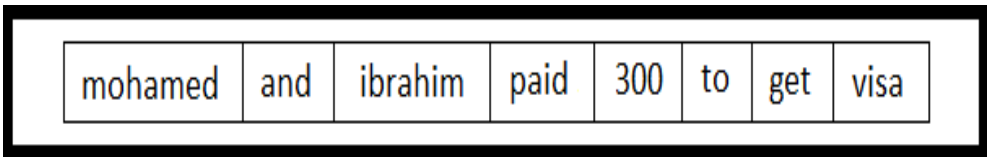

FIGURE 3.4: Document sans majuscule.
[4]

— **Élimination des mots vides :** Les mots vides sont les mots qui n'ont aucun sens lors du processus de la catégorisation de texte. La figure 3.5 montre la liste des mots vides qui appartient à la langue anglaise.

—

'a', 'about', 'above', 'across', 'after', 'again', 'against', 'all', 'almost', 'alone', 'along', 'already', 'also', 'although', 'always', 'among', 'an', 'and', 'another', 'any', 'anybody', 'anyone', 'anything', 'anywhere', 'are', 'area', 'areas', 'around', 'as', 'ask', 'asked', 'asking', 'asks', 'at', 'away', 'b', 'back', 'backed', 'backing', 'backs', 'be', 'became', 'because', 'become', 'becomes', 'been', 'before', 'began', 'behind', 'being', 'beings', 'best', 'better', 'between', 'big', 'both', 'but', 'by', 'c', 'came', 'can', 'cannot', 'case', 'cases', 'certain', 'certainly', 'clear', 'clearly', 'come', 'could', 'd', 'did', 'differ', 'different', 'differently', 'do', 'does', 'done', 'down', 'down', 'downed', 'downing', 'downs', 'during', 'e', 'each', 'early', 'either', 'end', 'ended', 'ending', 'ends', 'enough', 'even', 'evenly', 'ever', 'every', 'everybody', 'everyone', 'everything', 'everywhere', 'f', 'face', 'faces', 'fact', 'facts', 'far', 'felt', 'few', 'find', 'finds', 'first', 'for', 'four', 'from', 'full', 'fully', 'further', 'furthered', 'furthering', 'furthers', 'g', 'gave', 'general', 'generally', 'get', 'gets', 'give', 'given', 'gives', 'go', 'going', 'good', 'goods', 'got', 'great', 'greater', 'greatest', 'group', 'grouped', 'grouping', 'groups', 'h', 'had', 'has', 'have', 'having', 'he', 'her', 'here', 'herself', 'high', 'high', 'high', 'higher', 'highest', 'him', 'himself', 'his', 'how', 'however', 'i', 'if', 'important', 'in', 'interest', 'interested', 'interesting', 'interests', 'into', 'is', 'it', 'its', 'itself', 'j', 'just', 'k', 'keep', 'keeps', 'kind', 'knew', 'know', 'known', 'knows', 'l', 'large', 'largely', 'last', 'later', 'latest', 'least', 'less', 'let', 'lets', 'like', 'likely', 'long', 'longer', 'longest', 'm', 'made', 'make', 'making', 'man', 'many', 'may', 'me', 'member', 'members', 'men', 'might', 'more', 'most', 'mostly', 'mr', 'mrs', 'much', 'must', 'my', 'myself', 'n', 'necessary', 'need', 'needed', 'needing', 'needs', 'never', 'new', 'new', 'newer', 'newest', 'next', 'no', 'nobody', 'non', 'noone', 'not', 'nothing', 'now', 'nowhere', 'number', 'numbers', 'o', 'of', 'off', 'often', 'old', 'older', 'oldest', 'on', 'once', 'one', 'only', 'open', 'opened', 'opening', 'opens', 'or', 'order', 'ordered', 'ordering', 'orders', 'other', 'others', 'our', 'out', 'over', 'p', 'part', 'parted', 'parting', 'parts', 'per', 'perhaps', 'place', 'places', 'point', 'pointed', 'pointing', 'points', 'possible', 'present', 'presented', 'presenting', 'presents', 'problem', 'problems', 'put', 'puts', 'q', 'quite', 'r', 'rather',

'really', 'right', 'right', 'room', 'rooms', 's', 'said', 'same', 'saw', 'say',
'says', 'second', 'seconds', 'see', 'seem', 'seemed', 'seeming', 'seems',
'sees', 'several', 'shall', 'she', 'should', 'show', 'showed', 'showing',
'shows', 'side', 'sides', 'since', 'small', 'smaller', 'smallest', 'so', 'some',
'somebody', 'someone', 'something', 'somewhere', 'state', 'states',
'still', 'still', 'such', 'sure', 't', 'take', 'taken', 'than', 'that', 'the', 'their',
'them', 'then', 'there', 'therefore', 'these', 'they', 'thing', 'things',
'think', 'thinks', 'this', 'those', 'though', 'thought', 'thoughts', 'three',
'through', 'thus', 'to', 'today', 'together', 'too', 'took', 'toward', 'turn',
'turned', 'turning', 'turns', 'two', 'u', 'under', 'until', 'up', 'upon', 'us',
'use', 'used', 'uses', 'v', 'very', 'w', 'want', 'wanted', 'wanting', 'wants',
'was', 'way', 'ways', 'we', 'well', 'wells', 'went', 'were', 'what', 'when',
'where', 'whether', 'which', 'while', 'who', 'whole', 'whose', 'why',
'will', 'with', 'within', 'without', 'work', 'worked', 'working', 'works',
'would', 'x', 'y', 'year', 'years', 'yet', 'you', 'young', 'younger',
'youngest', 'your', 'yours', 'z'

FIGURE 3.5: Liste des mots vides.
[5]

Nous prenons l'exemple précèdent et appliquons l'étape d'élimination des mots vides à partir de la liste de figure 3.5, le résultat est dans la figure 3.6.

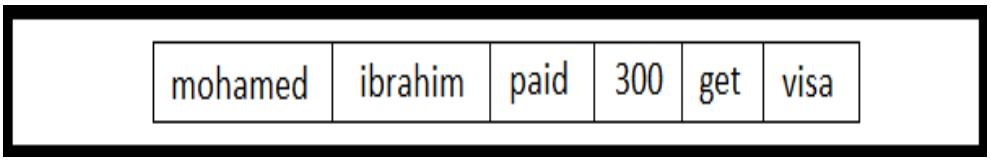

FIGURE 3.6: L'étape d'élimination des mots vides.
[6]

Après avoir terminé les étapes précédentes nous commençons par construire la matrice d'occurrence qui contient dans les colonnes les termes et dans les lignes les documents, l'intersection de la ligne i avec la colonne j

donne la fréquence du terme j dans le document i, cette matrice est nécessaire pour distinguer le contenu de chaque document par rapport aux autres documents.

— **Transformation des mots en concepts :**
Dans cette étape nous avons intégré la notion du sens, autrement dit transformer les mots en des concepts, pour cela nous avons commencé par l'étiquetage grammatical qui consiste à distinguer la catégorie grammaticale (verbe, adjective, adverbe, nom...) des mots dans le texte. La Figure 3.7 illustre les catégories grammaticales pour la langue anglaise.

A	adjective	P	preposition
B	adverb	Q	numeral
C	conjunction	V	verb
D	article	p	prefix
I	interjection	s	suffix
N	noun	?	undetermined
O	pronoun		

FIGURE 3.7: Catégories grammaticales pour la langue anglaise.
[7]

Une fois la catégorie grammaticale trouvée nous passons à l'étape suivante qui consiste à trouver pour chaque mot tous les concepts associés parce qu'un mot peut avoir plusieurs concepts.
Prenons comme exemple le mot « cat », ce mot peut avoir dix concept, la figure 3.8 donne un aperçu sur les concepts du mot « cat ».

```
les concepts de mot cat :

   concept 1 : cat.n.01
   definition : feline mammal usually having thick soft fur and no ability to roar:
domestic cats; wildcats
-------------------------------------------
   concept 2 : guy.n.01
   definition : an informal term for a youth or man
-------------------------------------------
   concept 3: cat.n.03
   definition : a spiteful woman gossip
-------------------------------------------
   concept 4 : kat.n.01
   definition : the leaves of the shrub Catha edulis which are chewed like tobacco
or used to make tea; has the effect of a euphoric stimulant
-------------------------------------------
   concept 5 : cat-o'-nine-tails.n.01
   definition : a whip with nine knotted cords
-------------------------------------------
   concept 6: caterpillar.n.02
   definition : a large tracked vehicle that is propelled by two endless metal belts;
frequently used for moving earth in construction and farm work
-------------------------------------------
   concept 7: big_cat.n.01
   definition : any of several large cats typically able to roar and living in the
wild
-------------------------------------------
   concept 8 : computerized_tomography.n.01
   definition : a method of examining body organs by scanning them with X rays and
using a computer to construct a series of cross-sectional scans along a single axis
-------------------------------------------
   concept 9 : cat.v.01
   definition : beat with a cat-o'-nine-tails
-------------------------------------------
   concept 10 : vomit.v.01
   definition : eject the contents of the stomach through the mouth
-------------------------------------------
```

FIGURE 3.8: Les dix concepts du mot « cat ».
8

Nous observons que chaque concept est codé en trois parties sous la forme « Mot ». «Pos ». « Numéro ».

