République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma-

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique

Mémoire de projet de fin d'étude Master

Filière : Informatique

Option : Système Informatique

Thème :

Sélection des caractéristiques basée sur le plongement lexical pour la classification des textes

Encadré Par :

Mr.Farek Lazhar.

Présenté par:

Ferdi Dounya

Septembre 2021

Remerciement

On dit souvent que le trajet est aussi important que la destination. Les cinq années qui j'ai passée à l'université j'ai permise de bien comprendre la signification de cette phrase toute simple. Ce parcours, en effet, ne s'est pas réalisé sans défis et sans soulever de nombreuses questions pour lesquelles les réponses nécessitent de longues heures de travail.

*Je tiens à la fin de ce travail à remercier « **Allah** » le tout puissant de me l'avoir donné la foi et de me l'avoir permis d'en arriver là.*

*Je tiens à remercie du fond du cœur mes parents « **Seliman** » et « **Akila** » pour son soutien moral tout au long de mes années d'études et tous les sacrifices qu'ils ont consenti pour me permettre de suivre mes études dans les meilleures conditions possibles et n'avoir jamais cessé de m'encourager.*

*Je tiens à exprimer ma profonde gratitude à « **Dr.Farek Lazher** », pour avoir encadrée et dirigée mes recherches. Je le remercie pour m'avoir soutenue et appuyée tout au long de ma thèse. Ses précieux conseils, son exigence et ses commentaires ont permis d'améliorer grandement la qualité de mes travaux et de ce mémoire. Sincèrement, grâce à-il, j'ai pu apprendre beaucoup de choses dont certaines fort utiles pour mes travaux académiques bien sûr, mais aussi des choses importantes pour mon développement personnel. Je n'oublie pas enfin son aide précieuse dans la relecture et la correction de ma thèse.*

Je remercie également mes sœurs, et mes frères qui m'ont toujours encouragée et soutenue moralement.

Et a tous mes amis qui m'ont soutenue de proche ou de loin un grand merci.

Enfin, je remercie tous les professeurs du département de l'informatique de l'université de 8 mai 1945 de Guelma

Dédicace

Je dédie ce travail

A mes parents

A ma mère, qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternel gratitude.

A mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit. Merci pour les valeurs nobles, l'éducation et le soutien permanents.

A ma sœur décédée Bouchra

Je serai diplômée ma sœur, le jour que j'ai souhaitée est venu, et tu n'es pas avec moi, tu t'es empressée de partir et tu ne m'as pas attendu. Je serai diplômée ma sœur, et toute joie sans toi est incomplète. J'obtiendrai mon diplôme et les pleurs m'étouffent.

Que Dieu te fasse miséricorde et nous rassemble avec toi au paradis.

A ma sœur Imen

Les mots ne suffisent guère pour exprimer l'attachement, l'amour et l'affection que je te porte, Mon ange gardien et fidèle compagnon dans les moments les plus sensibles de cette vie. Je vous dédie ce travail avec mes meilleurs vœux de bonheur, de santé et de réussite

A mes chères petits frères Ahmed et Anas

Ils sont présents dans tous mes moments d'examens par leur soutien moral et de belles surprises sucrées. Je vous exprime à travers ce travail mes sentiments de fraternité et d'amour.

DOUNYA.

Résumé

La sélection des caractéristiques est un processus largement utilisé pour trouver un sous-espace de variables pertinentes en vue d'augmenter la performance des techniques de classification. Les techniques classiques de sélection des caractéristiques comme IG (Information Gain), MI (Muetel Information), CH2 (Chi-Square), etc., ont prouvé leur efficacité de trouver des bons espaces représentatifs pour les corpus textuels. Malheureusement, ces techniques ne prennent en considération que la fréquence du mot sans considérer sa sémantique.

Dans ce travail, nous nous focalisons à étudier l'impact de la sélection de mots basée sur le plongement lexical (en. word embedding), en considérant la sémantique lexical du mot à l'aide des vecteurs numériques pour mesurer la similarité entre les mots par des métriques de distance, pour cela, nous choisissons le framework word2vec. Ensuite nous effectuons un clustering k-means de ces vecteurs pour écarter ceux qui augmentent la fonction objective de l'algorithme de clustering, où les vecteurs qui assurent une fonction objective optimale correspondent aux mots sélectionnés pour construire le modèle, où l'objectif est améliorer les performances des techniques de classification par la représentation des connaissances dans un espace plus réduit.

L'approche proposée est appelée **SCPL** (*Sélection des Caractéristiques par Plongement Lexical*) a été implémentée en langage **Python**. Les résultats d'expérimentations prouvent qu'elle offre de meilleurs performances que les algorithmes classiques existants dans la littérature.

Mots clés: sélection de caractéristique, classification, texte, plongement lexical, word2vec, clustering, K-means.

Abstract

Feature selection is a widely used process to find a sub-space of relevant features in order to increase the performance of classification. Classic techniques for feature selection such as GI (Information Gain), MI (Muetel Information), CH2 (Chi-Square), etc., have proven their efficiency by finding good representative spaces for textual corpora. Unfortunately, these techniques only take into consideration the frequency of the word without consider its semantics.

In this work, we focus on studying the impact of feature selection based on the Word embedding, considering the lexical semantic of the word as a vector of real values and that to measure the similarity between words by the metrics of distance. In this purpose, we choose the framework word2vec. Then we do a k-means clustering of these vectors to rule out those who increase the objective function of the clustering algorithm, where the vectors that provide an optimal objective function correspond to selected words used to build the model, where the main aim is to improve the performance of classification techniques by representating features in a small representative space.

The proposed approach is called **SCPL** (Sélection de caractéristiques par plongement lexicale) was implemented in **Python** language. Exprimental results show that our approach performs better than commun algorithms existing in the literature

Key words : feature selection, classification, text, word embedding, word2vec, clustering, K-means.

