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

République Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique
Université d'Abou bakr Belkaïd – Tlemcen
Faculté des Sciences
Département d'Informatique

MEMOIRE DE PROJET DE FIN D'ETUDES

En vue de l'obtention du diplôme de master en informatique
Spécialité : système d'information et de connaissances

(S.I.C)

Thème :

Évaluation de l'utilisation des relations léxico-sémantiques dans la catégorisation de textes.

Réalisé par :

- **TAHIR Fatima Zohra**
- **BELARBI Hidayat**

Soutenu publiquement le 29 septembre 2024 devant le jury :

Mme.EL YEBDRI Zineb	Présidente
Mme. AMGHAR Djazia	Examinatrice
Mr. BENTALLAH Mohammed Amine	Encadrant

Année Universitaire : 2023 / 2024

Dédicace

“

Je dédie ce mémoire

À mes chers parents ma mère et mon père pour leur soutien inconditionnel et leurs sacrifices qui m'ont permis de réaliser ce travail.

À ma sœur Norhane et mon frère Houssem Eddine

À mes amis pour leurs encouragements constants.

Merci.

”

- Fatima zohra

“

Je dédie ce mémoire

À ma famille, en particulier à ma mère Nezha et à mon père Benali, en reconnaissance de leur soutien, sacrifices et dévouement. Que Dieu leur accorde la santé,le bonheur et la longue vie.

À mes sœurs et mes frères Pour leur dévouement, leur compréhension et leur grande tendresse, merci Pour tous les bons moments passes ensemble.

À tous mes amis et a tous mes camardes, sans oublier mon binôme Tahir Fatima Zohra en souvenir des agréables moments partages et en témoignage de notre amitié.

”

- HIDAYAT

Remerciements

Tout d'abord, Nous remercions dieu le tout puissant de nous avoir donné la force et la patience et la volonté d'entamer et de terminer ce mémoire.

Nous remercions tout particulièrement notre encadrant ***Mr. BENTALLAH Mohammed Amine***, pour sa patience, ses précieux conseils et son soutien tout au long de ce projet académique. Ses orientations et sa disponibilité ont grandement enrichi notre travail et nous ont permis d'avancer avec confiance.

Nos vifs remerciements vont également aux membres du jury ***Mme. EL YEBDRI Zineb*** et **Mme.AMGHAR Djazia** pour l'intérêt qu'elles ont porté à notre recherche en acceptant d'examiner notre modeste travail.

Nos remerciements s'adressent également à tous les membres du département d'informatique, enseignants et administratifs, pour leurs générosités et la grande patience dont ils ont su faire preuve malgré leurs charges académiques et professionnelles.

Pour finir, nous souhaitons remercier toute personne ayant contribué de prés ou de loin à la réalisation de ce travail.

Table des matières

Table des figures

Liste des tableaux

Liste des algorithmes

Liste des sigles et acronymes

CT	*catégorisation de textes*
Depth	*profondeur d'une structure ou d'un arbre*
Fn	*faux négatif*
Fp	*faux positif*
JWNL	*Java WordNet Library*
KNN	*K-Nearest Neighbor*
P	*précision*
R	*rappel*
SVM	*machine à vecteur supports*
TF	*Term Frequency*
TF-IDF	*Term Frequency Inverse Document Frequency*
TFC	*Term Frequency Collection*
UKB	*Unsupervised Knowledge Based méthode de desambiguisation*
Vn	*vrai négatif*
Vp	*vrai positif*
WN	*WordNet*

Introduction générale

Dans le monde actuel, l'explosion des données textuelles numériques a engendré un besoin croissant de techniques efficaces pour extraire des informations pertinents et structurées à partir de vastes volumes de textes. Ce besoin a conduit au développement de la fouille de textes, un domaine interdisciplinaire combinant la linguistique, l'informatique et les statistiques pour analyser et interpréter des données textuelles, ainsi que la gestion des corpus textuels volumineux et non structurés.

Les données textuelles proviennent de diverses sources telles que les réseaux sociaux, les articles scientifiques, les forums en ligne et les rapports d'entreprises. La diversité et la complexité de ces textes exigent le développement de techniques avancées pour automatiser leur analyse. C'est dans ce contexte que s'inscrit la catégorisation de textes, une tâche essentielle de la fouille de textes.

La catégorisation de textes consiste à assigner des catégories prédéfinies à des documents en fonction de leur contenu. Cette tâche est importante pour organiser et structurer des masses de données textuelles, facilitant ainsi la recherche et la récupération d'information.

Cependant, la représentation de textes constitue une étape critique dans ce processus, et la méthode de représentation la plus couramment utilisée, le 'sac de mots', présente des limitations notables. Cette approche ignore les relations entre les mots et ne prend pas en compte les spécificités linguistiques telles que les variations orthographiques, grammaticales et morphologiques. Par exemple, la graphie d'un mot peut varier ('analyse' et 'analyze'), un mot peut avoir différentes catégories grammaticales ('port' en tant que nom ou verbe), et les variations morphologiques peuvent modifier la forme d'un mot ('enfant', 'enfants'). De plus, la composition lexicale complexe pose un défi supplémentaire pour cette méthode.

Face à ces limites, plusieurs travaux récents se sont concentrés sur l'exploitation des relations lexico-sémantiques, telles que l'homonymie, la meronymie, l'hyperonymie et l'hyponymie, afin d'améliorer les méthodes de catégorisation de textes, d'enrichir la représentation des documents et d'affiner les algorithmes de classification, notamment en renforçant la compréhension des connexions entre les concepts.

Notre mémoire vise à explorer en profondeur l'évaluation de l'utilisation des relations lexico-sémantiques dans la catégorisation de textes. L'objectif principales est d'intégrer ces relations dans les documents classés et le document à classer, et de comparer les résultats obtenus en utilisant les méthodes de classification.

Notre projet est structuré en trois chapitres distincts. Le premier chapitre s'articule autour de la catégorisation de textes, il débute par une présentation détaillée du processus de catégorisation en exposant ses différentes phases et les applications qui en découlent.

Le deuxième chapitre se concentre sur les différentes relations lexico-sémantiques, en mettant en lumière les principes fondamentaux des ontologies ainsi que la structure et l'organisation de WordNet .Le troisième chapitre consiste à exposer les approches implémentées ainsi que la représentation des résultats obtenus.

Chapitre 1

La catégorisation de textes

1.1 Introduction

On assiste aujourd'hui à un accroissement de la quantité d'information textuelle disponible et accessible d'une manière exponentielle. D'après les derniers chiffres, on parle de plus de 200 millions de serveurs hôtes sur Internet et plus de 3 milliards de pages, la taille des corpus tests utilisés est passée de quelques mégaoctets à plusieurs Giga-octets. De combien de temps a besoin un humain pour associer un texte à une catégorie ? En pratique, il s'agit d'une question difficile à répondre. Nous pouvons résumer les contraintes majeures qui s'opposent au traitement manuel de classification des documents textuels dans les trois points suivants :

- La réalisation manuelle de cette tâche par un expert est extrêmement coûteuse en termes de temps et personnel car il s'agit de lire attentivement chaque texte, au vu de la quantité phénoménale de textes aujourd'hui accessibles (par le biais du réseau Internet en particulier) [2][7]

- Les traitements manuels sont peu flexibles et leur généralisation à d'autres domaines est quasi impossible ; c'est pourquoi on cherche à mettre au point des méthodes automatiques [2][7]

- Cette opération peut être perçue comme subjective puisque basée sur l'interprétation du document, deux experts peuvent classer différemment un même document, ou encore un même expert peut classer différemment un même document soumis à deux instants différents [8]

C'est pour cette raison que nous adoptons une nouvelle approche de classification. Dans ce chapitre, nous présentons la catégorisation automatique des textes, en définissant le concept ainsi que les différentes étapes du processus, les méthodes utilisées et les critères d'évaluation. Nous abordons ensuite les applications et les problèmes rencontrés.

1.2 Définition de la catégorisation de textes

La catégorisation automatique de textes sert à associer chaque document à une ou plusieurs classes (catégories) prédéfinies avec l'analyse de contenu et la signification de texte. Elle va permettre de faciliter les recherches par filtre, organiser de grandes quantités de documents textuels et analyser les sentiments.

La classification de textes (C.T) consiste à trouver construire un lien fonctionnel entre un ensemble de texte et un ensemble de catégories (étiquettes, classes).[1]

Mr SEBASTIANI a défini la catégorisation de textes comme un processus qui consiste à associer une valeur booléenne à chaque paire $(d_j, c_i)_{DC}$, où D est l'ensemble des textes et C est l'ensemble des catégories. La valeur V (Vrai) est alors associée au couple (d_j, c_i) si le texte d_j appartient à la classe c_i, tandis que la valeur F (Faux) est associée dans le cas contraire.

Le but de la catégorisation des textes est de construire une procédure (modèle, classificateur) notée :

$$\Phi : DC \rightarrow \{V, F\} \tag{1.1}$$

qui associe une ou plusieurs étiquettes (catégories) à un document d_j

1.3 Processus de la catégorisation de textes

Le processus de catégorisation de textes intègre la construction d'un modèle de prédiction qui en entrée reçoit un texte et en sortie lui associe une ou plusieurs étiquettes. [1]

La figure 1.1 représente le processus de catégorisation de texte qui comporte un ensemble d'étapes comme suit :

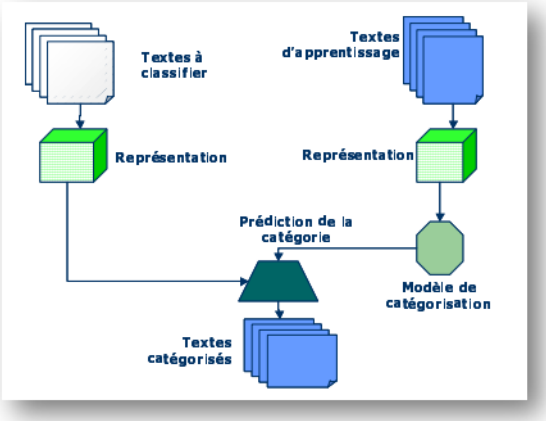

Figure 1.1 : processus de la catégorisation de texte [1]

1.3.1 Représentation de textes

Un codage préalable du textes est nécessaire, comme pour l'image, le son, etc., [Sebastiani, 2002], car il n'existe pas actuellement de méthode d'apprentissage capable de traiter directement des données non-structurées, ni dans la phase de construction du modèle, ni lors de son utilisation en classement. Pour la majorité des méthodes d'apprentissage, il faut transformer l'ensemble des textes en un tableau croisé "individus-variables" :

- **L'individu** est un texte (un document) dj, étiqueté lors de la phase d'apprentissage, et à classer dans la phase de prédiction. – Les variables sont les descripteurs (les termes) tk qui sont extraits des données textuelles.
- **Les variables** sont les descripteurs (les termes) tk qui sont extraits de données textuelles. [1]

L'utilisation des techniques comme TF, TF-IDF, TFC pour mesurer l'importance des mots dans les documents.