Où :

— «Mot» est le lemme du mot.
— « Pos » est la catégorie grammaticale du mot.
— « Numéro » est l'index du sens.

Par exemple, « cat.n.01 » signifie le premier concept du nom « cat ».

Une fois que nous avons obtenu le résultat de l'étape précédente nous avons remarqué qu'on a un problème d'ambiguïté ce qui nous a amené à faire un traitement de désambiguïsation.

Le traitement de désambiguïsation consiste à choisir pour chaque mot le concept adéquat parmi les concepts possibles, pour bien comprendre son fonctionnement on va prendre l'exemple suivant :

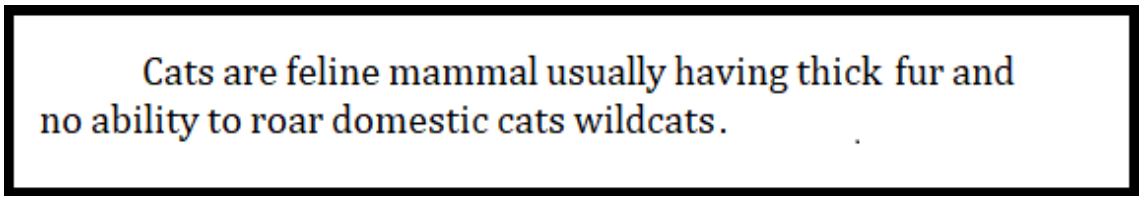

FIGURE 3.9: Exemple d'un deuxième document.
9

Le résultat de Désambiguïsation est illustré dans la figure 3.10. En effet, pour le mot « cats » la méthode de Désambiguïsation choisi le premier concept et pour le mot « wildcats » a choisi le troisième concept. .

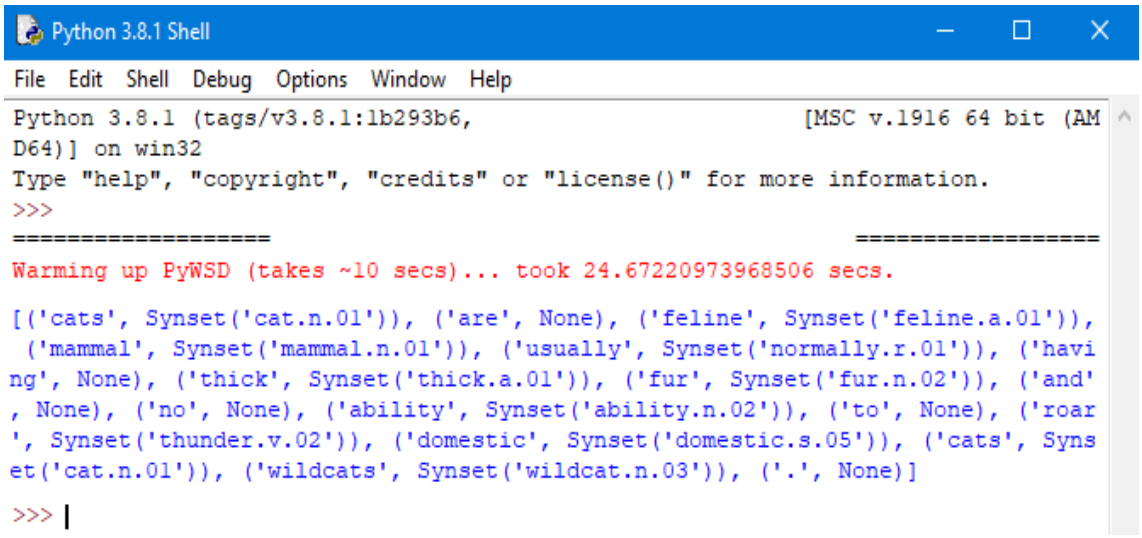

FIGURE 3.10: le vecteur des couples (Terme,Concept).
10

A partir du résultat de Désambiguïsation nous construisons le vecteur conceptuel du document, ce vecteur contient dans chaque composante le poids du concept dans le document, le tableau 3.11 illustre le vecteur conceptuel.

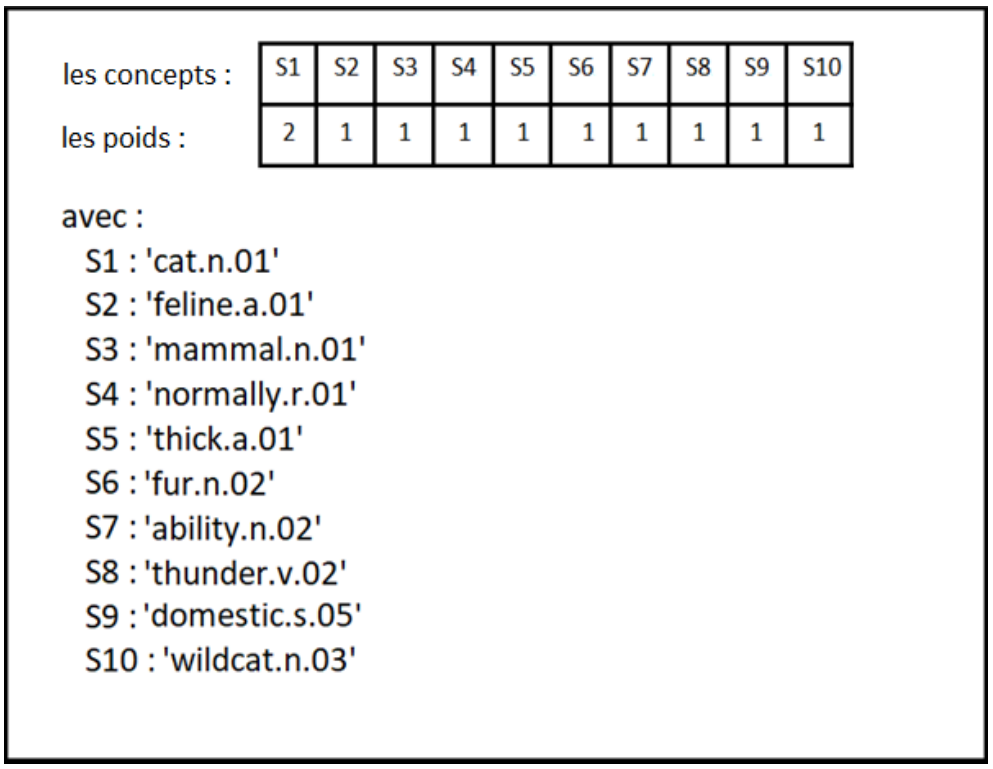

FIGURE 3.11: vecteur conceptuel.
11

Une fois la Désambiguïsation est terminée nous construisons une matrice conceptuelle (document, concept) et un vecteur conceptuel du document à catégoriser, la figure 3.12 illustre matrice (doc, concept) et vecteur conceptuel du document à catégoriser.