Table des matières

Liste des figures

Liste des tableaux

Abréviations et acronymes

<AD> <Arbre de Décision>

<BERT> <Bidirectional Encoder Representations from Transformers>

<BOW> <Bag of Words>

<CBOW> <Continous Bag-of-Word>

<CH2, KH2> <CH-Square>

<DBSCAN> <Density Based Spatial Clustering of Application with Noise>

<DF> <Document Frequency>

<EM> <Espérance-Maximisation>

<ELMO> <Embeddings Encoder Representations from Transformers>

<GI> <Gini Index>

<GLoVe> <Global Vectors for Words Representations>

<GMDH> <Group Method of Data Handling>

<IA> <Intelligence Artificielle>

<IG> <Information Gain>

<KNN> <K Nearest Neighbor>

<KPCA> <Kernel Principal Component Analysis>

<LSA> <Latent Semantic Analysis>

<LSTM> <long short-term memory>

<MI> <Mutuel Information>

<MSV> <Modèle d'Espace Vectoriel>

<MLM> < Masked Language Model>

<N-MRMCR-MI> <la Normalisation de la Pertinence Maximale et de la Redondance Commune Minimale>

<NSP> <Next Sentence Prediction>

<OVA> <One-Vs-All>

<OVO> <One-Vs-One>

<RFC> <Random Forest Classifier>

<SCPL> <Sélection de Caractéristiques par Plongement Lexical>

<Skip-Gram> <SG>

<SVD> <Singular Value Decomposition>

<SVM> <Support Vector Machines>

<SVM-REF> <Selection and Visualisation of the Most Relevant Features through non-linear kernels>

<TF> <Term Frequency>

<TF-IDF> <Term Frequency-Inverse Document Frequency>

Introduction Générale

Cette introduction présente la problématique traitée dans ce travail ainsi que la structure générale du mémoire.

1. Problématique

Depuis l'apparition des technologies de l'information et son développement au cours des dernières années, plusieurs nouvelles applications ont augmenté le nombre de textes viennent des sources comme les pages web, e-mail, blogs, bibliothèques numériques, médias sociaux, les annonces, les documents d'entreprises et les avis sur les produits, etc., ce qui a augmenté de plus en plus l'utilité de classification des textes.

La classification des textes est une tâche a un très important intérêt sur le plan socio-économique, où elle a beaucoup retenu l'attention des chercheurs dans divers domaines au cours de ces dernières années, elle a comme objectif d'associer automatiquement des documents à des classes (catégories, étiquettes, index) prédéfinies. Nous nous plaçons dans le cadre de l'apprentissage supervisé, cela consiste à superviser l'apprentissage en lui montrant des données (des exemples) de la tâche qu'elle doit réaliser. Dans les problèmes de classification supervisée, les étiquettes sont désignées en fonction de paramètres (les variables, les attributs), où la présence des attributs redondants ou non pertinents peut réduire la performance du modèle construit, ainsi que le temps d'apprentissage augmente à mesure que le nombre de caractéristiques inutiles dans la classification augmente. Par conséquent, il est indispensable de sélectionner les bons attributs ou caractéristiques (en. Features) pour une performance meilleure du modèle construit.

Dans la classification des textes, les attributs sont les mots qui constituent le vocabulaire utilisé par un algorithme de classification. Un mot du vocabulaire soit présent ou absent dans un document, d'où la performance du système dépend fortement des caractéristiques (mots) utilisées dans la tâche d'apprentissage. Donc la question qui se pose est comment distinguer entre les caractéristiques pertinentes qui ont un effet positif sur la performance de classification, et ceux qui ne sont pas pertinentes et qui ont un effet négatif sur la classification ?

Plusieurs algorithmes de sélection des caractéristiques, supervisés et non supervisés pour les données catégoriales et numériques existent dans la littérature, par exemple: Information Gain (IG), Gini Index (GI), CH-Square (CH2, KH2), Mutuel information (MI), etc., ces algorithmes de sélection ne s'utilisent seulement pour les données textuelles mais aussi pour n'importe quel type de données numériques ou catégoriales.

En fouille des données textuelles (Text-Mining), ces algorithmes classiques ont prouvé leurs efficacité en augmentant la performance du modèle entraîné tout en minimisant le coût et le temps d'apprentissage, mais malheureusement, ces algorithmes traitent le mot comme une unité isolée sans prendre en considération sa sémantique en considérant seulement sa fréquence. Mais la question qui se pose est comment faire une sélection de caractéristiques en se basant sur la sémantique des mots ?

Pour améliorer les performances des modèles construits autour de sélection des caractéristiques, dans ce travail nous proposons une approche basée sur la sémantique du mot par transformation des mots en vecteurs de valeurs réelles par exploration des techniques de plongement lexical, puis nous effectuons un clustering de ces vecteurs pour écarter ceux qui augmentent la fonction objective de l'algorithme de clustering. Les vecteurs qui assurent une fonction objective optimale correspondent aux mots sélectionnés pour construire le modèle.

2. Organisation du travail

La suite de ce mémoire est divisée en quatre chapitres :

- **Chapitre 1. Sélection de Caractéristiques :** dans ce chapitre nous allons présenter un aperçu général sur la sélection de caractéristiques, une revue sur quelques techniques de sélection de caractéristiques est effectuée. La deuxième partie de ce chapitre est consacrée aux travaux connexes.
- **Chapitre 2. Le plongement Lexical :** dans ce chapitre nous allons présenter le principe de fonctionnement de plongement lexical, son utilité pour la représentation de la sémantique, ainsi que les différents Frameworks qui existent la littérature.
- **Chapitre 3. Sélection de Caractéristiques par Plongement Lexical :** Nous décrivons dans ce chapitre, les différentes étapes proposées pour concevoir une approche de sélection de caractéristiques pour les données textuelles en se basant sur le plongement lexical comme technique de transformation de mots en vecteurs

numériques et le clustering comme technique non supervisée pour sélectionner les bons mots qui représentent le modèle construit.

- **Chapitre 4. Expérimentation et Evaluation des Résultats :** dans ce chapitre, nous évaluons la performance de notre approche par comparaison des résultats obtenus à ceux obtenus par d'autres approches ou d'autres algorithmes.

Chapitre 1

Sélection de Caractéristiques

(état de l’art)

1.1 Introduction

Dans ce chapitre nous allons découvrir ce qu'est la sélection de caractéristiques. Qu'il s'agit d'une partie très importante de la classification des textes, et est un processus de recherche utilisé pour l'apprentissage automatique et en traitement des données, elle sert à réduire la dimensionnalité de la représentation textuelle en modèle d'espace vectoriel (MSV) par supprimer les informations redondantes et bruitées, pour but de trouver un sous-espace représentatif pour améliorer les performances des modèles construits tout en minimisant la perte d'information. Nous allons donc présenter quelques concepts de base relatifs à la sélection de caractéristiques, puis nous ferons un état de l'art des techniques de sélection de caractéristiques. Par la suite, nous passons pour découvrir les algorithmes les plus largement utilisés dans la littérature. Puis, nous ferons une exploration non exhaustive des travaux de recherche connexes.

1.2 Définition de Sélection des Caractéristiques (en. Feature Selection)

La sélection de caractéristiques (en anglais 'Feature Selection') est un processus de recherche permettant de choisir un sous-ensemble de caractéristiques les plus pertinentes du problème étudié parmi un ensemble de départ (Figure 1.1), par éliminer les caractéristiques non-pertinentes et redondantes qui n'aiderons pas à discriminer entre les classes selon un certain critère, dans le but de réduire la dimension de l'espace des caractéristiques d'une base de données, afin d'améliorer les performances de système de reconnaissance [1].