1.3.2 Le prétraitement

Avant l'encodage vectoriel, Il est nécessaire d'effectuer au préalable du codage d'un document dans un espace de mots une transformation permettant le passage de l'espace du caractère à un espace de mots. Cette transformation est appelée prétraitement.

- **La Segmentation** : La phase de segmentation consiste à découper la séquence des caractères afin de regrouper les caractères formant un même mot. Une segmentation classique consiste à découper les séquences de caractères en fonction de la présence ou l'absence de caractères de séparation (de type « espace », « tabulation » ou « retour à la ligne »).

- · **Le traitement morphologique** : Le traitement morphologique consiste à effectuer un traitement au niveau de chacun des mots afin de regrouper un ensemble de mots significativement identiques.

    1. Suppression de mots vides : Certains mots comme les prépositions, conjonctions ou articles, qui ne contiennent aucune information sémantique, qui ne modifie pas le sens des mots qui les accompagnent, ont pu être retirés des documents. Ce prétraitement, permettant de réduire la taille du lexique,

    2. Le stemming : Consiste à regrouper sous un même identifiant des mots dont la racine est commune.

    3. Lemmatisation : La lemmatisation consiste à regrouper des mots dont la signification est la même alors même que leurs racines sont différentes (par exemple, « maison » et « baraque »). C'est une tâche plus complexe que le stemming et qui repose habituellement sur l'utilisation de grandes bases de connaissances. **Exemple** : le lemme d'un mot au pluriel sera sa forme au singulier, celui d'un verbe conjugué sera sa forme infinitive.[6]

1.3.3 Choix de classifieur

L'objectif de cette phase est de choisir une technique d'apprentissage (classifieur). Parmi les méthodes les plus utilisées l'analyse factorielle discriminante, la régression logistique, les réseaux de neurones, les arbres de décision, les plus proches voisins, les réseaux bayésiens, SVM (machines à vecteurs supports), et les méthodes de Boosting .

1.4 Les méthodes de catégorisation de textes

1.4.1 Machine à vecteur support

Voici un problème de classification : pour chaque nouvelle entrée, être capable de déterminer à quelle catégorie cette entrée appartient.

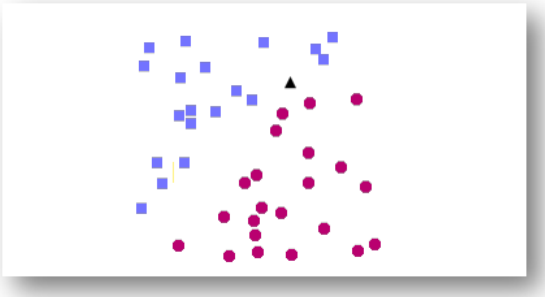

Figure 1.2 : L'objet à classer (le triangle noir) est-il un rond rouge ou bien un carré bleu ? [3]

Le SVM est une solution à ce problème de classification. Le SVM appartient à la catégorie des classificateurs linéaires (qui utilisent une séparation linéaire des données), et qui dispose de sa méthode à lui pour trouver la frontière entre les catégories.

Pour que le SVM puisse trouver cette frontière, il est nécessaire de lui donner des **données d'entraînement**. En l'occurrence, on donne au SVM un ensemble de points, dont on sait déjà si ce sont des carrés rouges ou des ronds bleus, comme dans la Figure 1.3. A partir de ces données, le SVM va estimer l'emplacement le plus plausible de la frontière.

Le SVM est maintenant capable de prédire à quelle catégorie appartient une entrée qu'il n'avait jamais vue avant.

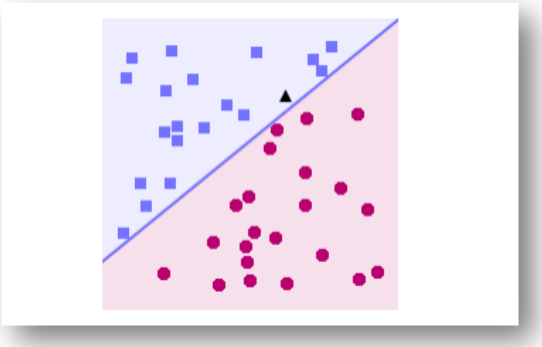

Figure 1.3 : Notre SVM, muni des données d'entraînement (les carrés bleus et les ronds rouges déjà indiqués comme tels par l'utilisateur), a tranché : le triangle noir est en fait un carré bleu.[3]

1.4.2 K plus proches voisins

L'algorithme des k-plus proches voisins (KNN) est une méthode d'apprentissage à base d'instances. Il ne comporte pas de phase d'apprentissage en tant que telle. Les documents faisant partie de l'ensemble d'apprentissage sont seulement enregistrés. Lorsqu'un nouveau document à classer arrive, il est comparé aux documents d'apprentissage à l'aide d'une mesure de similarité. Ses k plus proches voisins sont alors considérés : on observe leur catégorie et celle qui revient le plus parmi les voisins est affectée au document à classer. La méthode utilise donc deux paramètres : le nombre k et la fonction de similarité.

L'une des mesures de similarité la plus utilisée est le cosinus qui consiste e à quantifier la similarité entre deux documents en calculant le cosinus de l'angle entre leurs vecteurs.[4]

1.4.3 Les arbres de décision

La méthode des arbres de décision est une méthode d'apprentissage supervisée dont le but est de calculer automatiquement les valeurs de la variable endogène (à prédire), fixée à priori, à partir d'autres informations (variables exogènes ou prédictives). Le principe des arbres de décision repose sur un partitionnement récursif des données.

Le but du partitionnement est d'obtenir des groupes homogènes du point de vue de la variable à prédire. Le résultat est un enchaînement hiérarchique de règles. Un chemin, partant de la racine jusqu'à une feuille de l'arbre, constitue une règle d'affectation du type « Si condition Alors conclusion ». L'ensemble de ces règles constitue le modèle de prédiction.[1][7]

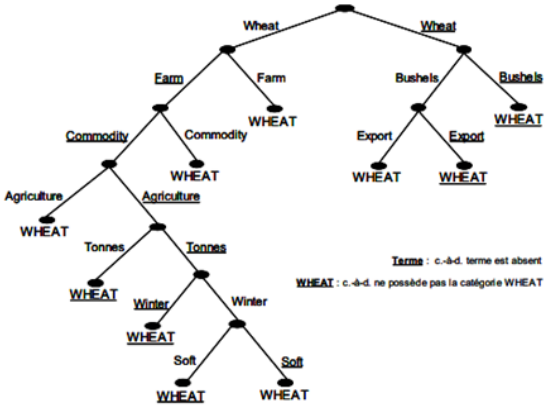

Figure 1.4 : Exemple d'arbre de décision appliquée sur le corpus Reuters. 'WHEAT' correspond au concept le document possède la catégorie 'WHEAT' [1]

1.4.4 Naïve Bayes

La classification naïve bayésienne est un type de classification probabiliste qui utilise le théorème de Bayes pour prédire la classe d'une observation en fonction des caractéristiques observées. L'algorithme suppose que chaque caractéristique est conditionnellement

indépendante des autres, ce qui simplifie considérablement les calculs nécessaires à la classification.[30][31]

Fonctionnement :

1. **Entraînement :** Le classificateur est entraîné sur un ensemble de données où les classes sont déjà connues. Il calcule les probabilités a priori des classes ainsi que les probabilités conditionnelles des caractéristiques données chaque classe.

2. Prédiction : Pour classifier une nouvelle observation, le classificateur utilise les probabilités calculées pour déterminer la classe ayant la probabilité a posteriori la plus élevée, en appliquant la formule du théorème de Bayes.[32]

3. **Hypothèse d'indépendance :** L'hypothèse clé est que la présence ou l'absence d'une caractéristique n'affecte pas la présence d'une autre caractéristique. Par exemple, dans le cas de la classification de fruits, un fruit pourrait être classé comme une pomme en se basant indépendamment sur ses caractéristiques telles que la couleur et la taille

Avantages et applications :

Rapidité et efficacité : Le classificateur naïf de Bayes est rapide à entraîner et à exécuter, nécessitant relativement peu de données d'entraînement pour estimer les paramètres nécessaires. [32]

Applications variées : Il est couramment utilisé dans des domaines tels que le filtrage anti-spam, la classification de documents et l'analyse de sentiments

1.4.5 Les réseaux de neurones

C'est une structure constituée de suite successive de couches de nœuds et qui permet de définir une fonction de transformation non linéaire des vecteurs d'entrées (composés dans le cas de classification des mots pondérés de leur poids) en vecteur de catégories. La disposition des neurones dans le réseau ainsi que le nombre de couches utilisées ont une influence sur le résultat de 27 classification. Comparés aux autres méthodes de classification par apprentissage supervisé, les réseaux de neurones ont l'inconvénient que le coût d'apprentissage est assez élevé. [9]

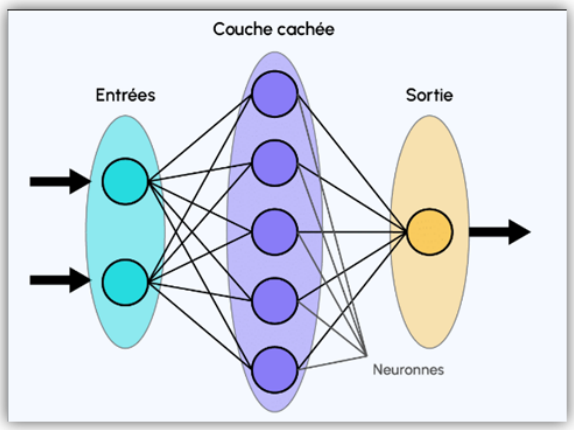

Figure 1.5 : architecture d'un réseau de neurones artificiels [14]

1.5 Comparaison des méthodes de classification

Le tableau si dessous représente une comparaison des différentes méthodes de classification : Naïve Bayes, SVM , KNN ,arbres de décision et réseaux de neurones.