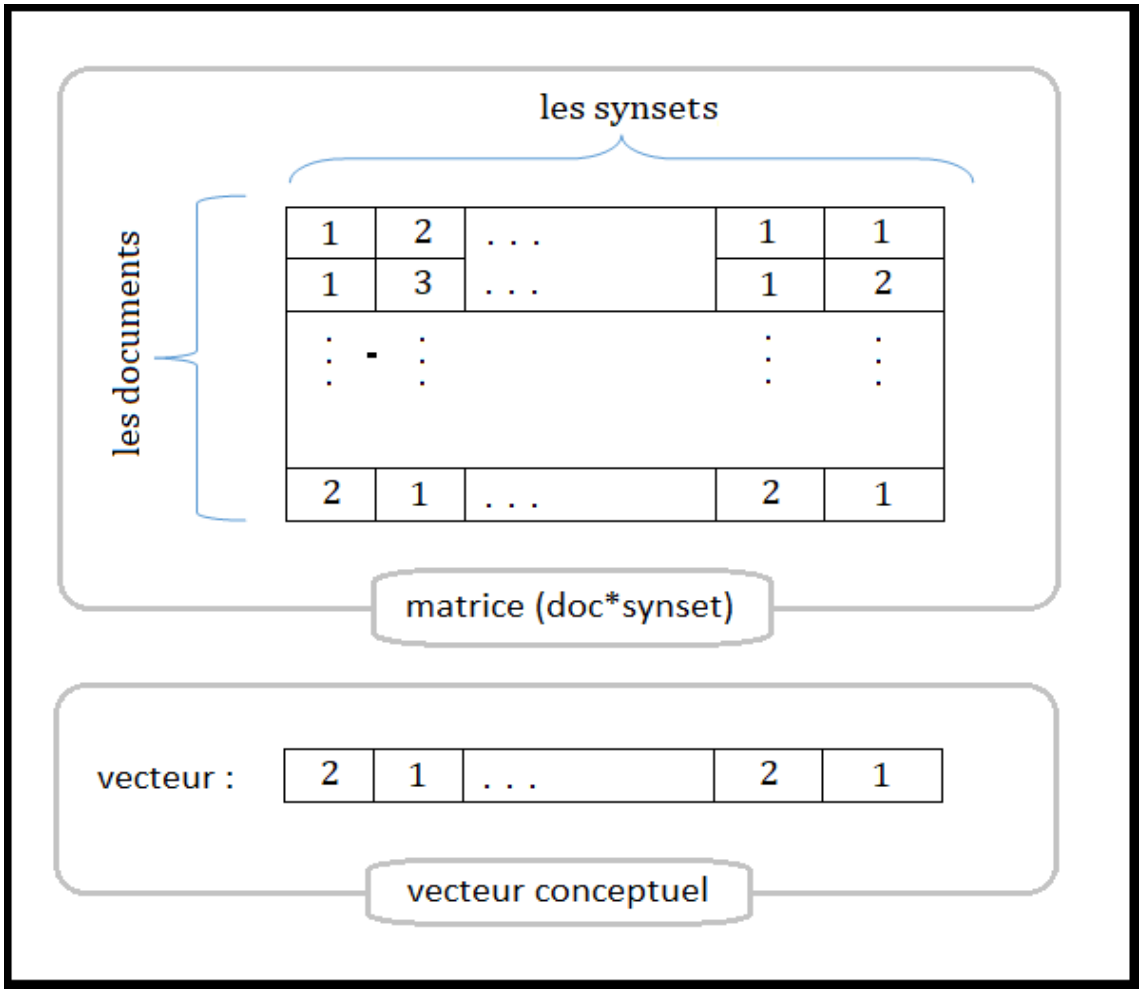

FIGURE 3.12: La matrice (document, concept) et le vecteur conceptuel de document à catégoriser.

3.2.2 La phase d'enrichissement

Après la phase représentation nous obtenons la matrice conceptuelle et le vecteur conceptuel puis on commence la phase d'enrichissement qui est la base de notre projet, on distingue dans cette phase trois approches (approche sans enrichissement, approche avec enrichissement globale et local).

Approche sans enrichissement

L'implémentation de cette approche est simple, il suffit de construire un vecteur (Initialisé par des zéros) qui à la taille du dictionnaire, ensuite nous cherchons les concepts de notre nouveau document qui existe dans le dictionnaire. Ainsi, si le concept du nouveau document n'existe pas dans le dictionnaire il sera tout simplement ignoré. L'algorithme présenté dans la figure3.13 illustre les étapes de cette approche.

Les entrées :
S1 :l'ensemble des concepts des documents classées.
S2 :l'ensemble des concepts de document à classer.
V : vecteur à la taille de l'ensemble S1 initialisé par des zéros.
Début :
Pour chaque concept C_i de S2 faire:
Pour chaque concept C_j de S1 faire:
Si $C_i = C_j$ alors:
V[j] ← Tf de C_i.
sortie de boucle
Fin Si
Fin pour
Fin pour
Fin

FIGURE 3.13: Algorithme de l'approche sans enrichissement.

Approche avec enrichissement global

Pour remédier au problème présenté au niveau de la première approche (l'ignorance des concepts absents de dictionnaire) nous avons proposé une approche d'enrichissement globale, qui consiste à éviter l'élimination des concepts (du document à catégoriser) ne figurons pas dans le dictionnaire en recherchant pour chaque nouveau concept le concept le plus proche parmi les concepts du dictionnaire. Les mesures de similarité sémantiques sont utilisées à ce stade afin de calculer le degré de rapprochement entre concepts. Ainsi, le concept le plus proche est choisi globalement sans prendre en considération sa présence / absence dans les documents. L'algorithme présenté dans la figure 3.14 illustre les étapes de cette approche.

Les entrées :

S1 : l'ensemble des concepts des documents classées.
S2 : l'ensemble des concepts de document à classer.
V : vecteur à la taille de l'ensemble S1 initialisé par des zéros.
P : la liste des concepts qui n'exister pas dans l'ensemble S1.

Les paramètres :

MSS : mesure de similarité sémantique utiliser.

Début :

```
Pour chaque concept Ci de S2 faire:
    bool ← vrai
    Pour chaque concept Cj de S1 faire:
        Si Ci = Cj alors:
            V[ j ] ← Tf de Ci .
            bool ← Faux
            sortie de boucle
        Fin Si
    Fin pour
    Si bool = vrai alors:
        Ajouter Ci à la liste P.
    Fin Si
Fin pour

Pour chaque concept Ci de P faire:
    max_sim ← -1
    ind ← -1
    Pour chaque concept Cj de S1 faire:
        sim ← MSS( Ci , Cj )
        Si max_sim < sim and sim > 0.8 alors:
            max_sim ← sim
            ind← j
        Fin Si
    Fin Pour
    Si max_sim <> -1 alors:
        V[ ind ] ← Tf de Ci * sim
    Fin Si
Fin Pour
```

Fin

Sortie : le vecteur V enrichi.

FIGURE 3.14: Algorithme de l'approche d'enrichissement globale.

Approche avec enrichissement local

A la différence d'enrichissement globale, nous avons proposé une nouvelle approche d'enrichissement qui consiste à localiser pour chaque nouveau concept, le concept qui lui est le plus proche dans chaque document étiqueté. Cette approche se base sur les concepts du nouveau document plutôt que ceux du dictionnaire. Ainsi le poids du nouveau concept sera mis à jour en fonction du poids du concept le plus proche dans le document étiqueté. Pour chaque document à catégoriser, une nouvelle matrice d'enrichissement local est créée pour but d'avoir une taille de matrice plus petite que celle de la méthode d'enrichissement globale. En effet la taille de cette matrice aura la même largeur que le vecteur conceptuel et la même longueur que la matrice conceptuelle. Cette matrice sera le résultat d'intersection entre le vecteur conceptuel et la matrice conceptuelle. Le déroulement de cet algorithme est représenté dans la figure 3.15.

Les entrées :
S1 : l'ensemble des concepts des documents classées.
S2 : l'ensemble des concepts de document à classer.
M : matrice (nbr des lignes de V , la taille de S2) initialisé par des zéros.
V : la matrice conceptuelle.
ml : vecteur à la taille de l'ensemble S1.
les paramétres :
MSS : mesure de similarité sémantique utiliser.
Début :

```
Pour chaque colonne j de V faire:
    max ← 0
    Pour chaque ligne i de V faire:
        Si l'element de V à la ligne i et la colonne j > max alors:
            max ← V[i][j]
        Fin Si
    Fin pour
    ml[j] ← max
Fin pour
Pour chaque ligne i de V faire:
    Pour chaque concept Cj de S2 faire:
        bool ← 0
        Pour chaque concept Ck de S1 faire:
            Si Cj = Ck and V[i][k] > 0 alors:
                M[i][j] ← Tf de Ck / ml[k] * Tf de Cj
                bool ← 1
            Fin Si
        Fin pour
        Si bool = 0 alors:
            max_sim ← -1
            ind ← -1
            Pour chaque concept Ce de S1 faire:
                Si V[i][e] > 0 alors:
                    sim ← MSS(Cj, Ce)
                    Si max_sim < sim and sim > 0.8 alors:
                        max_sim ← sim.
                        ind ← e
                    Fin Si
                Fin Si
            Fin pour
            Si max_sim <> -1 alors:
                M[i][j] ← Tf de Cind / ml[ind] * Tf de Cj * max_sim
            Fin Si
        Fin Si
    Fin pour
Fin pour
```

Sortie : nouvelle matrice conceptulle enrichie.

FIGURE 3.15: Algorithme de l'approche d'enrichissement local.

3.2.3 Application pratique sur les trois approches

A fin de distinguer entre les trois méthodes, prenons le petit exemple illustratif de la figure 3.16 qui se compose d'un corpus contenant deux catégories (chaque catégorie contient un seul document étiqueté) et d'un document à catégoriser.