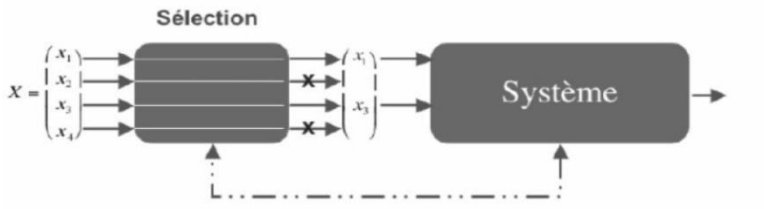

Figure 1.1: Principe de sélection de caractéristiques.

En effet, les objectifs de sélection de caractéristiques principales sont :

- Identifier les caractéristiques pertinentes et non-pertinentes selon un critère d'optimisation.
- Eliminer les caractéristiques redondantes et non-pertinentes pour réduire la dimensionnalité de la base de données d'entrée.

- Réduire les bases d'apprentissages de test.
- Améliorer les performances et la vitesse de classification.
- Diminuer le temps d'apprentissage.

La définition proposée par dans (Pudil et Novovi, 1994) [2] est la suivante :

« étant donnée une fonction permettant de mesurer la qualité d'un sous-ensemble de caractéristiques, la sélection des caractéristiques est réduite au problème de recherche d'un sous-ensemble optimal par rapport à cette mesure ».

Et dans (Jain et al., 2000) [3], la définition proposé est la suivante :

« *Étant donné un ensemble de dimension n, il faut sélectionner le sous-ensemble de dimension m tel que m<n, conduisant au taux d'erreur le plus faible* ».

1.3 Processus de Sélection de Caractéristiques

La figure 1.2, présente comment fonctionne le processus de la sélection de caractéristiques [4].

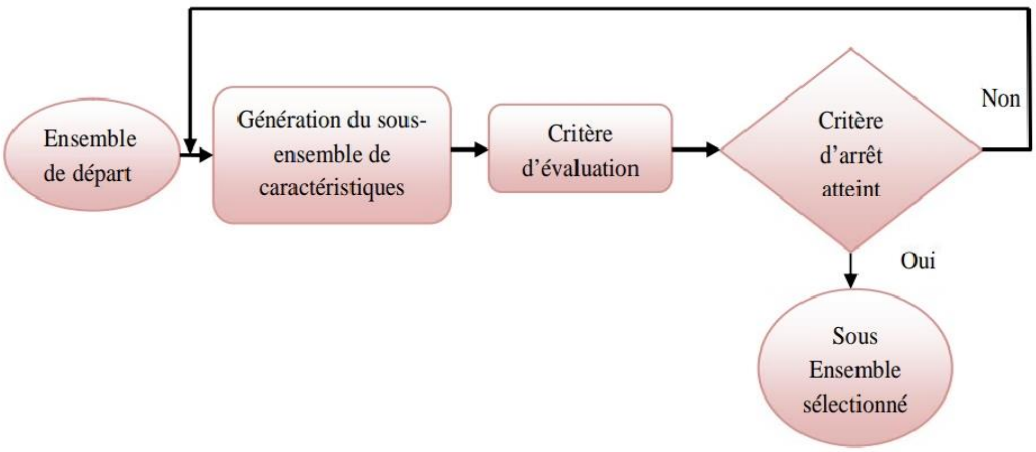

Figure 1.2: Fonctionnement de processus de sélection de caractéristiques.

- Générer successivement des sous-ensembles optimaux des caractéristiques pertinentes par supprimer séquentiellement à chaque itération les caractéristiques moins pertinentes.
- Quantifier l'importance du sous-ensemble des caractéristiques par des mesures de pertinence entre les caractéristiques.

- Puisque le nombre optimal de caractéristiques n'est pas connu apriori, alors on doit définir un critère d'arrêt pour stopper la procédure de sélection, et choisir le sous-ensemble optimal.

1.5 Apprentissage Automatique

L'apprentissage automatique est un ensemble de techniques de l'intelligence artificielle qui a pour objectif de comprendre et de reproduire les capacités d'apprentissage des machines dans des systèmes artificiels. L'objectif est de concevoir un algorithme capable d'absorber ses caractéristiques à partir d'un grand nombre d'exemples appelé base de données d'apprentissage. Le modèle obtenu de l'apprentissage est nommé classifieur, afin d'appliquer les connaissances acquises pour étiqueter des nouveaux exemples inconnus. Les étiquettes des exemples peuvent être quantitative dans les problèmes de régression et qualitative dans les problèmes de classification. Par conséquent, l'objectif fondamental de l'apprentissage automatique est de déterminer la relation entre les objets et leurs catégories afin de prédire et de découvrir des connaissances (Silva et Ribeiro, 2009). L'apprentissage est divisé en trois types: l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage semi-supervisé [5]. La figure suivante représente les différents types d'apprentissage.

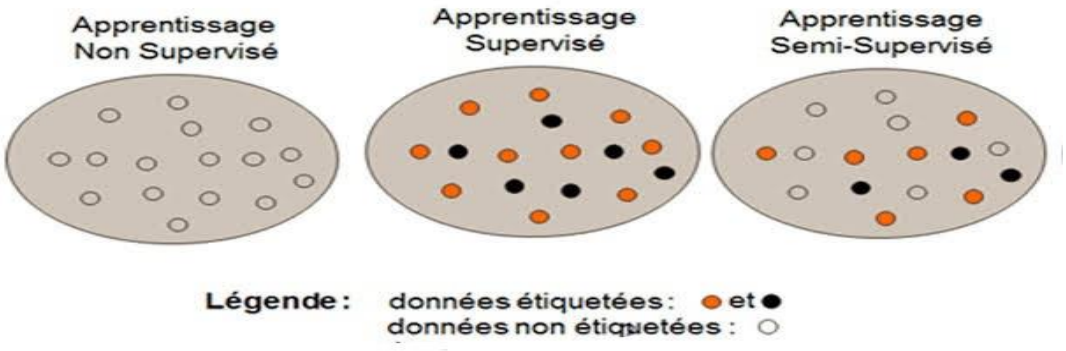

Figure 1.3: Représentation graphique de différents types d'apprentissage.