Méthode	Avantages	Inconvénients	Applications
Naïve Bayes	- Rapide et efficace - Fonctionne bien avec peu de données - Bon pour la classification de texte	- Hypothèse d'indépendance des caractéristiques souvent violée - Moins performant sur des données complexes	Filtrage anti-spam, classification de documents
SVM [33]	- Efficace pour les grandes dimensions - Gère bien les marges entre classes - Performant avec des données non linéaires (avec noyaux)	- Sensible aux choix de paramètres et à la mise à l'échelle des données - Peut être lent avec de grands ensembles de données	Classification d'images, détection de fraudes
KNN [33]	- Simple à comprendre et à mettre en œuvre - Pas besoin d'entraînement explicite	- Lenteur lors des prédictions sur grands ensembles - Sensible au bruit dans les données	Recommandations, reconnaissance de formes
Réseaux de Neurones [9]	- Capacité d'apprentissage complexe et non linéaire - Excellente performance sur des jeux de données volumineux	- Nécessite beaucoup de données pour un bon entraînement - Temps d'entraînement élevé et risque de surajustement	Vision par ordinateur, traitement du langage naturel
Arbres de Décision [1]	- Interprétable et visuel - Gère bien les données catégorielles et continues	- Risque de surajustement si non élagué - Peut être instable avec des variations dans les données	Classification médicale, analyse de risque

Tableau 1.1 : Comparaison des différentes méthodes de classification

1.6 Évaluation de processus

1.6.1 Les critères d'évaluation

Nous considérons ici un problème simple de classification pour lequel nous nous intéressons à une classe unique C et nous voulons évaluer un système qui nous indique si un document peut être associé ou non à cette classe C. Ce problème est un problème de classification à deux classes (C et non C). Si on peut maîtriser ce problème simple, on

pourra fusionner par la suite, les mesures de performance de plusieurs systèmes bi-classes afin d'obtenir une mesure de la performance d'un classifieur multi-classes.

- **Matrice de contingence :**
  - Vrai Positif (**VP**) : Documents attribués à leurs vraies catégories.
  - Faux Positif (**FP**) : Documents attribués à des mauvaises catégories.
  - Faux Négatif (**FN**) : Le nombre de documents inconvenablement non attribués.
  - Vrai Négatif (**VN**) : Le nombre de documents non attribués à une catégorie convenablement.
- **Rappel :**

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

$$\text{Rappel (R)} = \frac{\text{Vp}}{\text{Vp} + \text{Fn}} \tag{1.2}$$

- **Précision :**

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

$$\text{Précision (P)} = \frac{\text{Vp}}{\text{Vp} + \text{Fp}} \tag{1.3}$$

- **F-mesure :**

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

$$\text{F-mesure} = \frac{2PR}{P + R} \tag{1.4}$$

1.6.2 Corpus d'évaluation

- **Reuters :[11]** Reuters est un corpus de dépêches en langue anglaise qui a été proposé par l'agence de presse Reuters en 1987. Il correspond à une problématique de classification en plusieurs classes (un document appartient à une ou plusieurs classes). Deux qualités principales caractérisent les documents de Reuters c'est qu'ils sont courts et plutôt homogènes avec un vocabulaire riche (environ 17000 mots).[5]
- **20 Newsgroups :[12]** L'ensemble de données 20 newsgroups comprend environ 18 000 messages de newsgroups sur 20 sujets répartis en deux sous-ensembles : l'un pour la formation (ou le développement) et l'autre pour le test (ou l'évaluation des performances). La répartition entre l'ensemble de formation et l'ensemble de test est basée sur les messages postés avant et après une date spécifique.
- **SpamAssassin :[13]** L'ensemble de données Spam Assassin est une sélection de messages électroniques qui peuvent être utilisés pour tester les systèmes de filtrage du spam. Cet ensemble particulier a été obtenu à partir des Apache Public Datasets, nettoyé et organisé dans un fichier csv d'une manière qui peut être pratique à utiliser.

1.7 Les applications de la catégorisation de textes

La catégorisation de textes peut être un support pour différentes applications parmi lesquelles [1] :

- Le filtrage qui consiste à déterminer si un document est pertinent ou non (décision binaire) par exemple la détection de spams (les courriers indésirables) pour ensuite les supprimer.
- Le routage qui permet d'affecter un document à une ou plusieurs catégories parmi n, par exemple la diffusion sélective d'information.
- L'identification de la langue
- La reconnaissance d'écrivains
- La catégorisation de documents multimédia.
- L'attribution de la paternité (c'est-à-dire l'identification automatique de l'auteur d'un texte parmi un ensemble prédéfini).
- Classification des pages Web (c'est-à-dire le regroupement des pages Web [ou des sites] selon les schémas de classification taxonomique typiques des portails Web.
- Le codage d'enquête (c'est-à-dire classification des répondants à une enquête en fonction des réponses textuelles qu'ils ont renvoyées à une question ouverte).

1.8 Les problèmes de la catégorisation de textes

Beaucoup difficultés peuvent s'opposer au processus de catégorisation de textes. Des problèmes connus dans la discipline liée à l'apprentissage automatique supervisé comme la subjectivité de la décision prise par les experts, le sur-apprentissage, etc.. Mais aussi des problèmes particuliers liés à la nature des données traitées à savoir des données textuelles comme la polysémie, la redondance, Les variations morphologiques ou même L'homographie, etc..

Nous allons signaler quelques problèmes principaux dans ce qui suit [5] :

1.8.1 Redondance (Synonymie)

La redondance et la synonymie permettent d'exprimer le même concept par des expressions différentes, plusieurs façons d'exprimer la même chose.

Cette difficulté est liée à la nature des documents traités exprimés en langage naturel contrairement aux données numériques.[10]

1.8.2 Polysémie (Ambiguïté)

Un mot possède, dans différents cas, plus d'un sens et plusieurs définitions lui sont associées. Par conséquent, à cause de la polysémie, les mots seuls sont parfois de mauvais descripteurs.

1.8.3 La graphie

Un terme peut comporter des fautes d'orthographe ou de frappe comme il peut s'écrire de plusieurs manières ou s'écrire avec une majuscule. Ce qui va peser sur la qualité des résultats.

1.8.4 Les mots composés

La non prise en charge des mots composés comme : Arc-en-ciel, peut-être, etc.. Dont le nombre est très important dans toutes les langues.

1.8.5 Subjectivité de la décision

Parmi les problèmes classiques usuels dans le domaine de l'apprentissage supervisé c'est la subjectivité de la décision prise par les experts qui décident de la classe à laquelle le texte va être attribué , un même document peut être classé différemment par deux experts.[8]

1.9 Conclusion

Dans ce chapitre nous avons présenté quelques techniques importantes dans le domaine du traitement du langage naturel (NLP) qui consiste à étiqueter chaque texte avec une ou plusieurs classes. Cette pratique est essentielle pour faciliter la recherche, l'organisation et l'analyse de grands volumes de données textuelles, Nous avons également introduit les différents moyens d'évaluation d'un classificateur.

Les applications de la classification des textes sont diverses et incluent l'analyse de sentiments, la détection de spams, l'identification de la langue, le routage et bien d'autres. Les méthodes de classification peuvent être basées sur des modèles de Machine Learning, des algorithmes de classification probabilistes ou des méthodes déductives. Les modèles de classification sont entraînés avec des données de texte spécifiques pour pouvoir étiqueter de nouveaux textes en fonction des classes prédéfinies.

La classification des textes est utilisée dans de nombreuses industries pour extraire des informations pertinentes, détecter des tendances et prendre des décisions éclairées

Chapitre 2

Les relations lexico-sémantiques

2.1 Introduction

Dans un monde où la quantité d'informations textuelles croit de manière exponentielle, il devient impératif de développer des outils capables de les exploiter en facilitant la recherche, la consultation et l'organisation de ces données.

Les ontologies en général et Wordnet en particulier sont des piliers dans ce domaine jouant un rôle essentiel dans la structuration des connaissances. Ces outils permettent de mettre en évidence les différentes relations lexico-sémantiques, ce qui permet d'accéder rapidement à des connaissances structurées pour comprendre un contexte donné.

Ce chapitre explore en profondeur les concepts d'ontologies et de Wordnet en mettant en lumière leur rôle dans le traitement des relations lexico-sémantiques. Nous abordons d'abord les principes fondamentaux des ontologies, ensuite nous examinons la structure et l'organisation de Wordnet ,et en dernier, nous verrons les différentes relations lexico-sémantiques qui transforment la manière dont nous interagissons avec les informations textuelles.

2.2 Les ontologies

2.2.1 Définition

Une ontologie est une représentation partagée, consensuelle, et sous format compréhensible d'un domaine donné. Elle caractérise un moyen pour fournir un vocabulaire spécifique à ce domaine. Par ailleurs, elle permet d'offrir tous les moyens aidant à comprendre les constituants d'un domaine. En effet, elle facilite la modélisation du monde réel à travers des formalismes et des langages de représentation très spécifiques.[15]

Les définitions les plus populaires et originales de l'ontologie sont celles énoncé par Gruber qui la définie comme une "*spécification explicite d'une conceptualisation*" et celle de Borst qui la défini comme "*une spécification formelle d'une conceptualisation partagée*". En 1998, Studer et Al ont fusionné dans les deux définitions pour définir l'ontologie comme une "*spécification formelle et explicite d'une conceptualisation partagée*". Ces définitions se basent sur les quatre notions suivantes :

- Formelle : Description de l'ontologie par un langage compréhensible par la machine en excluant le langage naturel.
- Explicite : les concepts avec leurs contraintes d'utilisation doivent être explicitement définis.
- Conceptualisation : le modèle abstrait d'un phénomène du monde réel par identification des concepts clefs de ce phénomène.
- Partagée : l'ontologie n'est pas la propriété d'un individu, mais elle représente un consensus accepté par une communauté d'utilisateurs.[16]

2.2.2 Les constituants d'une ontologie

La formalisation d'une ontologie se base sur cinq notions comme suit : les concepts, les relations, les fonctions, les instances et les axiomes [18].