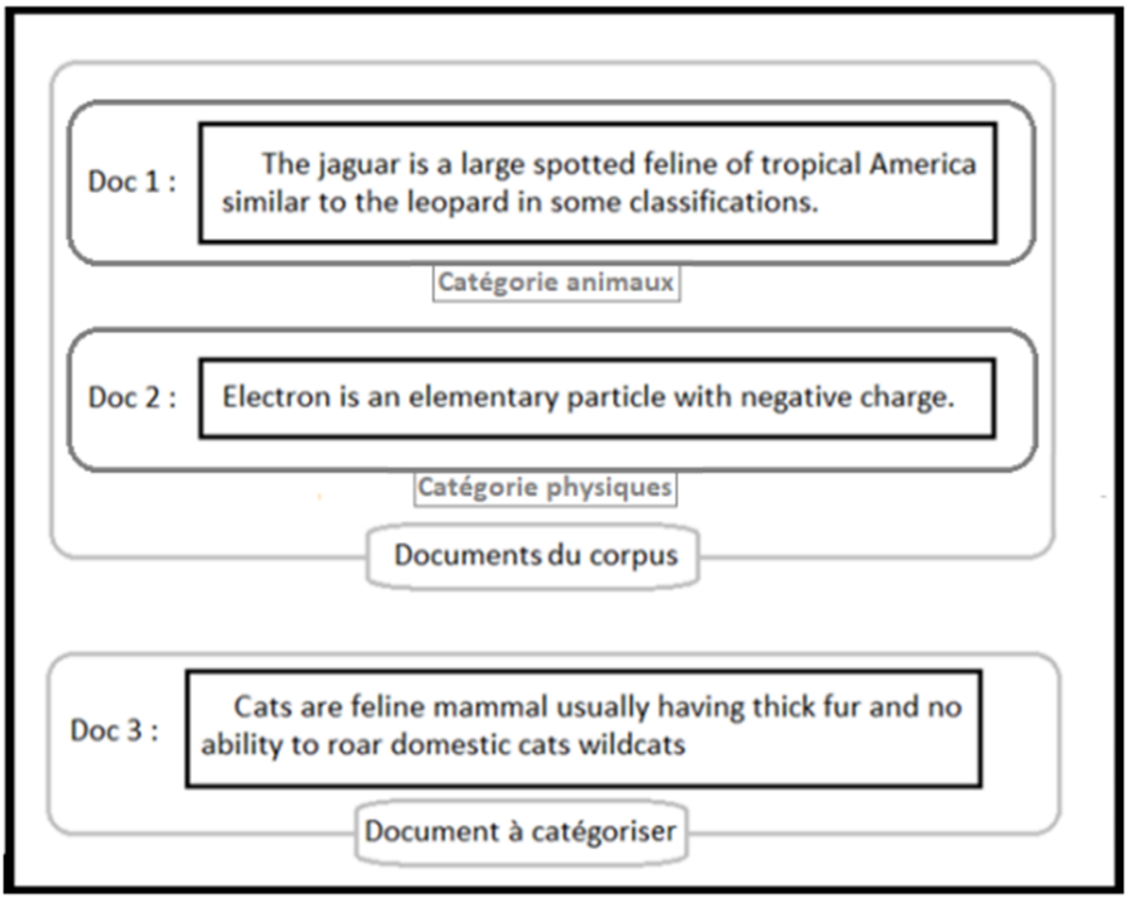

FIGURE 3.16: le corpus et le document à catégoriser.

La phase de représentation consiste à construire la matrice conceptuelle du corpus et le vecteur conceptuel du document à catégoriser, ceci est illustré dans la figure 3.17.

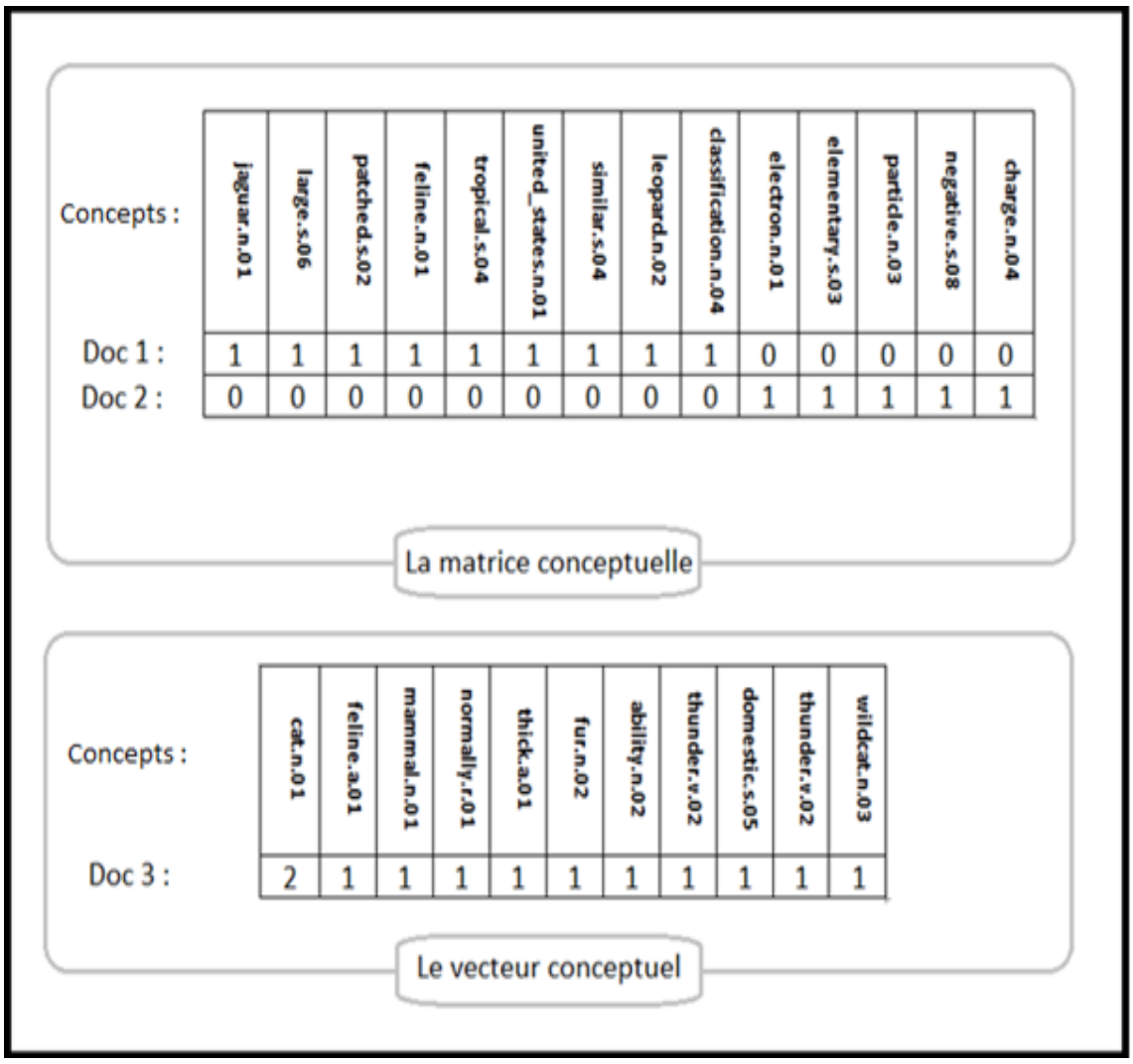

FIGURE 3.17: la matrice et le vecteur conceptuelle.
17

Dans la phase d'enrichissement nous avons parcouru les trois approches :

La première approche (approche sans enrichissement), nous avons parcouru les étapes de l'algorithme sans enrichissement (figure 3.13) et nous avons obtenu le résultat représenter dans la figure 3.18.

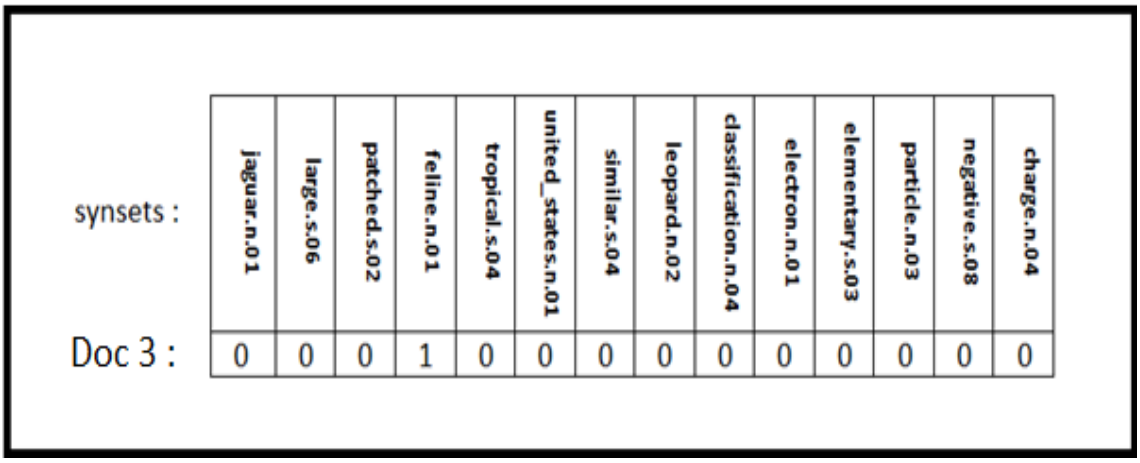

FIGURE 3.18: Résultat de l'approche sans enrichissent.
18

Nous observons dans cette approche que le nouveau vecteur ne partage qu'un seul concept avec les documents de corpus.