1.4.1 Apprentissage Supervisé

L'apprentissage supervisé consiste à construire un modèle basé sur un jeu de données étiquetées (nom de catégories ou des classes) [5], a pour but de créer des algorithmes aptes à recevoir des ensembles de données et à réaliser une analyse statistique pour prédire un résultat. Cette technique a été utilisée dans une variété d'applications. Dans la littérature, il

existe plusieurs techniques et algorithmes utilisés pour faire l'apprentissage supervisé tel que [5]:

1.4.1.1 La classification Bayésienne

C'est une méthode de classification statique, basé sur le théorème de Bayes avec une forte indépendance (dite naïve), qui permet de classifier un ensemble d'observations selon des règles déterminées par l'algorithme lui-même, leur particularité est de prédire la valeur des paramètres du modèle en termes de probabilité. Leur format graphique est composé de nœud et des arcs directionnels, il est appliqué dans différentes applications [5].

1.4.1.2 Support Vector Machines (SVM)

SVM est une classe d'algorithmes d'apprentissage, inventé par Vapnik et Chervonenkis en 1994, destiné à résoudre des problèmes de discrimination et de régression, qui définit pour la prévision d'une variable qualitative binaire (classification binaire), Ensuite, il a été généralisé à la prévision d'une variable quantitative (classification multi-classe) [6, 7]. Leur principe est la construction d'un hyperplan optimal, en cherchant la plus grande marge possible pour séparer les données de classes opposées. La marge est une distance entre l'hyperplan optimal et un plus proche vecteur. La figure 1.4 présente un exemple d'hyperplan.

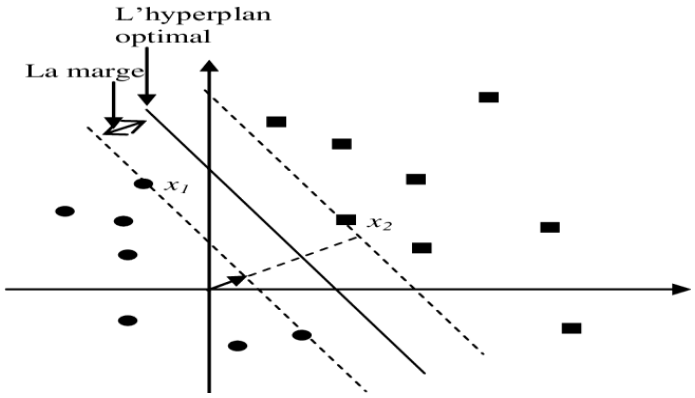

Figure 1.4: Exemple d'hyperplan optimal qui sépare deux classes.

1.4.1.3 Réseau Neuronaux

Un réseau de neurone est un algorithme d'apprentissage, proposé la première fois par les biophysiciens de l'université de Chicago, Mac Cullogh et Pitts en 1943 [8], inspirées du modèle de neurone biologique, est une technique de type induction, généralement organisé en couches, qu'ils contiennent un maillage de plusieurs neurones reliés entre eux par des liaisons

synaptiques. Il est basé sur l'expérience qui se constitue une mémoire lors de la phase d'apprentissage [9]. La figure 1.5 illustre le principe de fonctionnement d'un réseau de neurones.

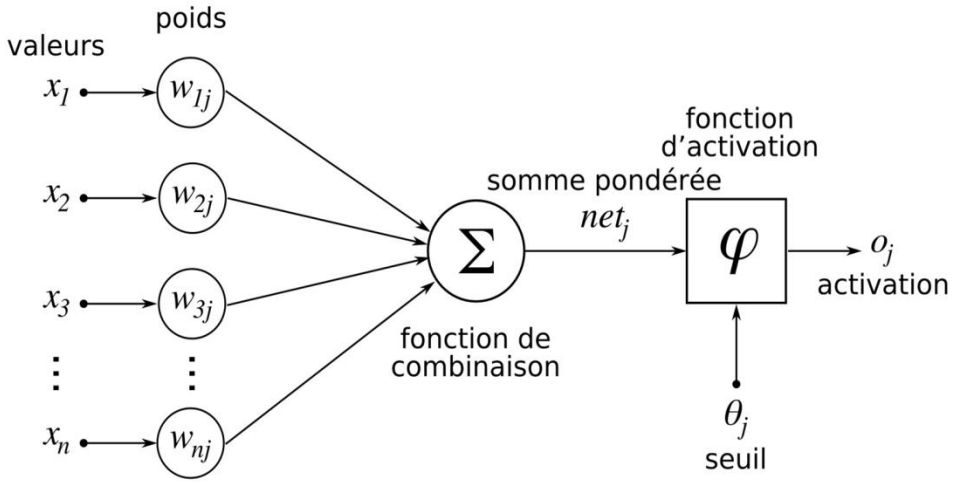

Figure 1.5: Modèle de neurones formels.

La capacité de traitement d'un réseau est stockée sous forme de poids d'interconnexions obtenus par un processus d'apprentissage à partir d'un ensemble exemples d'apprentissage [10].

1.4.1.4 Arbre de Décision (AD)

Les arbres de décision (en. 'Decision Trees') est une classe d'algorithmes d'apprentissage, un **arbre** est un graphe non orienté, acyclique et connexe. Il se compose d'un ensemble des nœuds et se divise en trois catégories : *Nœud racine, Nœuds internes et les feuilles (les nœuds terminaux).* Ils apprennent à partir d'observations qu'on appelle des exemples (attributs et classes associées), ils structurent les données sous la forme d'une séquence de décisions, par une représentation hiérarchique pour but de distinguer les similitudes et les différences entre les attributs des exemples du jeu de données [11].

1.4.1.5 Forêts d'Arbres Décisionnels (en. Random Forest Classifier - RFC)

Les forêts d'arbres décisionnels (ou forêts aléatoires), est un ensemble de classificateurs, qui à travers un sous ensemble sélectionné au hasard d'échantillons et de variables d'apprentissage, produisent plusieurs arbres de décision permettant ainsi la modélisation de chaque résultat sur une branche en fonction des choix précédents. Ensuite, en fonction de

résultat à suivre, on prend la meilleure décision [12, 5]. La figure 1.6 donne un aperçu général sur le principe de fonctionnement de l'algorithme RFC.

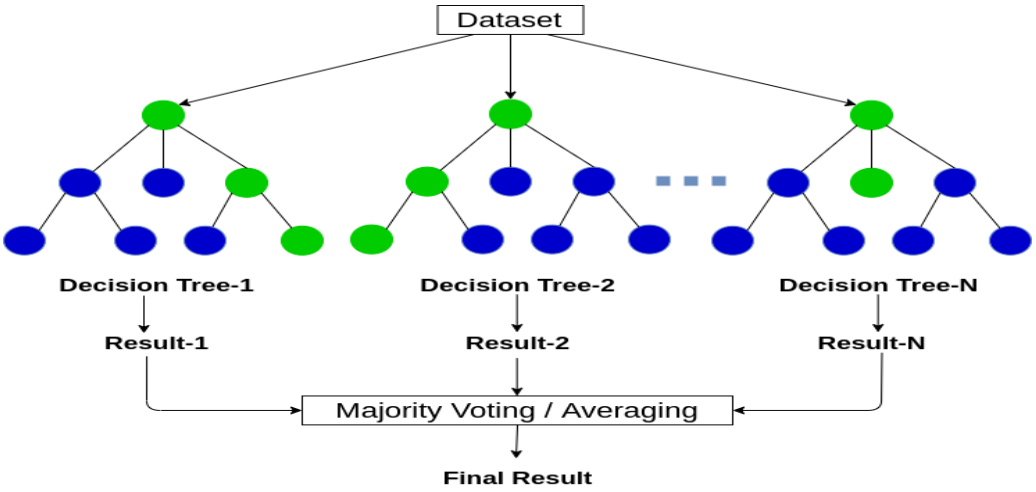

Figure 1.6: Structure de l'algorithme Random Forest.