- Concepts : un concept représente un ensemble d'individus qui ont des caractéristiques en commun. Il est composé de trois parties [17] : Un ou plusieurs termes permettent de désigner le concept pour qu'il soit reconnu de façon non ambigüe par la machine. La notion est appelée intention elle est définie à travers ses propriétés et sers attributs. L'ensemble d'objets définis par le concept, appelé extension du concept.
- Relations : la relation est un type d'interaction entre les concepts. Les relations les plus courantes sont : les relations taxonomiques qui sont des relations binaires entre un concept générique et un autre plus spécifiques et les relations associatives qui sont des relations d'interaction non-taxonomique.
- Les fonctions : Elles constituent des cas particuliers de relation dans laquelle un élément de la relation, le nième, est défini en fonction des n-1 éléments précédents.
- Les instances : les instances sont les éléments décrit par le concept. Une ontologie contenant des instances devient alors une base de connaissances.
- Les axiomes : Les axiomes sont des expressions qui sont toujours vraies. Ils sont utilisés pour définir la signification des composants, restrictions sur la valeur des attributs, ou vérifier la validité des informations.[17]

2.2.3 Classification d'une ontologie

Plusieurs critères de classification ont été proposés pour catégoriser une ontologie. Parmi ces classifications, on trouve selon la structure utilisée dans la conceptualisation ou le sujet de conceptualisation.

En se basant sur la structure de conceptualisation, une ontologie peut être :

- Ontologie terminologique spécifiant les termes utilisés pour spécifier les connaissances (lexique, glossaires, ...etc).
- Ontologie d'informations structurant les termes (schéma d'une base de données).
- Ontologie de connaissances spécifiant la conceptualisation des connaissances. [16]

En se basant sur le sujet de conceptualisation, une ontologie peut être :

- Ontologie d'application qui est la plus spécifiques. Elle mettant en jeu des concepts spécifiques à un domaine et à une activité particulière.
- Ontologie de domaine décrivant les connaissances d'un domaine particulier. Les concepts et les relations de cette ontologie renvoient à des objets du domaine.

- Ontologie générique écrivant des concepts très généraux qui sont indépendants d'un domaine ou d'un problème particulier.
- Ontologie pour la représentation des connaissances décrivant des notions utilisées dans toutes les ontologies pour spécifier les connaissances.
- Ontologie de tâche décrivant des connaissances liées à une tâche ou une activité particulière. À l'instar des ontologies de domaine.[19]

Uschold et Gruninger proposent une autre classification des ontologies qui se basent sur leurs niveaux de formalisation. Ainsi, une ontologie peut être :

- Informelle : si elle est exprimée en langage naturel. L'inconvénient majeur de cette ontologie réside dans l'absence de formalisation qui rend difficile l'opérationnalisation.
- Semi-formelle : si elle est exprimée par un langage formel artificiel. C'est le cas de la version Onto lingua de l'ontologie Entreprise.
- Formelle : si elle est exprimée par un langage formel artificiel dont les termes sont méticuleusement définis par la sémantique formelle.[16]

2.3 Wordnet

2.3.1 Présentation de Wordnet

Wordnet (Miller,1995) est une base de données lexicales conçue par des linguistes de l'université Princeton à partir de 1985.[20]

Son objectif est de classifier et de mettre en relation le contenu sémantique et lexicale de l'anglais. Plusieurs ressources linguistiques ont été développé en relation avec wordnet pour former un ensemble complet englobant les aspects lexicaux, syntaxiques et sémantiques.

Ces ressources fournissent une base solide pour des avancées dans des domaines tels que la recherche d'information, la compréhension automatique de texte, et la désambiguïsation lexicale.

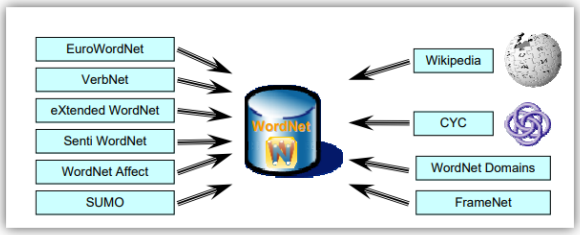

Figure 2.1 : Ressources disposant d'une traçabilité vers WordNet (liste non exhaustive) [20]

2.3.2 Conception et structure de wordnet

La conception de wordnet repose sur des théories de la représentation des connaissances mentales, organisant les mots et concepts de manière hiérarchique à travers la relation d'inclusion. Ces ensembles sont appelés "Synsets".

WordNet veut représenter la structure du lexique en regroupant les noms, les verbes, les adjectifs et les adverbes en synsets et en reliant les synsets à d'autres synsets en fonction de leurs liens lexicaux ou sémantico-conceptuels.[21]

1. **Synset**
   Un synset est un groupe de mots ou de phrases exprimant le même concept. Chaque synset est accompagné d'une description du sens qu'il présente "gloss" et d'exemples d'usage.

2. **Organisation**
   Wordnet classe les mots de la langue anglaise en quatre catégories syntaxiques principales : les noms, les verbes, les adjectives et les adverbes. Les noms et les verbes sont organisés en hiérarchies qui sont structurés en type de base au niveau de la racine. Les relations d'hyperonymie et d'hyponymie relient les concepts généraux aux concepts spécifiques des noms et des verbes. Les adjectifs et les adverbes dans WordNet sont organisés en paires d'antonymes. De nombreux adjectifs en anglais ont des antonymes, contrairement aux verbes et aux noms. Les adverbes sont souvent définis par les adjectifs dont ils dérivent, héritant ainsi de leur structure. [22]

3. **Relations**
   Dans WordNet, deux relations fondamentales sont à l'œuvre : les relations lexicales, qui concernent les formes de mots (entre lemmes) par exemple, la synonymie, et les relations sémantiques, qui associent les significations des mots (entre synsets) par exemple, l'hyperonymie.

Relation	Description	Exemple
Hypernym	Is a generalzation of	Fourniture is a hypernym of chai
Hyponym	Is a kind of	Chair is a hyponym of furniture
Troponym	Is a way to	Amble is a troponym of walk
Meronym	Is part/substance/member of	Wheel is a (part) meronym of a bicycle
Holonym	Contains part	Bicycle is a holonym of a wheel
Antonym	Opposite of	Ascend is an opposite of descend
attribute	Attribute of	Heavy is a attribute of weight
Entailment	entails	Ploughing entails digging
Cause	Cause to	To offend causes to resent
Also see	Related verb	To lodge is related to reside
Similar to	Similar to	Dead is similar to assassinated
Participale of	Is participle of	Stored (adj) is the participle of ”to store”
Pertainym of	Pertains to	Radial pertains to radius

Tableau 2.1 : Quelques relations dans wordnet [22]

2.3.3 l’organisation des résultats de Wordnet[23]

- Wordnet contient des unités de base
  - Composés
  - Verbes à phrase
  - Collocations et phrases idiomatiques
- Wordnet en tant que dictionnaire
  - Donne des définitions
  - Exemples de phrases
  - Contient des ensembles de synonymes
- Wordnet comme thésaurus
  - Niveau conceptuel : relations conceptuelles sémantiques
  - Niveau lexical : relations lexicales
- Sens lexical
  - Wordnet utilise deux moyens pour définir le sens d’un mot :
  - Les synsets
  - Les relations lexicales

2.3.4 les limites de Wordnet

Bien que Wordnet soit une ressource linguistique puissante, elle présente certaines limites, voici quelques-unes :

- Manque d'information : Wordnet ne comprend pas d'information sur l'étymologie et la prononciation des mots.
- Relations limitées : Wordnet ne représente pas adéquatement certaines relations sémantiques plus contextuelles comme les relations pragmatiques ou les nuances de connotation, par exemple : la relation pragmatique entre (SOAP#1 / BATH#2) est absente de wordnet.
- Ambiguïté sémantique : Wordnet peut parfois contribuer à l'ambiguïté en fournissant trop de sens pour un mot donné, ce qui complique la tâche de la désambiguïsation lexicales.

2.4 Les relations lexico-sémantiques

Deux relations fondamentales articulent le lexique d'une langue, celle entre des formes des mots (lemmes) appelés relations lexicales, et celle autour des significations de ces mots (sens) appelés relations sémantiques.

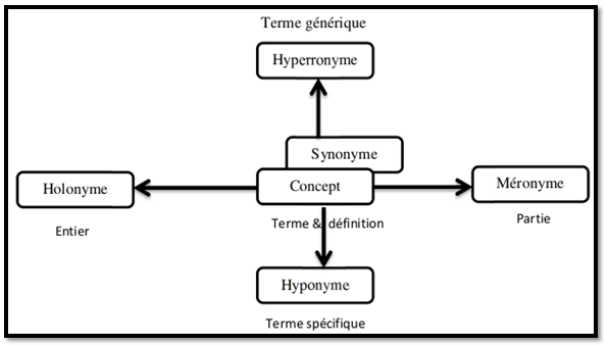

Figure 2.2 : Principales relations sémantiques dans Wordnet.[24]

2.5 Relations d'hiérarchie et d'inclusion

2.5.1 L'hyperonymie

L'hyperonymie est la relation sémantique hiérarchique d'un lexème à un autre selon Laquelle l'extension du premier terme, plus général, englobe l'extension du second, plus spécifique. Le premier terme est dit hyperonyme de l'autre, ou super ordonné par rapport à l'autre.[24]

2.5.2 L'hyponymie

L'hyponymie est une relation d'inclusion entre deux mots dont l'un est l'hyponyme de l'autre. La relation d'hyponymie est l'expression linguistique de la relation logique d'inclusion d'une classe dans une autre « on dit qu'un concept représenté par le synset {x, x'…} est l'hyponyme du concept représenté par le synset {y, y'…} si les locuteurs dont l'anglais est la langue maternelle acceptent les phrases du type Un x est une sorte de y. ».[24]

Exemple :
Chat (hyponyme) > animal (hyperonyme)

Rose (hyponyme) > fleur (hyperonyme)

2.5.3 La relation partie-tout

On les appelle aussi Méronymes (partie) et holonymes (tout).

1. La méronymie :
   La méronymie est une relation sémantique entre mots d'une même langue. Des termes liés par méronymie sont des méronymes. La méronymie est une relation partitive hiérarchisée :
   Un méronyme X d'un mot Y est un mot dont le signifié désigne une sous-partie du signifié de Y.