Dans l'approche avec enrichissement globale nous avons utilisé dans notre exemple la mesure de similarité sémantique du Wu Palmer avec un seuil supérieur à 80 %, le résultat représenter dans la figure 3.19.

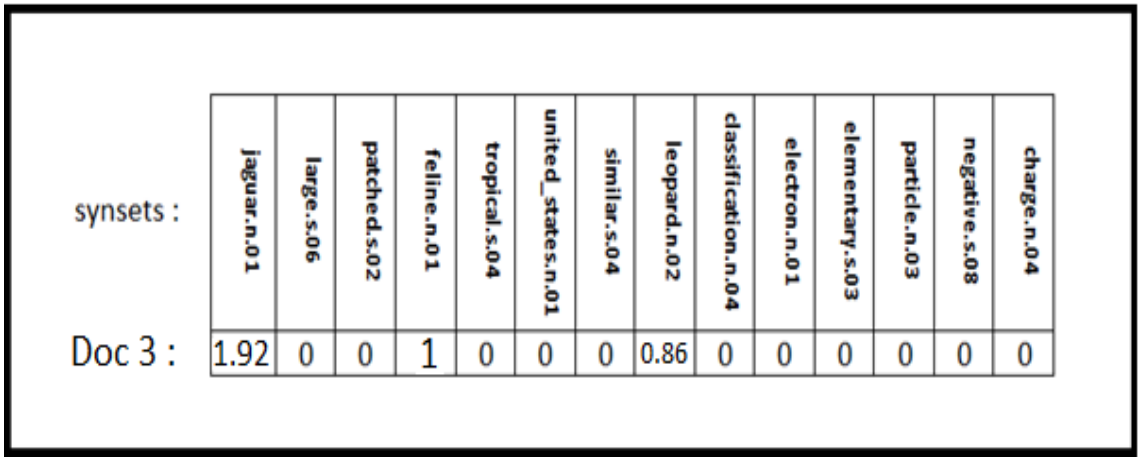

FIGURE 3.19: Résultat de l'approche d'enrichissement globale.
19

Nous finirons notre exemple par l'approche d'enrichissement locale qui utilise aussi la mesure de similarité sémantique du Wu Palmer avec un seuil supérieur à 80 %, le résultat représenter dans la figure 3.20.

Concepts :	cat.n.01	feline.a.01	mammal.n.01	normally.r.01	thick.a.01	fur.n.02	ability.n.02	thunder.v.02	domestic.s.05	thunder.v.02	wildcat.n.03
Doc 1 :	1.92	1	0.86	0	0	0	0	0	0	0	0.92
Doc 2 :	0	0	0	0	0	0	0	0	0	0	0

FIGURE 3.20: Résultat de l'approche d'enrichissement locale.
[20]

3.2.4 La phase de classification

Cette phase consiste à trouver la catégorie du nouveau document parmi les catégories de corpus, pour cela nous utilisons deux types des classificateurs (avec apprentissage, sans apprentissage) comme Les arbres de décisions, réseaux de neurone, les k plus proches voisins ...etc. Chaque classificateur à une méthode différant pour catégoriser les documents. Les entrées fournies aux classificateur se diffèrent selon la méthode d'enrichissement utilisée comme suit :

- Le cas d'approche sans enrichissement : la matrice conceptuelle avec le vecteur conceptuel pondéré.
- Le cas d'approche avec enrichissement globale : la matrice conceptuelle avec le vecteur conceptuel pondéré enrichi.
- Le cas de l'approche avec enrichissement locale : la nouvelle matrice enrichie avec le vecteur conceptuel.

La sortie de classificateur sera la catégorie du nouveau document.

3.3 Déroulement de notre programme

3.3.1 La représentation

FIGURE 3.21: Le chargement de corpus et le document à classer.
21

Cette fenêtre du figure 3.21 composée de trois parties, chaque patrie contient deux boutons, nous avons défini chaque bouton comme suite :

« **Importer une liste** » : importer une nouvelle liste de mots vides.

« **Liste par défaut** » : utiliser la liste de mots vides chargée dans l'application.

« **Importer nouveaux corpus** » : importer un corpus.

« **Importer dataset corpus** » : permet d'importer la matrice du corpus déjà traitée.

« **Écrire le document** » : utilisé dans le cas où l'utilisateur veut saisir son document.

« **Importer le document** » : importer un document avec une extension txt.

« **Suivant** » : utilisé pour passer à l'étape suivant.

3.3.2 L'enrichissement

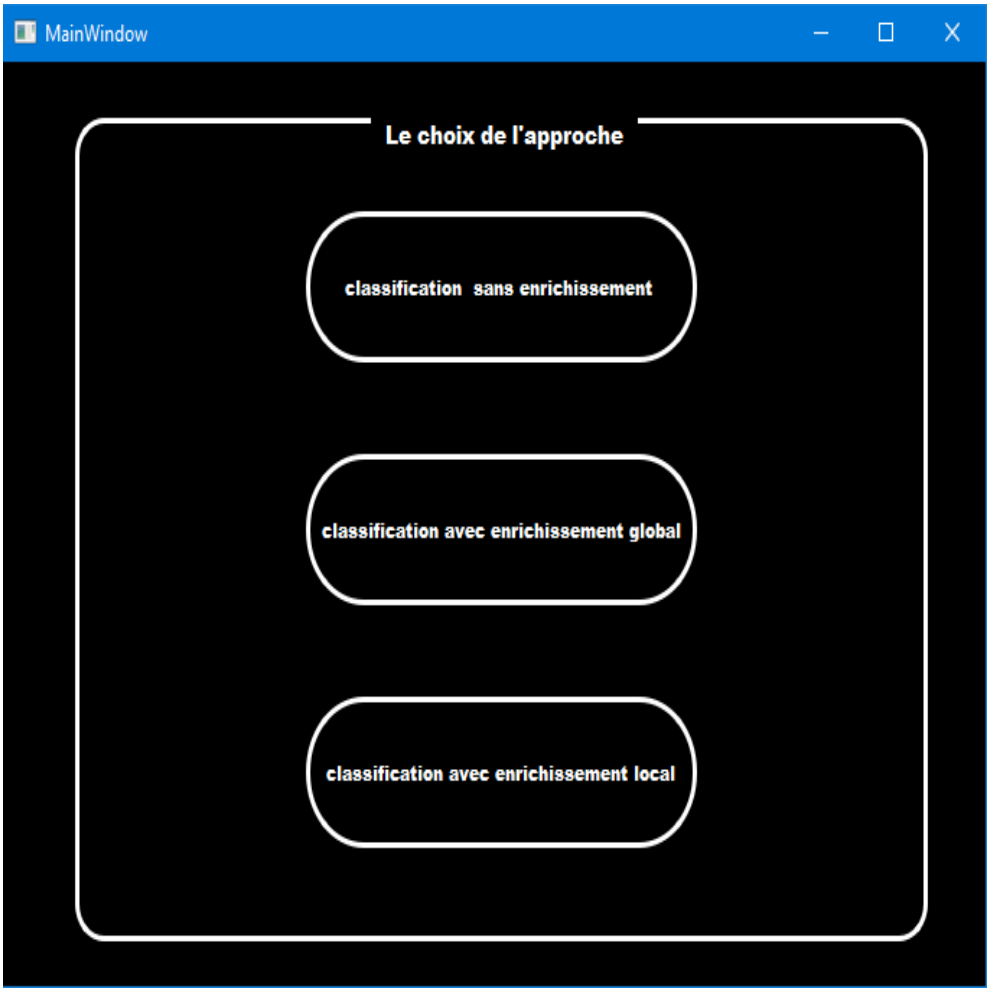

FIGURE 3.22: Le choix de l'approche.
22

Cette fenêtre se compose de trois boutons, le premier bouton sert à une classification sans enrichissement, le deuxième bouton sert à une classification avec enrichissement global et le dernier bouton sert à une classification avec enrichissement local.