1.4.1.6 Boosting

Le classificateur boosting a été proposé pour combiner plusieurs classificateurs faibles pour améliorer la performance de la classification, il est utilisé pour résoudre divers problèmes de classification tels que la classification des textes, traitement du langage naturel, etc., L'algorithme le plus pratiqué s'appelle AdaBoost [13].

1.4.1.7 K Nearest Neighbor (KNN)

En reconnaissance des formes, l'algorithme KNN (ou k plus proches voisins) est une méthode de classification des objets, basé sur les exemples d'apprentissage les plus proches dans l'espace de caractéristiques. KNN est un type d'apprentissage basé sur des instances. Le KNN est la technique de classification fondamentale et la plus simple lorsqu'il y a peu ou pas connaissance préalable de la distribution des données. On peut estimer la classe d'une nouvelle donnée, à partir d'une base de données libellée, en regardant quelle est la classe majoritaire des k données voisines les plus proches. Les performances d'un classificateur KNN sont principalement déterminées par le choix de K ainsi que par la métrique de distance appliquée, L'estimation est affectée par la sensibilité de la sélection de la taille de voisinage K [14].

1.4.2 Apprentissage Non-Supervisé

L'apprentissage non-supervisé consiste à extraire des groupes d'objets similaires (clusters) à partir d'une collection d'objets hétérogènes. Chaque cluster produit par ce processus doit vérifier la cohérence interne des objets qui appartient à ce cluster et l'isolation externe des objets qui appartient aux autres clusters. Le clustering a pour but de rechercher les particularités de la distribution des caractéristiques étudiées, où le système dispose en entrée une base de données non-étiquetée. De plus, la nature et le nombre de classes ne sont pas toujours connus. L'algorithme doit découvrir la structure de données cachée par lui-même.

Le processus de « clustering » est basé sur la mesure précise de la similarité des objets que nous voulons regrouper. Cette mesure est appelée distance ou métrique. Le clustering est utilisé pour diverses applications, telles que le traitement d'images, recherche démographique, recherche génétique, extraction de données et analyse d'opinion, il existe plusieurs algorithmes telle que K-means, fuzzy K-means, espérance-maximisation (EM), Regroupement hiérarchique, etc. [15].

1.4.3 Apprentissage Semi-Supervisé

L'apprentissage semi-supervisé est une forme mixte entre l'apprentissage supervisé et non-supervisé, il utilise un ensemble de données étiquetées et non étiqueté. La combinaison de données non étiquetées avec des données étiquetées peut considérablement améliorer la qualité d'apprentissage. Quand les données deviennent très volumineuses, cette opération peut être fastidieuse. Dans ce cas, un apprentissage semi-supervisé est nécessaire ou les étiquettes ont une signification pratique évidente et incontestable [15].

1.5 Techniques de Sélection de Caractéristiques

Dans cette section, nous présentons les méthodes de sélection de caractéristiques, qui peuvent être généralement répertoriées en deux grandes catégories: Supervisées et Non-Supervisées.

1.5.1. Méthodes Supervisées

Ces méthodes utilisent la variable cible (par exemple, supprime les variables non pertinentes), ils se composent par trois méthodes principales : la méthode filtrante *'filter'*, la méthode enveloppante *'wrapper'* et la méthode intégrée *'embedded'*.

1.5.1.1 Méthode Filter (Filtrage)

Pour les modèles de filtre, les caractéristiques sont sélectionnées en fonction des caractéristiques des données sans utiliser les algorithmes d'apprentissage. Le modèle filtre utilise la métrique sélectionnée pour identifier les attributs non pertinents et filtrer les colonnes redondantes du modèle. Il choisit une seule mesure statistique qui convient à donner au modèle un score pour chaque colonne de caractéristique. Les colonnes sont renvoyées et classées en fonction de leurs scores.

En choisissant les bonnes caractéristiques, il permet potentiellement d'améliorer la précision et l'efficacité de la classification.

Il n'utilise généralement que les colonnes avec les meilleurs scores pour créer le modèle prédictif. Les colonnes avec des scores mauvais peuvent être laissées dans le dataset et ignorées lorsque vous créez un modèle [16].

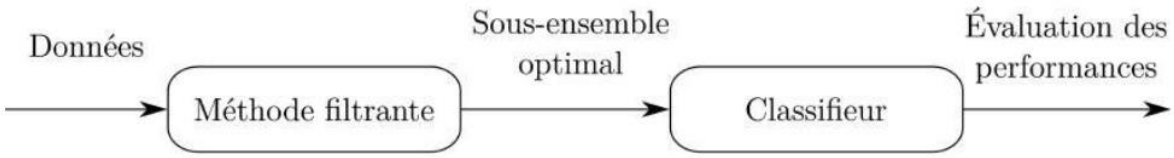

Figure 1.7 : Principe de la méthode filtrante de sélection de caractéristiques.

1.5.1.2 Méthode Wrapper

Dans les méthodes Wrapper, le processus de sélection des caractéristiques est basé sur un algorithme d'apprentissage automatique spécifique qui essaie d'adapter le dataset. Il suit approche de recherche Gourmand (en. Greedy), en évaluant toutes les combinaisons possibles de caractéristiques par rapport au critère d'évaluation (en. 'Evaluation Criterion').

Le critère d'évaluation est simplement la mesure de la performance qui dépend du type de problème. Enfin, il sélectionne la combinaison de caractéristiques qui donne les résultats optimaux pour l'algorithme d'apprentissage automatique spécifié [17].