2. L'holonymie :
   L'Holonymie est une relation sémantique entre mots d'une même langue. Des termes liés par holonymie sont des holonomes. L'holonymie est une relation partitive hiérarchisée :
   Un holonyme A d'un mot B est un mot dont le signifié désigne un ensemble comprenant le signifié de B. [24]

2.6 Relations d'équivalence et d'opposition

2.6.1 La synonymie

La synonymie est la relation d'équivalence sémantique qui existe entre les signifiés de deux unités linguistiques appartenant à la même catégorie grammaticale, mais ayant des signifiants différents.[25]

Le mot synonyme est couramment utilisé pour décrire une relation de synonymie approximative ou grossière. Mais de plus, la synonymie est en réalité une relation entre des sens plutôt qu'entre des mots.[26]

Exemple : surprise = étonnement / façon = manière / manger = croûter.

2.6.2 La polysémie

La plupart des mots ont plus d'un sens. Homonyme même son et orthographe avec une signification différente, comme, banque (rivière) contre banque (financière). Polysémie différents sens du même mot.[23]

2.6.3 L'antonymie

Les antonymes sont définis comme des mots de sens contraire, et comme tels, ils paraissent opposés aux synonymes. Ils fonctionnent comme eux : synonymie partielle / antonymie partielle. Ils participent dans le même processus puisqu'un mot polysémique a selon les acceptions et les emplois des antonymes différents :

L'adjectif clair

Eau trouble / eau claire

Couleur foncée / couleur claire

Idée obscure / idée claire [25]

2.7 Conclusion

Dans ce chapitre, nous avons examiné le rôle essentiel des ontologies pour la représentation des connaissances. Ces modèles conceptuels visent à capturer les connaissances de manière générale et à fournir une représentation consensuelle, réutilisable et partageable.

Wordnet, en tant que base lexicale sémantique et informatique, est présenté comme un exemple emblématique d'ontologie. Au fil du temps, WordNet a acquis un caractère de plus en plus ontologique, devenant ainsi une référence dans ce domaine.

Les ontologies, à l'instar de Wordnet, ne sont pas seulement des instruments de structuration des connaissances, mais aussi des moteurs de progrès dans la compréhension et le traitement du langage naturel. Ainsi, l'intégration des relations lexico-sémantiques dans les systèmes de traitement automatique des langues ouvre la voie à des applications plus intelligentes et efficaces, capables de gérer la complexité du langage naturel et d'en tirer des insights significatifs.

Chapitre 3

Expérimentations

3.1 Introduction

Dans la catégorisation de textes, plusieurs approches ont été explorées, les plus courantes étant basées sur le ”sac de mots”. Cette technique transforme chaque texte en un vecteur représentant la fréquence des mots. Bien qu'elle soit appréciée pour sa simplicité et sa clarté, elle présente des inconvénients notables. La représentation ”sac de mots” présente plusieurs limites, notamment son incapacité à prendre en compte les expressions multi-mots qui modifient le sens, comme ”pomme de terre”. Elle ne capture pas non plus les relations sémantiques entre les termes, comme les synonymes ou les hiérarchies. De plus, cette approche manque de capacité de généralisation, peinant à regrouper des mots spécifiques sous des catégories générales.

Face à ces problématiques, de nombreux travaux de recherche se sont tournés vers l'utilisation des relations lexico-sémantiques. L'objectif est de proposer de nouvelles méthodes de représentation plus performantes.

Notre travail porte sur la catégorisation automatique de textes, qui consiste à classer des documents en utilisant des relations lexico-sémantiques. Ce chapitre est crucial, car il relie les concepts théoriques à des données textuelles. Nous abordons les défis de la représentation des textes, en proposant une méthode où chaque document est modélisé par un vecteur de concepts, enrichi par des synonymes et des relations sémantiques. Ensuite, nous explorons la phase de classification, où l'apprentissage du modèle dépend du choix du corpus et de l'algorithme utilisé, qu'il s'agisse de Naïve Bayes ou de SVM, influençant directement la performance du modèle. Enfin, nous procédons à une évaluation pour mesurer l'efficacité du modèle de catégorisation. La figure 3.1 illustre l'architecture de notre approche pour la catégorisation de texte naturel.

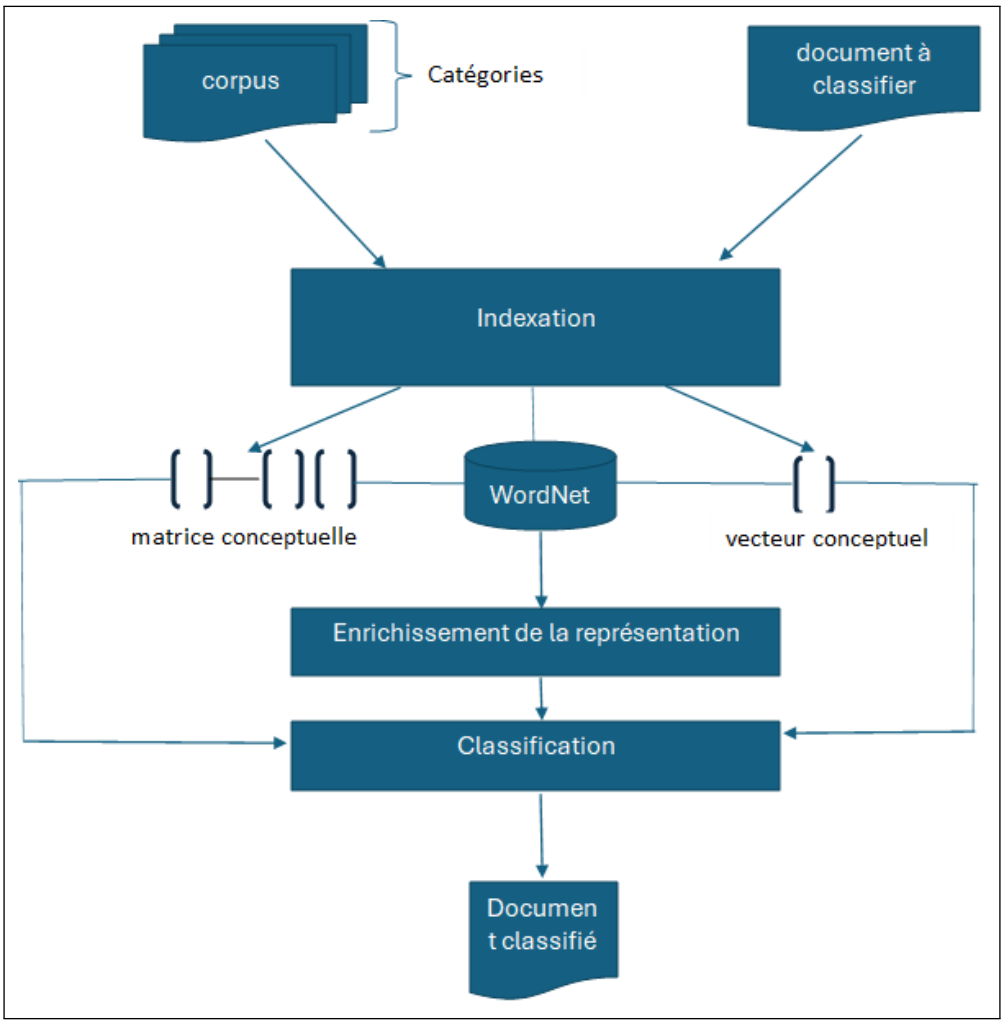

Figure 3.1 : Architecture de notre approche

3.2 Les phases d'implémentation

3.2.1 Phase d'indexation

Le principal défi dans la catégorisation de textes est de les représenter pour faciliter le traitement tout en conservant les informations pertinentes. Une bonne représentation améliore la performance des algorithmes de classification. Traditionnellement, la méthode du "sac de mots" a été utilisée, reposant sur les mots et leurs fréquences.

Pour aller plus loin, nous proposons une représentation conceptuelle enrichie qui saisit les concepts et leurs relations sémantiques, quelle que soit la langue. Cette approche aboutit à une matrice document-concept, dans laquelle chaque document est représenté par un vecteur de concepts.

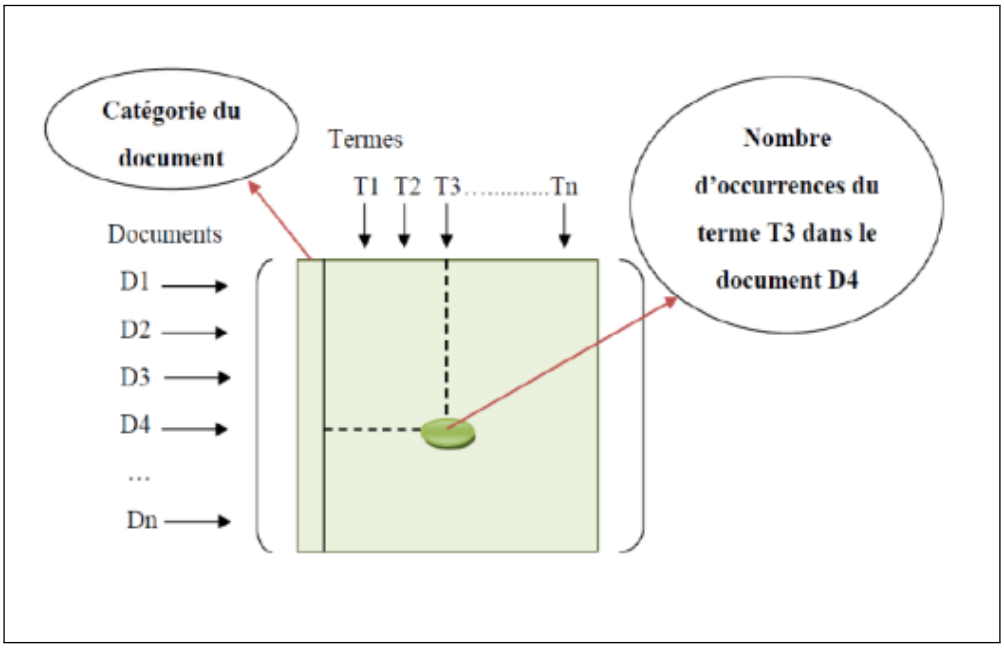

Figure 3.2 : Représentation matricielle d'un corpus

Afin de générer une telle représentation pour chaque document, nous appliquons les étapes suivantes :

3.2.1.1 Tokenisation

Cette étape consiste à découper le texte en un ensemble de mots. Cela permet de reconnaître les espaces de séparation entre les mots, les codes et les marques de numérotation.