Classification sans enrichissement

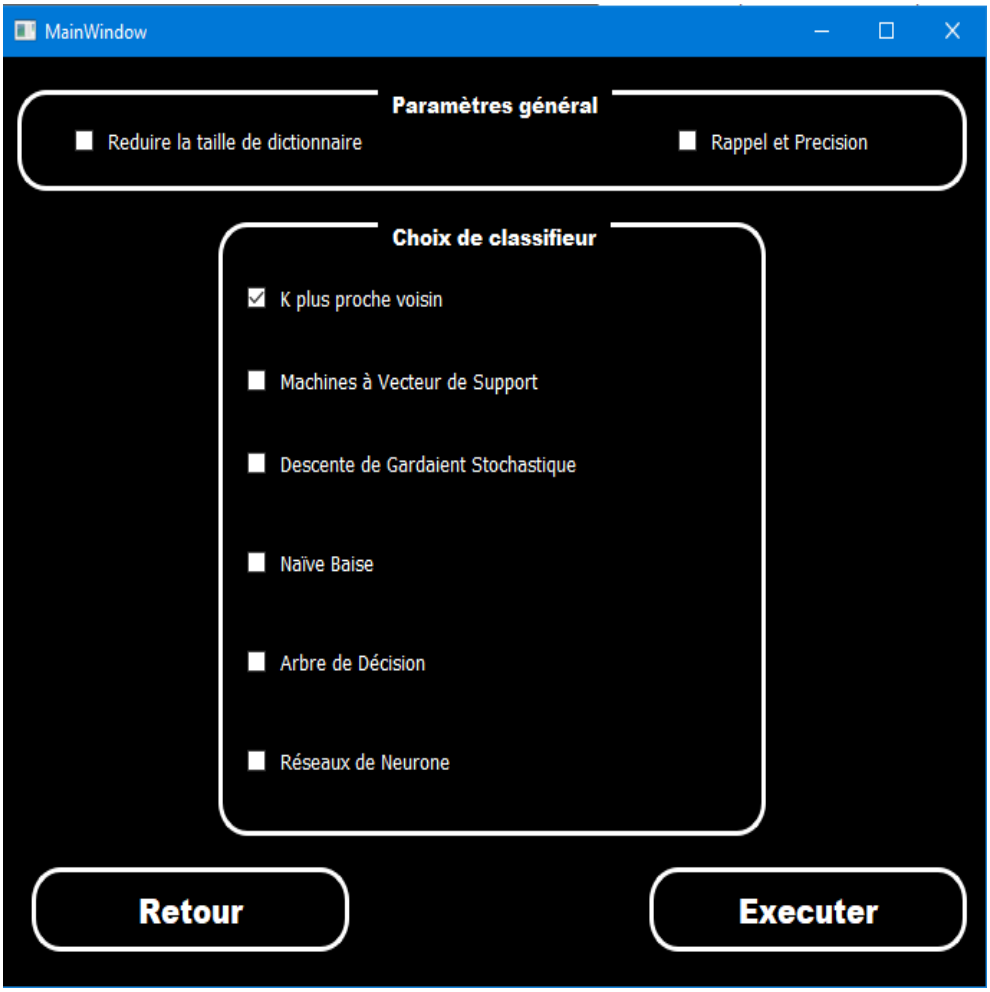

FIGURE 3.23: L'interface de l'approche sans enrichissement.
23

La figure 3.23 illustre notre fenêtre qui contient le choix de six classificateurs et on a deux options dans l'entête, réduire la taille du dictionnaire pour le rôle d'élimination les concepts qui existe dans plusieurs classes, rappel et précision pour afficher le rappel et la précision du dictionnaire.

Le bouton « Exécuter » apparaît lors de la sélection d'un classifieur, a le rôle d'exécuter la phase de classification. Le bouton « Retour » pour revenir à la fenêtre précédente.

Classification avec enrichissement global et locale

Dans cette fenêtre du figure 3.24 nous avons le choix de six mesures de similarité. Après la sélection d'une des mesures apparaît un bouton à droite pour passer à la fenêtre suivante qui contient six classificateur, la figure 3.25 illustre les différents classificateurs utilisés.