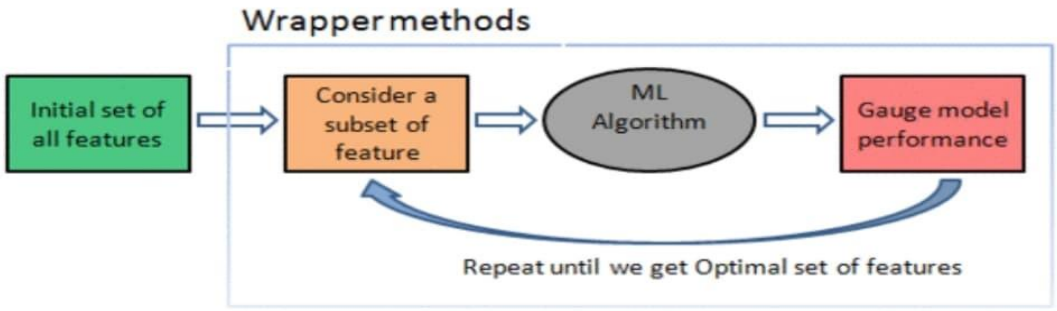

Figure 1.8 : Un cadre général des modèles Wrapper.

1.5.1.3 Méthode Embedded

A la différence des méthodes 'Wrapper' et 'Filter', les méthodes 'Embedded' (appelées aussi méthodes intégrées) incorporent la sélection des variables lors du processus d'apprentissage. Un tel mécanisme intégré pour la sélections des caractéristiques peut être trouvé, par exemple, dans les algorithmes de type SVM, AdaBoost, ou dans les arbres de décisions. Dans les méthodes de sélection de type 'Embedded', la base d'apprentissage est divisée en deux parties, une base d'apprentissage et une base de validation pour valider le sous-ensemble de caractéristiques sélectionné. En revanche, les méthodes intégrées peuvent se servir de tous les exemples d'apprentissage pour établir le système. Cela constitue un avantage qui peut améliorer les résultats. Un autre avantage de ces méthodes est leur plus grande rapidité par rapport aux approches 'Wrapper' parce qu'elles évitent que le classificateur recommence de zéro pour chaque sous-ensemble de caractéristiques [18].

1.5.2 Méthodes Non-Supervisé

Les méthodes non-supervisé sont généralement utilisés pour les tâches de regroupement**,** ils sont très similaires à la sélection de caractéristiques supervisée, sauf qu'il n'y a aucune information d'étiquette impliquée dans la phase de sélection de caractéristiques et la phase d'apprentissage du modèle. Sans l'étiquette pour définir la pertinence de la caractéristique, la sélection de caractéristiques non supervisée repose sur un autre critère alternatif pendant la phase de sélection de caractéristiques. Un critère couramment utilisé choisit les caractéristiques qui peuvent le mieux préserver la structure multiple des données d'origine.

Une autre méthode fréquemment utilisée est de rechercher des indicateurs de cluster à travers les algorithmes de clustering, puis, transformer la sélection de caractéristiques non supervisée en un cadre supervisé. Il existe deux manières différentes d'utiliser cette méthode.

Une façon consiste à rechercher des indicateurs de cluster et à effectuer simultanément la sélection de caractéristiques supervisée dans un cadre unifié. L'autre façon consiste à rechercher d'abord des indicateurs de cluster, puis à effectuer une sélection de caractéristiques pour supprimer ou sélectionner certaines caractéristiques. Et enfin de répéter ces deux étapes de manière itérative jusqu'à ce qu'un certain critère soit rempli. De plus, certains critères de sélection de caractéristiques supervisés peuvent toujours être utilisés avec quelques modifications [18].

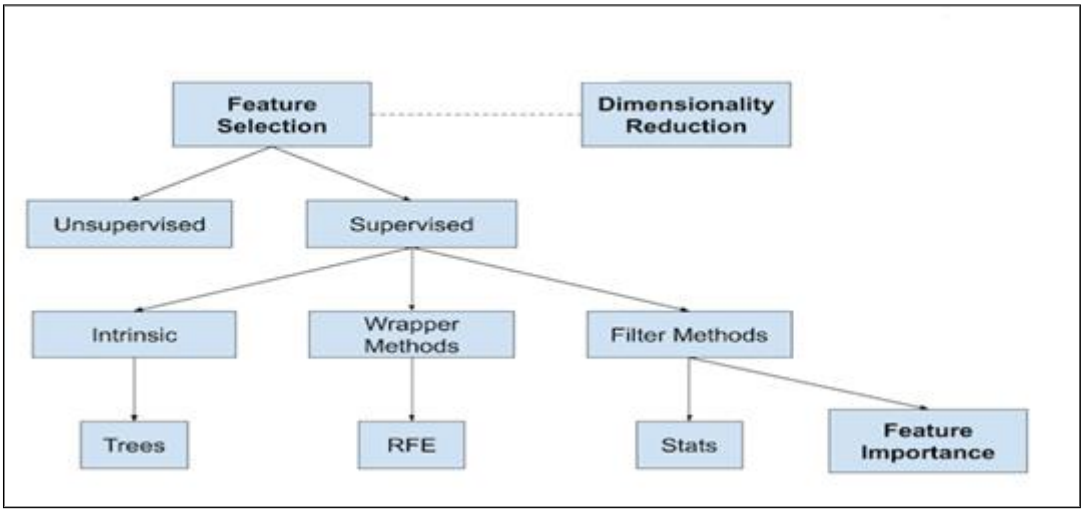

Figure 1.9 : Organigramme de techniques de sélection de caractéristiques.

1.6. Algorithmes de Sélection des Caractéristiques

Dans cette section nous présentons quelques algorithmes de sélection de caractéristiques existant dans la littérature.

1.6.1. Document Frequency (DF)

DF est une méthode de sélection de caractéristiques non supervisée simple et efficace, qui note les caractéristiques en fonction du nombre d'apparitions dans le document, C'est-à-dire des caractéristiques plus fréquentes sont plus importants. DF est calculé en comptant le nombre de documents dans lesquels une caractéristique apparaît. Un inconvénient évident est que certains termes à haute fréquence qui ne sont pas utiles pour la classification, seront

comptés comme des caractéristiques [19, 20]. Document Frequency d'un terme est calculé comme suit :

$$\boldsymbol{DF(t) = Nombre\ de\ documents\ contenant\ t}$$

1.6.2. Gini Index (GI)

Gini Index est une méthode non-supervisé de division non-pure et largement utilisée dans les algorithmes d'arbres de décision. Shang et al. (2007) ont proposé la méthode améliorée de Gini Index pour l'appliquer directement au sélection de caractéristiques de texte. Il mesure la pureté de la caractéristique t_k vers une catégorie C_i. Plus la valeur de pureté est élevée, meilleure est la caractéristique [21]. La formule de Gini Index est définie comme suit :

$$\boldsymbol{Gini(t_k) = P(t_k)\left(1 - \sum_{i=1}^{m} P(c_i/t_k)\right)^2 + P(\bar{t_k})\left(1 - \sum_{i=1}^{m} P(c_i/\bar{t_k})\right)^2}$$

Où m est le nombre de classe, $P(t_k)$ et $P(\bar{t_k})$ indiquent respectivement la probabilité de présence et d'absence du terme t_k dans la classe c_i. $P(c_i/t_k)$ et $P(c_i/\bar{t_k})$ sont respectivement la probabilité conditionnelle de c_i sachant que t_k a été observé dans c_i, et la probabilité conditionnelle de c_i sachant que t_k n'a pas été observé dans c_i.