3.2.1.2 Élimination des mots vides

Dans cette étape, on élimine les mots les plus répandus mais les moins porteurs d'information, tels que les articles, les prépositions et les pronoms, pour concentrer l'analyse sur les mots les plus importants.

Dans la figure 3.3, nous montrons la liste des mots vides qu'ont été employés dans notre analyse.

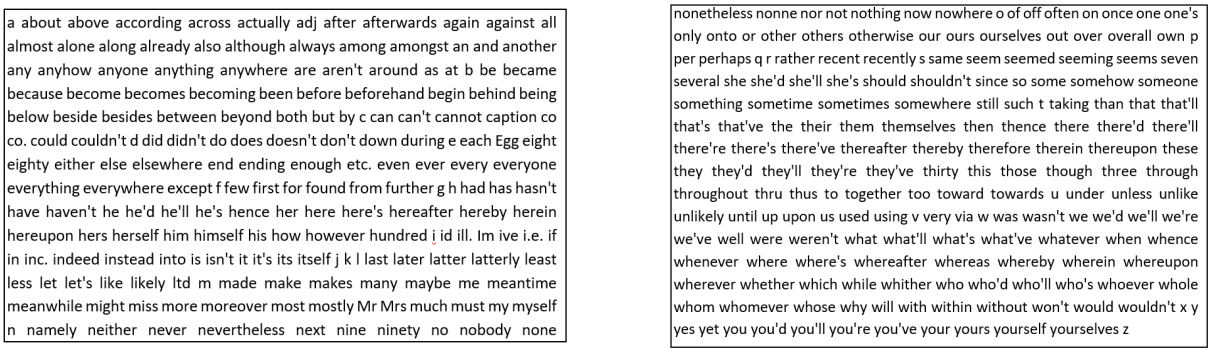

Figure 3.3 : la liste des mots vides utilisée

3.2.1.3 Mapping des mots en concepts et desambiguisation

Après avoir effectué le prétraitement de chaque document et l'avoir représenté sous forme de vecteurs de mots étiquetés grammaticalement, nous passons à l'étape suivante : la substitution de chaque mot par le concept le plus approprié. Ce processus de mapping permet de réduire la dimensionnalité de l'espace de représentation, en remplaçant tous les mots ayant une signification similaire par un même concept.

Étant donné qu'un mot peut posséder plusieurs sens, il est crucial de choisir le sens le plus pertinent. Cette étape consiste à identifier le sens le plus approprié d'un mot en fonction du contexte. Il existe plusieurs méthodes pour réaliser cette tâche [28], dans notre travail nous avons utilisé la méthode UKB[29]. Cette démarche est essentielle pour déterminer le sens adéquat.

Voici les principales étapes de l'utilisation de UKB pour la désambiguïsation sémantique :

- Prétraitement du texte : Tokenisation, étiquetage morphosyntaxique, etc.
- Construction du graphe sémantique : Utilisation de ressources lexicales comme WordNet pour créer un graphe où les nœuds représentent les sens de mots et les arêtes les relations sémantiques entre eux.
- Calcul des scores de pertinence des sens : Pour chaque mot ambigu, UKB calcule un score de pertinence pour chaque sens possible en se basant sur les relations sémantiques dans le graphe.
- Sélection du sens le plus pertinent : Le sens avec le score le plus élevé est sélectionné comme le sens le plus approprié dans le contexte.

3.2.2 Phase d'enrichissement

Cette étape consiste à améliorer et étendre la matrice conceptuelle en impliquant l'ajout des relations lexico-sémantiques. L'objectif est de rendre les données plus riches

pour une meilleur compréhension du corpus. Dans notre approche, nous avons considéré les principales relations sémantiques offertes par la ressource lexicale WordNet , telles qu'Hyponymie, Hypéronymie, Méronymie et Holonymie expliqués en profondeur dans chapitre 2.

La figure 3.4 montre un exemple d'un hypéronyme :

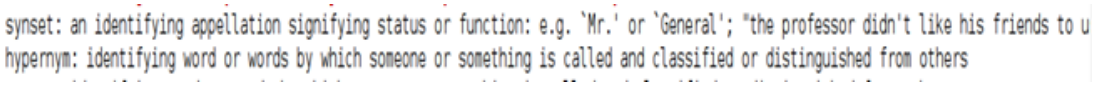

Figure 3.4 : exemple d'hypéronyme

Algorithm 1 : Phase d'enrichissement

Input : La matrice conceptuelle M, le vecteur de concepts C, le depth d

foreach concept $c_j \in C$ **do**
  for $i \leftarrow 0$ **to** d **do**
    N $\leftarrow$ Liste vide pour les nouveaux concepts
    Extraire c_n en relations lexico-sémantiques R avec c_j
    Ajouter c_n à la liste N
    if c_n *n'existe pas dans* C **then**
      Ajouter c_n dans C
    else
      foreach document $d_k \in D$ **do**
        Mettre à jour le TF : $TF(c_n, d_k) = TF(c_n, d_k) + 1$

3.2.3 Phase de classification

Au cours de cette phase, l'objectif est de classer le document à catégoriser en s'appuyant sur des documents préalablement classés. Plusieurs méthodes de classification sont disponibles, et nous avons opté pour la méthode Naïve Bayes dans notre implémentation. Cet algorithme calcule la probabilité a posteriori pour chaque classe, permettant ainsi d'attribuer le document à celle qui présente la probabilité la plus élevée.

Les principales étapes de ce processus sont les suivantes :

- Effectuer un apprentissage afin de créer un vecteur conceptuel correspondant au nouveau document.
- Déterminer la classe du document en se basant sur l'indexation {**Tokenisation, Élimination des mots vides, Mapping des mots en concepts**}.
- appliquer le modéle construit sur le nouveau document à fin de trouver sa catégorisation .

Algorithm 2 : Phase de classification

Input :
Le vecteur conceptuel d'un document d à classifier,
La matrice conceptuelle M du corpus,
L'ensemble de classes $C = \{c_1, c_2, c_3, \ldots, c_m\}$
Output :
La classe c du nouveau document

```
foreach classe c_i ∈ C do
    Calculer les probabilités a priori pour chaque classe c_i
Trier les probabilités calculées par ordre décroissant
Attribuer le document d à la classe c ayant la plus grande probabilité a priori
```

3.2.4 Phase d'évaluation

Après avoir classé les documents selon notre modèle, nous avons évalué les performances de ce modèle de catégorisation en utilisant le rappel, la précision et le F-Mesure comme critères d'évaluation. (Voir chapitre 1 Évaluation de processus)

3.3 Environnement et outils de développement

Pour réaliser notre approche nous avons utilisé pas mal d'outils ainsi que langages dans l'implémentation de notre approche qui sont détaillés comme suit :

3.3.1 Environnements matériel

Notre travail a été développé sur une machine Intel core i5 11éme génération avec une capacité RAM de 4Go.

3.3.2 Système d'exploitation Linux

Linux est un système d'exploitation open source basé sur le noyau Linux, développé initialement par Linus Torvalds en 1991. C'est l'un des systèmes d'exploitation les plus populaires pour les serveurs, les superordinateurs et les systèmes embarqués, bien qu'il soit également utilisé sur des ordinateurs personnels et des appareils mobiles via des distributions comme Ubuntu, Fedora ou Android.

3.3.3 Environnement de développement

On a choisi de développer notre implémentation en Java, au sein de l'environnement de développement intégré (IDE) Apache NetBeans version 21. Un des premiers IDE pour Java, est un environnement de développement intégré open-source soutenu depuis près de

30 ans. La version récente, Apache NetBeans 21, offre plusieurs avantages justifiant son choix :

- Principalement utilisé pour le développement Java, il prend également en charge C/C++, PHP, HTML5, JavaScript et d'autres langages.
- Il propose des fonctionnalités d'édition de code, de compilation, de débogage et de déploiement d'applications.
- Il prend en charge de nombreux Framework et technologies Java.
- Ses modules de base sont axés sur Java et agissent sur les fichiers inclus dans l'espace de travail.

3.3.4 Langage JAVA

Pour le développement de ce projet, on a choisi le langage de programmation Java. Ce choix s'appuie sur plusieurs atouts majeurs de Java :

- Il est robuste. On dit de Java qu'il est le langage le plus fiable et le plus puissant.
- L'informatique distribuée. Il s'agit d'une méthode où plusieurs ordinateurs travaillent ensemble dans un réseau.
- Il est multithread. En interne, un programme Java peut effectuer plusieurs tâches à la fois.
- Puisque Java appartient à la programmation orientée objet, il permet à un développeur d'écrire des programmes types et de réutiliser le code.

3.3.5 WordNet

Nous avons utilisé la version 3.0 de WordNet qui est une base de données lexicographique. C'est une vaste ressource lexicale qui s'est imposée comme une référence pour la langue anglaise. Le chapitre 2 fournit une présentation détaillée sur WordNet .

Le tableau ci-dessous présente la structure de WordNet 3.0, une version antérieure de cette base de données lexicale de la langue anglaise. Il indique le nombre de mots, de synsets (ensembles de synonymes) et de sens (significations) que contient WordNet 3.0, globalement et par catégorie grammaticale :

POS	Unique Strings	Synsets	Total Word-Sense Pairs
Noun	117798	82115	146312
Verb	11529	13767	25047
Adjective	21479	18156	30002
Adverb	4481	3621	5580
Totals	155287	117659	206941

Tableau 3.1 : Caractéristiques du nombre de mots, synsets et sens dans WordNet [27]

3.3.6 Plate-forme Weka 3.8.6

Weka est un environnement d'apprentissage automatique "open source" qui se présente sous la forme d'une collection d'algorithmes d'apprentissage destinés à réaliser des tâches de fouille de données. L'utilisation de Weka s'appuie sur ses fonctionnalités avancées et sa facilité d'utilisation :

- Il est libre et gratuit, distribué selon les termes de la licence publique générale GNU.
- Il 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 ;
- Il contient une collection complète de préprocesseurs de données et de techniques de modélisation ;
- Il est facile à utiliser par un novice en raison de l'interface graphique qu'il contient.