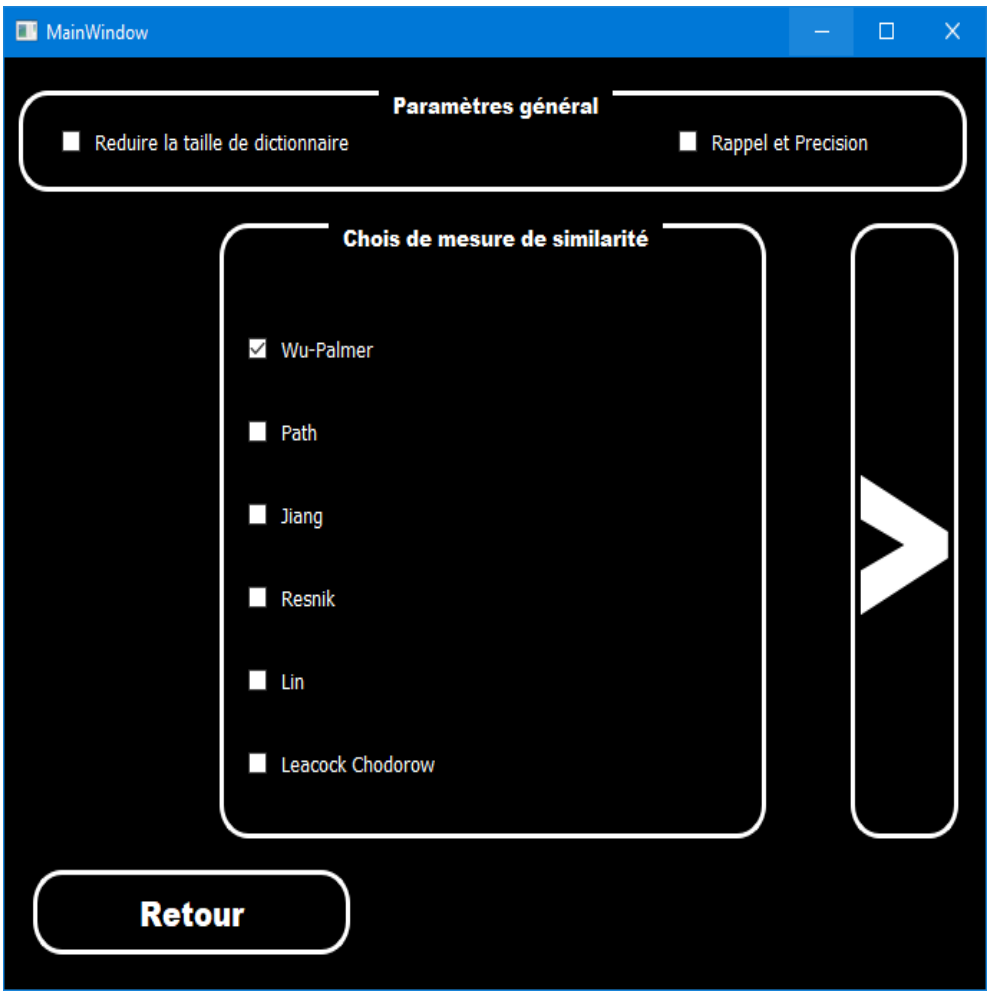

FIGURE 3.24: L'interface de l'approche d'enrichissement (partie 1).
24

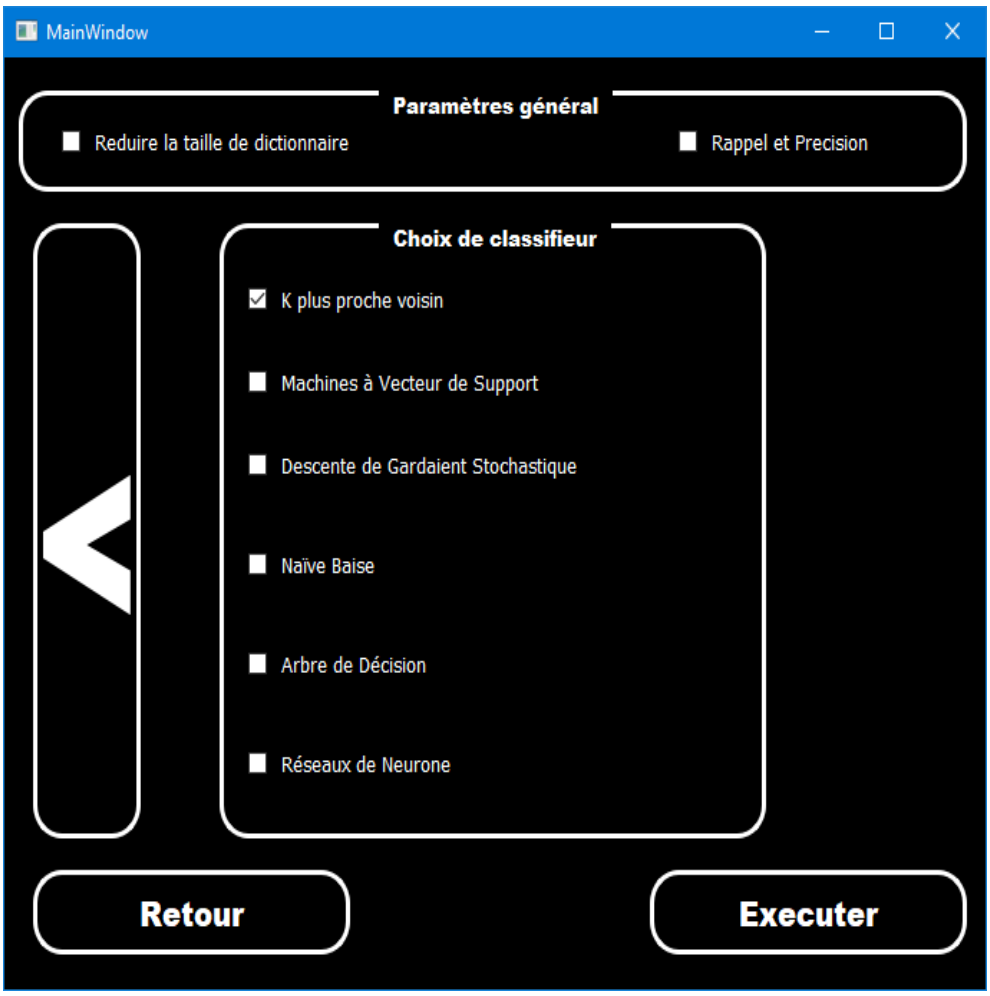

FIGURE 3.25: L'interface de l'approche d'enrichissement (partie 2).
25

A la fin nous cliquons sur le bouton exécuter pour démarrer l'algorithme d'enrichissement et la classification.

3.3.3 Résultat

La figure 3.25 illustre la fenêtre de résultat.

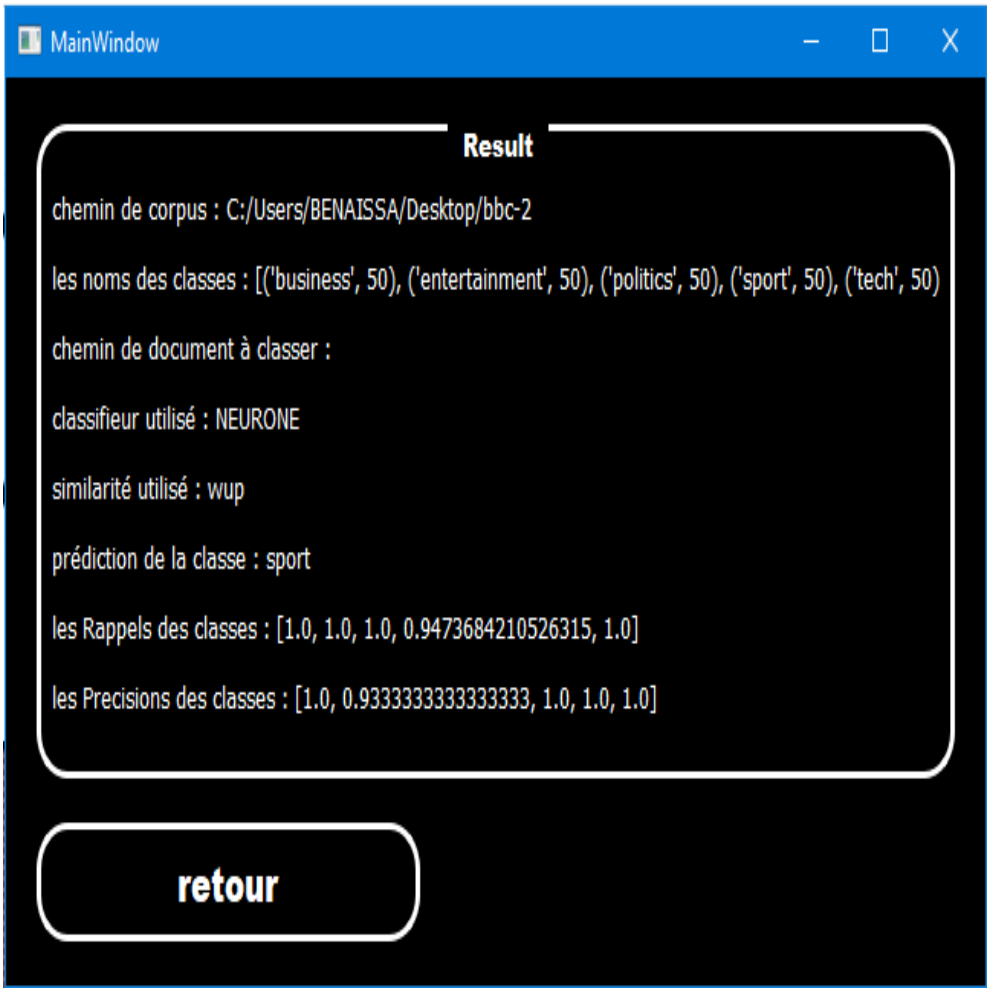

FIGURE 3.26: Exemple d'un résultat.
26

3.4 Les ressources utilisées

3.4.1 Description de corpus utilisé

Nous avons utilisé un corpus qui appartient à la chaine « BBC NEWS », ce corpus contient cinq catégories dont les détails sont présentés dans le tableau suivant :

catégorie	nombre de document
business	510
entertainment	386
politics	417
sport	511
tech	401

TABLE 3.1: les caractéristiques du corpus [27].

3.4.2 Environnement de travail :

Nous avons utilisé l'environnement de travail IDLE (Integrated Development and Learing Environment) avec la version 3.8.1 qui est un environnement par défaut pour le langage Python. Parmi ces avantages :

— Plus rapide en exécution du code python.

— Compiler le code même avec des erreurs.

3.4.3 Python

Python est un langage de programmation interprété, de haut niveau, multi-paradigme et multiplateformes avec une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions. Créée par Guido van Rossum. Ses constructions de langage et son approche orientée objet visent à aider les programmeurs à écrire un code clair et logique pour les projets à petite et grande échelle.

Nous avons utilisé dans notre travaille la version du python 3.8.1.

3.4.4 Bibliothèque pywsd

Pywsd « Python Word Sense Disambiguation » sert à une Désambiguïsation des textes à l'aide des algorithmes et des probabilités pour prédire le sens exact pour chaque mot, nous avons utilisé dans notre travail la dernière version qui est 1.2.4.

3.4.5 WordNet

WordNet est une base de données lexicale et des relations sémantiques entre les mots. WordNet relie les mots à des relations sémantiques, y compris des synonymes, des hyponymes et des méronymes [30], son utilisation principale est l'analyse automatique de texte et les applications d'intelligence artificielle, nous avons utilisé dans notre travail la version 3.1.

3.4.6 QT Designer

C'est un outil Qt pour la conception et la construction d'interfaces utilisateur graphiques (GUI) avec les widgets Qt. Vous pouvez composer et personnaliser vos fenêtres ou boîtes de dialogue à la manière de ce que vous voyez est ce que vous obtenez et les tester en utilisant différents styles et résolutions, l'interface crée est compatible avec python, C++, QML et peut être convertie en code python pour faciliter la construction d'interfaces.

Nous avons utilisé dans notre travail la version QT 5.15.

3.4.7 Scikit learn

C'est une bibliothèque d'apprentissage automatique open source qui prend en charge l'apprentissage supervisé et non supervisé. Il fournit également divers outils pour l'ajustement de modèle, le prétraitement des données, la sélection et l'évaluation de modèle, et de nombreux autres utilitaires.

Nous avons utilisé la version 0.23.1 pour la phase de classification.

3.5 Expérimentations

Afin de pouvoir évaluer les approches proposées, nous avons construit plusieurs échantillons de tailles différentes (50 , 200,500) à partir du corpus « BBC NEWS »et nous avons utilisé la mesure de similarité wu palmer avec le

classificateur K plus proche voisin. Les résultats des expérimentations effectuées sont présentés dans le tableau 3.2.

	Sans enrichissement						Avec enrichissement global						Avec enrichissement local					
La taille du corpus	50		200		500		50		200		500		50		200		500	
	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R
Classe 1 (business)	0.2	1.0	0.8	0.57	0.8	0.33	0.2	1.0	0.8	0.57	0.8	0.33	1.0	1.0	0.8	1.0	0.8	1.0
Classe 2 (entertainment)	1.0	0.4	0.2	1.0	0.2	1.0	0.8	0.36	0.2	1.0	0.2	1.0	0.8	1.0	0.8	0.8	0.6	1.0
Classe 3 (politics)	0.25	1.0	0.4	1.0	0.4	0.66	0.4	1.0	0.4	0.67	0.4	0.66	1.0	0.71	0.8	0.8	0.8	0.8
Classe 4 (sport)	1.0	0.31	1.0	0.67	0.6	0.6	1.0	0.5	1.0	0.41	0.6	0.6	0.6	1.0	0.8	0.8	0.8	0.8
Classe 5 (tech)	0.2	1.0	0.4	0.42	0.8	1.0	0.2	1.0	0.4	1.0	0.8	1.0	0.6	0.5	1.0	0.83	1.0	0.62
Le nombre d'enrichissements effectués							370		197		120		1963		3024		3337	
La mesure f	0.42285		0.53193		0.55856		0.48095		0.53193		0.55856		0.80354		0.83960		0.80162	

Table 3.2 : Expérimentations.