1.6.3. Information Gain (IG)

Information Gain (IG) (Quinlan, 1986) est une méthode supervisée utilisé comme critère dans le domaine de l'apprentissage automatique (Yang & Pedersen, 1997), Information Gain d'un terme t_k par rapport à la classe c_i est la réduction de l'incertitude sur la valeur de ci quand on connait la valeur de t_k [19]. Information Gain d'une caractéristique t_k vers catégorie c_i, est calculé comme est la différence d'entropie avant et après la division de l'ensemble de données par rapport le terme t_k :

$$IG(t_k) = H(c) - H(c/t_k)$$

$$= -\sum_{j=1}^{m} p(c_j) \times \log\left(p(c_j)\right) + p(t_k) \sum_{j=1}^{m} p(c_j/t_k) \times \log\left(p(c_j/t_k)\right)$$

$$+ p(\bar{t_k}) \sum_{j=1}^{m} p(c_j/\bar{t_k}) \times \log\left(p(c_j/t_k)\right)$$

Où m est le nombre de classes, $p(c_j)$, est la probabilité d'un document appartenant à la classe c_j. $p(t_k)$ et $p(\bar{t_k})$ sont les probabilités d'un document dans le corpus inclut la présence et l'absence du terme t_k. $p(c_j/t_k)$ et $p(c_j/\bar{t_k})$ sont les probabilités conditionnelles de la classe c_j étant donné la présence ou l'absence du terme t_k, respectivement.

1.6.4. Mutuel Information (MI)

MI (Fano, 1961) est une méthode supervisée, couramment utilisé dans la modélisation statistique du langage des associations de mots et applications connexes, qui représente la corrélation entre les classes et les caractéristiques [22]. Elle est calculée par la formule suivante :

$$\mathbf{MI}\ (t, C_i) = \log \frac{P(t|\ Ci)}{P(t)}$$

Pour calculer le MI global du terme t, on utilise **MI (t) = maxi MI(t, Ci)**, pour représenter le poids global du terme t [20].

1.6.5. Chi-square

X^2 Chi-square (Yang & Pedersen, 1997) est une méthode de sélection des caractéristiques supervisée, utilisée pour tester l'indépendance de deux variables statistiques, par calcul de la corrélation du terme t avec la classe C_i. Si $X^2(C_i,t_k)$ = 0 alors la caractéristique t_k et le classe c_i sont indépendantes, donc la caractéristique t_k ne contient aucune information sur la classe. La formule de Chi-square est définie comme suit :

$$\chi^2(t_k, c_i) = \frac{N(a_{ki}d_{ki} - b_{ki}c_{ki})^2}{(a_{ki} + b_{ki})(a_{ki} + c_{ki})(b_{ki} + d_{ki})(c_{ki} + d_{ki})}$$

Où N est le nombre total de documents (textes), a_{ki} est la fréquence quand la caractéristique t_k et la catégorie Ci existe, b_{ki} est la fréquence quand la caractéristique t_k existe et n'appartient

pas à la catégorie C_i, c est la fréquence quand la catégorie existe et ne contient pas la caractéristique t_k, d est le nombre de fois où et existe [19].

1.7. Travaux Connexes

Plusieurs travaux similaires ont été réalisés dans le domaine de sélection des caractéristiques, durant ce travail, nous avons eu l'occasion d'aborder plusieurs projets liés à notre travail de recherche. Nous présentons dans ce qui suit les différents travaux dans ce contexte qui sont basés sur les techniques définies précédemment.

1.7.1 Basés sur la méthode Filter

Parce que les méthodes de filtrage, qui n'utilisent pas de méthodes d'apprentissage et ne prennent en compte que la pertinence entre les caractéristiques, ont une faible complexité temporelle. De nombreux chercheurs se sont concentrés sur ces méthodes. Parmi ces chercheurs Moradi et Rahamninia [23], ils ont proposé les deux méthodes OSFSMI et OSFSMI-K, pour objectif de sélectionner les caractéristiques les plus informatives de la diffusion en ligne. Les deux méthodes basées sur la méthode Mutuel Information (MI) de manière continue pour évaluer la corrélation entre les caractéristiques et également pour évaluer la pertinence et la redondance des caractéristiques dans des tâches de classification complexes. Les méthodes proposées n'utilisent aucun modèle d'apprentissage dans leur processus de recherche et peuvent donc être classées comme des méthodes basées sur des filtres.

Ainsi, les chercheurs Jinxing et al., [24], ils ont proposé une méthode de sélection de caractéristiques basée sur Mutuel Information (MI) et la normalisation de la pertinence maximale et de la redondance commune minimale (N-MRMCR-MI), qui produit une valeur normalisée dans la plage [0, 1] et entraîne un problème de regression.

1.7.2 Basés sur la méthode Wrapper

Ces méthodes évaluant l'utilité des caractéristiques sélectionnées à l'aide des performances de l'apprenant. Les chercheurs Sanz et al., ont proposés une méthode appelée SVM-REF (Selection and visualisation of the most relevant features through non-linear kernels) [25]. Cette approche étend RFE pour visualiser l'importance des variables dans le contexte de la SVM avec noyaux non linéaires pour les réponses de survie. La méthode utilise

deux variantes des algorithmes RFE basés sur la représentation des variables dans un espace multidimensionnel tel que l'espace KPCA.

Dans une étude distincte, une méthode de sélection des caractéristiques a été proposée par les chercheurs Viegas et al., [26], basée sur la programmation génétique, qui est résiliente aux problèmes d'asymétrie des données, en d'autres termes, elle fonctionne bien avec les données équilibrées et déséquilibrées. Cette stratégie vise à combiner les ensembles de caractéristiques les plus discriminants sélectionnés par des métriques de sélection de caractéristiques, afin d'obtenir un ensemble plus efficace des caractéristiques les plus discriminantes, partant de l'hypothèse que des métriques de sélection de caractéristiques distinctes produisent des caractéristiques différentes (et potentiellement complémentaires).

Les méthodes de sélection de caractéristiques améliorent non seulement les performances du modèle, mais facilitent également l'analyse des résultats, une étude examine l'utilisation des SVM dans les problèmes multi-classes, où les chercheurs Izetta et al., [27], ont proposés six méthodes basées sur une combinaison de listes, conçues pour produire des sélections améliorées à partir des importances données par les problèmes binaires. Ils les évaluent sur des ensembles de données artificiels et réels, ils sont utilisés à la fois les stratégies One-Vs-One (OVO) et One-Vs-All (OVA).