3.3.7 Corpus utilisé

Corpus Reuters : le corpus Reuters est une ressource précieuse pour la recherche en classification de texte, offrant un ensemble riche et bien structuré de document qui sont souvent étiquetés avec des catégories pré-assignées, ce qui le rend idéal pour l'entraînement et l'évaluation des algorithmes de classification.

Dans notre expérimentation, nous allons utiliser ce corpus en anglais repartis en 8 catégories.

La figure 3.5 illustre un exemple d'un document de cette collection.

```
<TITLE>COMPUTER TERMINAL SYSTEMS &lt;CPML> COMPLETES SALE</TITLE>
<TEXT>
Computer Terminal Systems Inc said above according
it has completed the sale of 200,000 shares of its common
stock, and warrants to acquire an additional one mln shares, to
&lt;Sedio N.V.> of Lugano, Switzerland for 50,000 dlrs.
    The company said the warrants are exercisable for five
years at a purchase price of .125 dlrs per share.
    Computer Terminal said Sedio also has the right to buy
additional shares and increase its total holdings up to 40 pct
of the Computer Terminal's outstanding common stock under
certain circumstances involving change of control at the
company.
    The company said if the conditions occur the warrants would
be exercisable at a price equal to 75 pct of its common stock's
market price at the time, not to exceed 1.50 dlrs per share.
    Computer Terminal also said it sold the technolgy rights to
its Dot Matrix impact technology, including any future
improvements, to &lt;Woodco Inc> of Houston, Tex. for 200,000
dlrs. But, it said it would continue to be the exclusive
worldwide licensee of the technology for Woodco.
    The company said the moves were part of its reorganization
plan and would help pay current operation costs and ensure
product delivery.
    Computer Terminal makes computer generated labels, forms,
tags and ticket printers and terminals.
 Reuter
</TEXT>
```

Figure 3.5 : exemple d'un document Reuters

3.3.8 Bibliothèques Utilisées

Le développement de nos expérimentations s'est appuyé sur les bibliothèques suivantes :

1. **JWNL API :** JWNL (Java WordNet Library) est une API pour accéder aux dictionnaires relationnels de style WordNet. Il fournit également des fonctionnalités au-delà de l'accès aux données, telles que la découverte de relations et le traitement morphologique. Elle est compatible avec les versions WordNet 2.0 à 3.0.

2. **JFreeling API :** JFreeling est une bibliothèque Java open-source qui fournit des outils de traitement automatique du langage naturel, notamment pour l'analyse morphologique, syntaxique et sémantique de textes.

3. **WEKA API :** Weka est une bibliothèque Java open-source qui fournit une API riche pour l'exploitation des techniques d'apprentissage automatique et d'exploration de données.

3.4 Résultats et discussion

Notre expérimentation consiste à évaluer les performances de notre approche sur le corpus utilisé. Les résultats de cette expérimentation sont recapitulés dans les tableaux 3.2 et 3.3 et les figures 3.6 et 3.7.

3.4.1 Expérimentation 1

Notre première expérimentation consiste à comparer les performances avec différentes depth sur les quatre relations (hyperonymie, hyponymie, meronymie et holonymie). Les résultats de cette expérimentation sont récapitulés dans table 3.2 et figure 3.6.

	Précision			Rappel			F-mesure		
	Depth1	Depth2	Depth3	Depth1	Depth2	Depth3	Depth1	Depth2	Depth3
Hyperonymie	0,881	0,893	0,897	0,800	0,811	0,815	0,815	0,825	0,832
Hyponymie	0,911	0,911	0,911	0,820	0,820	0,820	0,835	0,835	0,835
Meronymie	0,911	0,911	0,911	0,820	0,820	0,820	0,835	0,835	0,835
Holonymie	0,914	0,914	0,914	0,824	0,824	0,824	0,839	0,839	0,839

Tableau 3.2 : Comparaison des résultats des relations lexico-sémantiques selon différentes profondeurs.

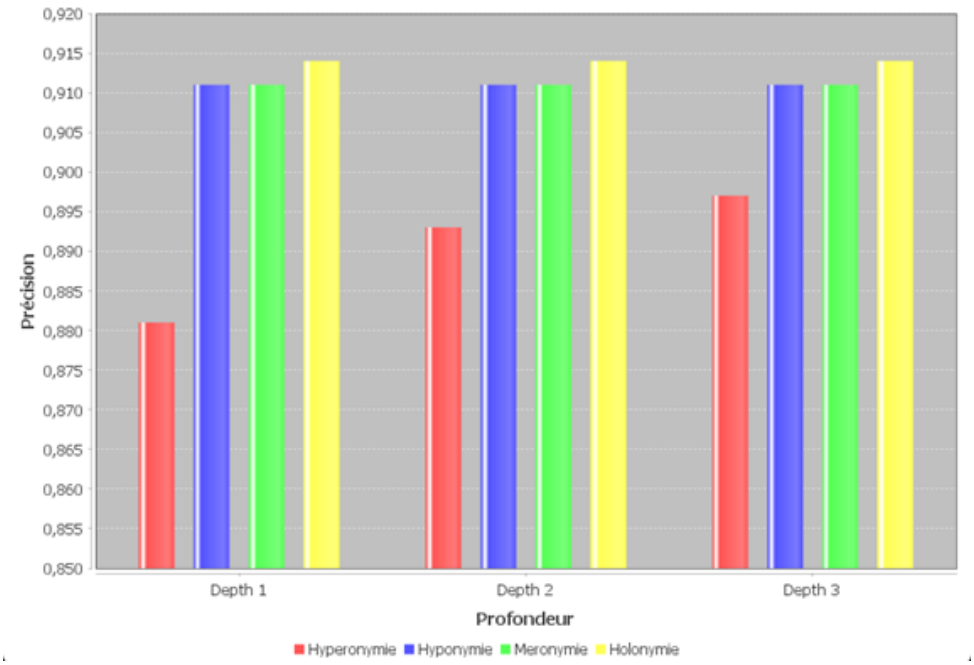

Figure 3.6 : Représentation des résultats fournis dans table 3.2

Les résultats obtenus montrent que :

- La meilleure relation est l'holonymie cependant l'hyperonymie est la moin performante.
- Concernant le depth avec l'hyperonymie, on constate qu'en augmentant la valeur du depth , les valeurs s'améliore successivement.
- En ce qui concerne les relations hyponymie, meronymie et holonymie on remarque qu'en augmentant la valeur de la profondeur, les valeurs se stabilisent pour une certaine valeur pour chacune de ces relations.

3.4.2 Expérimentation 2

expérimentation sans utiliser les relations lexico sémantiques. Les résultats de cette expérimentation sont récapitulés dans le tableau 3.3.

Précision	Rappel	F-mesure
0,405	0,426	0,402

Tableau 3.3 : comparaison des résultats sans intégration des relations lexico-sémantiques

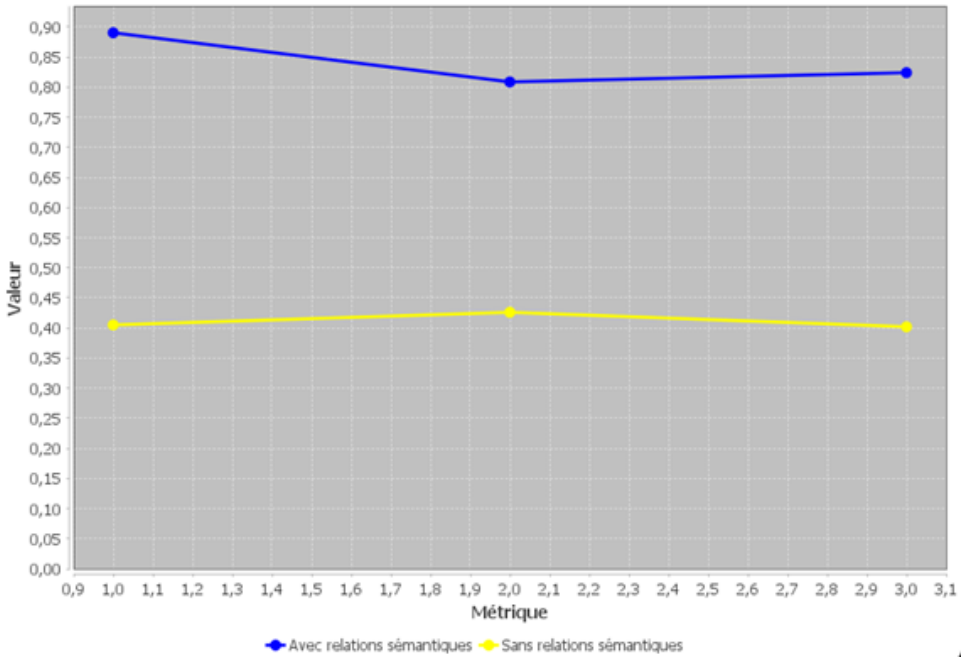

Figure 3.7 : représentation de comparaison entre table 3.2 et 3.3

En analysant ces résultats ainsi que la figure 3.7, on voit clairement que les résultats sont dégradés de la valeur 0.8 avec les relations lexico-sémantiques à la valeur 0.4 sans d'utilisation de ces relations. Ce qui prouve l'utilité de notre travail.

3.5 Conclusion

Ce chapitre a détaillé la conception et la mise en œuvre de notre approche, qui exploite les relations lexico-sémantiques pour catégoriser et classifier des textes. Pour cela, nous avons construit une représentation conceptuelle basée sur WordNet et le corpus Reuters. Ensuite, nous avons utilisé la méthode Naïve Bayes pour attribuer chaque document à une catégorie et avons évalué les performances du modèle à l'aide de critères d'évaluation. Les expérimentations présentées dans ce chapitre ont démontré l'utilité des relations lexico-sémantiques pour la catégorisation de textes.

Conclusion générale

Dans ce mémoire, la problématique de la catégorisation automatique des textes naturels a été abordée, en réponse aux limites des méthodes traditionnelles, telles que le sac de mots, qui peinent à saisir la complexité du langage naturel. Pour surmonter ces insuffisances,Notre recherche adopte une approche fondée sur les relations lexico-sémantiques.L'objectif principal est d'enseigner à une machine à classer correctement des documents en fonction de leur contenu textuel et sémantique, offrant ainsi une représentation plus riche et précise des textes.