Ces résultats mènent aux constatations suivantes :

- Les résultats d'approches d'enrichissements (local et global) sont nettement meilleurs que ceux de l'approche sans enrichissement.
- Les meilleurs résultats en été obtenues avec l'approche d'enrichissement local.
- L'enrichissement local s'adapte mieux avec des corpus d'apprentissage de petites tailles.
- Plus le nombre de concepts enrichis augmente plus les performances s'améliore.

3.6 Conclusions

Dans ce chapitre nous avons présenté la description et la mise en œuvre des étapes implémentées pour nos approches, à l'aide des mesures de similarité sémantique base sure WordNet et des classificateurs pour catégoriser les textes.

Conclusion générale

Ce mémoire avait pour ambition d'évaluer l'utilisation des mesures de similarité sémantique dans le cadre de l'enrichissement de la représentation conceptuelle dans la catégorisation des textes. Cet enrichissement se base sur la distance sémantique entre les concepts dans le but de fournir une représentation conceptuelle riche permettant aux classificateurs de donner des meilleurs performances.

Pour remédier le problème d'ignorance des concepts au niveau de vecteur conceptuel, nous avons proposé deux approches d'enrichissements, l'enrichissement globale sert à enrichir le vecteur conceptuel et aussi l'enrichissement locale qui se base sur le vecteur conceptuel pour former une nouvelle matrice conceptuelle en utilisant les mesures de similarité sémantiques.

Malheureusement, le temps est court et il a été nécessaire d'ajouter d'autre mesures de similarité sémantique moderne, Notre perspective dans un premier temps est de consolider la démarche implémentée en évaluant sur d'autres collections, puis élargir notre domaine en ajoutant des nouvelles fonctionnalités et aussi de travailler avec une version de "pywsd" plus développé.

Références

[1] R. Jalam, "Apprentissage automatique et catégorisation de textes multilingues," *PhD Tesis, Université Lumiere Lyon*, vol. 2, 2003.

[2] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, p. 1–47, Mar. 2002.

[3] I. Moulinier, *Une approche de la categorisation de textes par l'apprentissage symbolique*. PhD thesis, 1996. Thèse de doctorat dirigée par Ganascia, Jean-Gabriel Sciences et techniques communes Paris 6 1996.

[4] J. Clech and D. A. Zighed, "Une technique de réétiquetage dans un contexte de catégorisation de textes," *Document numérique*, vol. 8, no. 3, pp. 55–69, 2004.

[5] J. Clech, *Contribution méthodologique à la fouille de données complexes*. PhD thesis, Lyon 2, 2004.

[6] D. D. Lewis, "An evaluation of phrasal and clustered representations on a text categorization task," in *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '92, (New York, NY, USA), p. 37–50, Association for Computing Machinery, 1992.

[7] M. F. Porter *et al.*, "An algorithm for suffix stripping.," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[8] H. MATALLAH, *Classification Automatique de Textes Approche Orientée Agent*. PhD thesis, 2011.

[9] E. Miller, D. Shen, J. Liu, and C. Nicholas, "Performance and scalability of a large-scale n-gram based information retrieval system," *Journal of digital information*, vol. 1, no. 5, pp. 1–25, 2000.

[10] M. F. Caropreso, S. Matwin, and F. Sebastiani, "A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization," *Text databases and document management : Theory and practice*, vol. 5478, pp. 78–102, 2001.

[11] R. Jalam and J.-H. Chauchat, "Pourquoi les n-grammes permettent de classer des textes ? recherche de mots-clefs pertinents à l'aide des n-grammes caractéristiques," in *6th International Conference on Textual Data Statistical Analysis, France*, pp. 381–390, 2002.

[12] S. Réhel, "Catégorisation automatique de textes et cooccurrence de mots provenant de documents non étiquetés," 2005.

[13] H. P. Luhn, "A statistical approach to mechanized encoding and searching of literary information," *IBM Journal of research and development*, vol. 1, no. 4, pp. 309–317, 1957.

[14] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," in *Proceedings of the seventh international conference on Information and knowledge management*, pp. 148–155, 1998.

[15] H. Schütze, D. A. Hull, and J. O. Pedersen, "A comparison of classifiers and document representations for the routing problem," in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 229–237, 1995.

[16] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Icml*, vol. 97, p. 35, Nashville, TN, USA, 1997.

[17] S. Jaillet, M. Teisseire, J. Chauche, and V. Prince, "Classification automatique de documents," in *INFORSID*, vol. 3, pp. 87–102, 2003.

[18] F. Z. BENSFIA and A. TERKIA DERDRA, *La Représentation Conceptuelle pour la Catégorisation des Textes Multilingue.* PhD thesis.

[19] C. Touzet, *les réseaux de neurones artificiels, introduction au connexionnisme.* 1992.

[20] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, pp. 1137–1145, Montreal, Canada, 1995.

[21] R. Baeza-Yates, B. Ribeiro-Neto, *et al.*, *Modern information retrieval*, vol. 463. ACM press New York, 1999.

[22] P. Jaccard, "Étude comparative de la distribution florale dans une portion des alpes et des jura," *Bull Soc Vaudoise Sci Nat*, vol. 37, pp. 547–579, 1901.

[23] C. Leacock and M. Chodorow, "Combining local context and wordnet similarity for word sense identification," *WordNet : An electronic lexical database*, vol. 49, no. 2, pp. 265–283, 1998.

[24] Z. Wu and M. Palmer, "Verb semantics and lexical selection," *arXiv preprint cmp-lg/9406033*, 1994.

[25] G. Hirst, D. St-Onge, *et al.*, "Lexical chains as representations of context for the detection and correction of malapropisms," *WordNet : An electronic lexical database*, vol. 305, pp. 305–332, 1998.

[26] T. Slimani, B. BenYaghlane, and K. Mellouli, "Une extension de mesure de similarité entre les concepts d'une ontologie," in *International conference on sciences of electronic, technologies of information and telecommunications*, vol. 69, 2007.

[27] A. N. Ngom, "Étude des mesures de similarité sémantique basées sur les arcs.," in *CORIA*, pp. 535–544, 2015.

[28] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," *arXiv preprint cmp-lg/9511007*, 1995.

[29] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," *arXiv preprint cmp-lg/9709008*, 1997.

[30] C. Fellbaum, "Wordnet," *The encyclopedia of applied linguistics*, 2012.

Résumé

Le domaine du Web sémantique a connu une très forte croissance ces dernières années, pour cela nous avons une problématique liée à l'évaluation d'utiliser des mesures de similarité sémantique dans le cadre de la catégorisation des textes. Le but de ce mémoire est de représenter les documents sous forme d'une représentation conceptuelle, pour la machine devint capable à faire un apprentissage des catégories et d'associer à chaque document non classé sa catégorie en se basant sur la sémantique.

L'implémentation et la conception de notre travaille sont faites avec le langage python 3.8 et Qt Designer et l'expérimentation est effectuée sur un corpus extrait de la chaîne « BBC NEWS ».

Mots clés : Catégorisation des textes, mesure de similarité, représentation conceptuelle, enrichissement.

Abstract

The field of the Semantic Web has experienced very strong growth in recent years, for this we have a problem related to the evaluation of using semantic similarity measures within the framework of the categorization of texts. The purpose of this dissertation is to represent documents in the form of a conceptual representation, for the machine became able to learn categories and associate each unclassified document with its category based on semantics.

The implementation and design of our work are done with the python 3.8 language and Qt Designer and the experimentation is carried out on a corpus extracted from the "BBC NEWS" channel.

Keywords : Categorization of texts, measurement of similarity, conceptual representation, enrichment.

ملخص

شهد مجال الويب الدلالي نمواً قوياًا للغاية في السنوات الأخيرة ، ولهذا لدينا مشكلة تتعلق بتقييم استخدام مقاييس التشابه الدلالي في إطار تصنيف النصوص. الغرض من هذه الرسالة هو تمثيل الوثائق في شكل تمثيل مفاهيمي ، لأن الآلة أصبحت قادرة على تعلم الفئات وربط كل وثيقة غير مصنفة بفئتها على أساس الدلالات.

يتم تنفيذ وتصميم عملنا باستخدام لغة بايثون و يتم إجراء التجربة على مجموعة مستخرجة من قناة أخبار ببسي.

الكلمات الرئيسية : تصنيف النصوص ، قياس التشابه ، التمثيل المفاهيمي ، الإثراء.