La catégorisation que nous proposons est ancrée dans une représentation conceptuelle tirant parti de WordNet. Elle se base sur les relations lexicales et sémantiques entre concepts plutôt qu'entre mots. cette méthode a considérablement amélioré les performances du classificateur. La richesse sémantique de WordNet permet une meilleure prise en compte des différents sens et usages des termes, surpassant les limites des méthodes classiques. Toutefois, des défis demeurent dans l'identification optimale des termes similaires.

Ce mémoire est structuré en trois chapitres distincts, chacun explorant divers aspects de notre approche. Le premier chapitre détaille le processus de catégorisation textuelle, ses principales étapes et les applications associées. Le deuxième chapitre est consacré aux ontologies, avec un accent particulier sur WordNet en tant qu'ontologie lexicale permettant de modéliser les relations lexico-sémantiques. Enfin, Le troisième chapitre décrit en détail les étapes de notre application et les résultats obtenus, qui montrent une performance très satisfaisante, mettant en évidence l'utilité des relations lexico-sémantiques.

Le temps alloué à ce travail étant relativement limité, il a été difficile de fixer certains paramètres nécessaires pour explorer plus en profondeur d'autres approches et algorithmes. En perspective, nous proposons les axes suivants :

- Explorer de nouvelles méthodes de représentation des textes, afin d'évaluer leur potentiel par rapport à notre méthode actuelle.
- Implémenter une variété d'algorithmes de classification (machines à vecteurs de support, arbres de décision, réseaux de neurones) pour les comparer de manière approfondie à notre classifieur Naive Bayes.
- Utiliser les ontologies et les dictionnaires pour enrichir la représentation sémantique des textes, dans le but d'améliorer la qualité des prédictions.
- Évaluer les performances de notre approche sur d'autres jeux de données pour confirmer sa robustesse et sa capacité de généralisation.
- Il serait intéressant d'utiliser une méthode alternative, semblable à celle de UKB, pour optimiser le processus de désambiguïsation

Bibliographie

[1] **Radwan JALAM (2003).** Apprentissage automatique et catégorisation de textes multilingues, Thèse de doctorat, Université Lumière Lyon 2, juin 2003.

[2] **Fabrizio SEBASTIANI (2002).** Machine Learning in Automated Text Catégorisation, Conseil recherché National, Italie, mars 2002.

[3] **Zeste de savoir (2020).** Un peu de Machine Learning avec les SVM, 21 août 2020.

[4] **Fatiha Barigou, Baghdad Atmani, Youcef Bouziane, Naouel Barigou (2020).** Accélération de la méthode des K plus proches voisins pour la catégorisation de textes, Département d'Informatique, Université d'Oran.

[5] **Matallah Houcine (2020).** Classification automatique de textes approche orientée agent, Mémoire de magister, Département d'informatique, Algérie.

[6] **Aouine Mohammed (2020).** Catégorisation automatique de texte arabe, Mémoire de magister, Département d'informatique, Algérie.

[7] **Moulinier (1996).** Une approche de la catégorisation de textes par l'apprentissage symbolique, Sciences et techniques communes, Paris.

[8] **J. Clech, D. A. Zighed (1996).** Une technique de réétiquetage dans un contexte de catégorisation de textes.

[9] **Ameni Bouaziz (2018).** Catégorisation automatique de news à l'aide de techniques d'apprentissage supervisé, Rapport de projet de fin d'études, Université Nice Sophia Antipolis.

[10] **A. McCallum, R. Rosenfeld, T. Mitchell, A. Y. Nigam (1998).** Improving Text Classification by Shrinkage in a Hierarchy of Classes.

[11] **Reuters (2024).** Disponible à https://www.reuters.com.

[12] **Scikit-learn (2024).** Twenty Newsgroups Dataset, disponible à https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html.

[13] **SpamAssassin (2024).** Apache SpamAssassin, disponible à https://spamassassin.apache.org.

[14] **DataScientest (2024).** Convolutional Neural Network : Everything You Need to Know, disponible à https://datascientest.com/en/convolutional-neural-network-everything-you-need-to-know.

[15] **Rim MESSAOUDI (2021).** Intégration d'ontologies dans la classification parallèle de données médicales pour le diagnostic de lésions du foie, Thèse de doctorat, L'Université Clermont Auvergne et l'Université de Sfax, 2 juin 2021.

[16] **Mohamed Amine BENTAALLAH (2021).** Utilisation des Ontologies dans la Catégorisation de Textes Multilingues, Thèse de doctorat, Université Djilali Liabes de Sidi Bel Abbes, Département d'Informatique.

[17] **Nathalie Hernandez (2005).** Ontologies de domaine pour la modélisation du contexte en Recherche d'Information, Thèse de doctorat, Université Paul Sabatier, Toulouse, France, décembre 2005.

[18] **Thomas R. Gruber (1993).** A translation approach to portable ontology specifications, Knowl. Acquis., 5(2) :199–220, June 1993.

[19] **Salah MAACHE (2020).** Apport des Ontologies pour les Placements Financiers Intelligents, Mémoire de Magister, Université Ferhat Abbas Sétif, Département d'informatique.

[20] **François-Régis Chaumartin (2007).** WordNet et son écosystème : un ensemble de ressources linguistiques de large couverture, Colloque BD lexicales, Montréal, Canada, avril 2007.

[21] **Revue française de linguistique appliquée (2008).** Ressources lexicales, terminologiques et ontologiques : une analyse comparative dans le domaine de l'informatique, 2008/1, Vol. XIII, p. 97-118.

[22] **Christiane, Fellbaum (1998).** Wordnet – An Electronic Lexical Database, The MIT Press, Cambridge, Mass.

[23] **Didaquest (2020).** Disponible à https://didaquest.org/wiki/WordNet.

[24] **G.A.Miller (1997).** "Nouns in Wordnet : A lexical inheritance system " . in : Five papers on Wordnet,10-25, revisedversion, 1997 http://www.cogsci.princeton.edu/wn/(sept.1993),

[25] **Université Echahid Hamma Lakhdar d'el Oued (2020).** E-learning, disponible à https://elearning.univ-oued.dz.

[26] **Daniel Jurafsky, James H. Martin (2020).** Speech and language processing – Chapter C : WordNet : word relations, senses, and disambiguation.

[27] **Princeton WordNet (2024).** Disponible à https://wordnet.princeton.edu/documentation/wnstats7wn.

[28] **Wikipedia (2024).** Word-sense disambiguation, disponible à https://en.wikipedia.org/wiki/Word-sense_disambiguation.

[29] **Padro, L., Reese, S., Agirre, E., & Soroa, A. (2024).** Semantic services in FreeLing 2.1: WordNet and UKB, TALP Research Center, Universitat Politécnica de Catalunya, ISAE-Supaero Université Paul Sabatier, IXA NLP Group, University of the Basque Country.

[30] https://www.xlstat.com/fr/solutions/fonctionnalites/classifieur-bayesien-naif

[31] **Tom M.Mitchell** ," chapter 3 GENERATIVEANDDISCRIMINATIVE CLASSIFIERS : NAIVEBAYESANDLOGISTICREGRESSION", CMU school of computer science, Pennsylvania

[32] https://datascientest.com/classificateur-naive-bayes-tout-savoir

[33] **Colas, F., Brazdil, P. (2006)** Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks. In : Bramer, M. (eds) Artificial Intelligence in Theory and Practice. IFIP AI 2006. IFIP International Federation for Information Processing, vol 217. Springer, Boston, MA.

Résumé

La catégorisation automatique de textes (CT) est une tâche clé en traitement automatique du langage naturel (TALN) qui consiste à assigner un document à une ou plusieurs catégories prédéfinies. Les relations lexico-sémantiques, qui représentent les liens de sens entre les mots, jouent un rôle crucial dans ce processus en permettant de mieux saisir la signification globale d'un texte. Notre travail se concentre sur l'évaluation de l'impact de l'intégration des relations lexico-sémantiques dans la catégorisation de textes. Pour ce faire, nous avons développé des modèles de classification qui exploitent ces relations et les avons évalués à l'aide de métriques standard telles que la précision, le rappel et la F-mesure, afin de mesurer leur efficacité. L'implémentation de ces modèles a été réalisée en JAVA, en utilisant la base de données lexicale WordNet pour l'enrichissement sémantique, et la plate-forme Weka pour l'évaluation des performances. Les résultats obtenus montrent amélioration notable ce qui prouve l'utilité de notre travail.

Mots clés : catégorisation de texte, relations lexico-sémantiques, WordNet , Weka,java,NLP.

Abstract

Automatic text categorization (CT) is a key task in natural language processing (NLP) that consists of assigning a document to one or more predefined categories. Lexico-semantic relationships, which represent the links of meaning between words, play a crucial role in this process by allowing a better understanding of the overall meaning of a text. Our work focuses on the evaluation of the impact of the integration of lexico-semantic relations in the categorization of texts. To do this, we developed classification models that exploit these Relationship and evaluated them using standard metrics such as precision, recall, and F-measure, to measure their effectiveness. The implementation of these models was carried out in JAVA, using the WordNet lexical database for semantic enrichment, and the Weka platform for performance evaluation. The results obtained show significant improvement, which proves the usefulness of our work.

Keywords : text categorization, lexico-semantic relations, WordNet, Weka,java.

ملخص

يعد التصنيف التلقائي للنص (CT) مهمة أساسية في معالجة اللغة الطبيعية (NLP) تتكون من تعيين مستند إلى فئة واحدة أو أكثر محددة مسبقا. تلعب العلاقات المعجمية الدلالية ، التي تمثل روابط المعنى بين الكلمات ، دورا حاسما في هذه العملية من خلال السماح بفهم أفضل للمعنى العام للنص. يركز عملنا على تقييم تأثير تكامل العلاقات المعجمية الدلالية في تصنيف النصوص. للقيام بذلك ، قمنا بتطوير نماذج تصنيف تستغل هذه العلاقات وتقييمها باستخدام مقاييس قياسية مثل الدقة والاستدعاء وقياس F ، لقياس فعاليتها. تم تنفيذ هذه النماذج في JAVA ، باستخدام قاعدة البيانات المعجمية WordNet للإثراء الدلالي ، ومنصة Weka لتقييم الأداء. النتائج التي تم الحصول عليها تظهر تحسنا

مفتاحية كلمات : التصنيف التلقائي للنص , العلاقات المعجمية الدلالية ,WordNet ، Weka, java, NLP.