1.7.3 Basés sur la méthode Embedded

Parmi quelques travaux connexes qui sont classés dans la méthode Embedded, un algorithme de sélection de caractéristiques semi-supervisé qui est proposé par les chercheurs Xiao et al., [28], pour traiter une classification de charge électrique, basée sur la technologie de la méthode de traitement des données en groupe (GMDH). L'algorithme proposés se compose de trois étapes principales :

- Apprentissage du classificateur de base.
- Marquer sélectivement les échantillons les plus appropriés à partir des données d'étiquettes non classifiées et les ajouter à un ensemble d'apprentissage initial.
- Entraîner les modèles de classification sur l'ensemble d'apprentissage final et classer les échantillons de test.

Ainsi, les chercheurs Liu et al., [29], ont proposé de sélectionner les caractéristiques en se basant sur leur capacité de distinction. Tout d'abord, ils ont défini le concept de voisin le plus proche, puis ils ont utilisé ce concept pour discriminer les voisins les plus proches des

échantillons. Ils ont présenté une nouvelle méthode de mesure pour évaluer la qualité des caractéristiques. Enfin, l'algorithme proposé est testé sur des ensembles de données de référence.

1.8 Conclusion

La sélection des caractéristiques est un processus efficace pour prétraiter et réduire la dimensionnalité des données et un élément essentiel du succès des applications d'exploration de données et d'apprentissage automatique, elle a pour but de créer des modèles plus simples et plus compréhensibles, améliorer les performances de l'analyse de données et aider à préparer, nettoyer et comprendre les données. En plus, il est considéré comme un sujet de recherche vaste avec une importance pratique dans de nombreux domaines tels que les statistiques, reconnaissance de formes, apprentissage automatique et analyse de données.

Dans ce chapitre, nous avons fourni un aperçu général et structuré sur la sélection des caractéristiques.

Chapitre 2

Plongement Lexical
(Word Embedding)

2.1 Introduction

La recherche en informatique sur la sémantique lexicale et le changement sémantique est devenue de plus en plus populaire en raison du développement des techniques de plongement lexical. Le plongement lexical (en. Word Embedding) est une forme de sémantique distribuée. Elle a pour but de représenter une sémantique lexicale à l'aide des vecteurs numériques pour mesurer la similarité entre les mots par des métriques de distance, par exemple on pourrait s'attendre à ce que les mots « table » et « chaise » soient représentés par des vecteurs relativement peu distant dans l'espace vectoriel où sont définis ces vecteurs.

Le plongement contient des techniques classiques TF, TF-IDF, Binary Encodeding, etc, qui est reposées sur des méthodes de réduction de dimensionnalités, et d'autres qui sont récentes, ils sont basés sur des modèles probabilistes et des réseaux de neurones qui ont prouvé leur efficacité en capturant la sémantique des mots comme word2vec.

2.2 Méthodes de Pondération

Les méthodes de pondération visent à représenter le mot par une valeur numérique discrète (poids), dans cette section nous présentons les méthodes suivantes :

2.2.1 BOW (Bag of Words)

Un modèle de sac de mots, ou BOW, est la méthode la plus simple de représenter un texte par des mots isolés du vocabulaire dans un format numérique compréhensible par la machine. En comptant le nombre de fois qu'un mot apparait dans le document. Ces nombres de mots nous permettent de comparer des documents et d'évaluer leur similarité pour des applications telles que la recherche, la classification de document, etc.

BOW répertorie les mots associés à leurs nombres de mots par document. Dans le tableau où sont stockés les mots et les documents qui deviennent effectivement des vecteurs, chaque ligne est un mot, chaque colonne est un document et chaque cellule représente le nombre de mots dans le document. Chacun des documents du contexte est représenté par des colonnes de longueur égale. Ce sont des vecteurs de nombre de mots [30, 31].

2.2.2 TF (Term Frequency)

TF mesure la fréquence d'apparition d'un terme dans un document. Etant donné que chaque document est de longueur différente, il est possible qu'un terme apparaisse beaucoup

plus de fois dans les documents longs que dans les plus courts. Ainsi, la fréquence du terme ***t*** est souvent divisée par la longueur du document ***d*** (c'est-à-dire le nombre total de termes dans le document) [32].

$$\boldsymbol{TF(t,d) = \frac{N_{t,d}}{N}}$$

Ici, $\boldsymbol{N_{t,d}}$ est le nombre de fois où le terme ***t*** apparaît dans le document ***d***. N est le nombre total des termes dans le document. Ainsi, chaque document et terme aurait sa propre valeur TF [33].

2.2.3 TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF est une autre façon de représenter le document par les mots qu'il contient. Il donne à chaque mot un poids pour mesurer sa pertinence, pas la fréquence. Autrement dit, il remplace le nombre de mots par des scores.

Premièrement, TF-IDF mesure le nombre d'occurrence de mot dans un document donnée (Term Frequencey), mais comme des mots tels que « et » ou « le » apparaissent fréquemment dans tous les documents, ceux-ci doivent être écartés. C'est la partie de IDF (Inverse-Document Frequency), le mot qui apparaît fréquemment dans un grand nombre de documents est le moins pertinent pour différencier un document donné. Cela vise à ne laisser que les mots fréquents et distinctifs comme marqueurs [34].

Nous pouvons calculer le score TF-IDF pour chaque mot. Les mots avec un score plus élevé sont plus importants, et ceux avec un score plus faible sont moins importants [33].

$$\boldsymbol{TFIDF(t,d) = TF(t,d) \times IDF(t,d)}$$
$$\boldsymbol{= TF(t,d) \times log\left(\frac{N}{DF(t)}\right)}$$

Où, $\boldsymbol{TF(t,d)}$, est le nombre d'occurrences du terme i dans le document j.

$\boldsymbol{DF(t)}$, est le nombre de documents qui contient le terme i. $\boldsymbol{N:}$ Le nombre total de documents.

2.3 Méthodes de Plongement Lexical

Les méthodes de plongement lexical visent à représenter le mot par un vecteur de valeurs réelles, dans cette section nous présentons les méthodes les plus largement utilisées :

2.3.1 word2vec

Word2vec est un groupe de modèles de représentation distribuée des mots, créé par Thomas *Mikilov* et développé par une équipe de recherche GOOGLE, il s'agit d'un réseau de neurone avec trois couches, qui a pour but de représenter les données d'entrée (mots) par des vecteurs multidimensionnels. Il comprend deux types de modèles: le modèle CBOW et le modèle Skip-Gram [35, 36], leurs structures sont illustrées sur la figure 2.1.

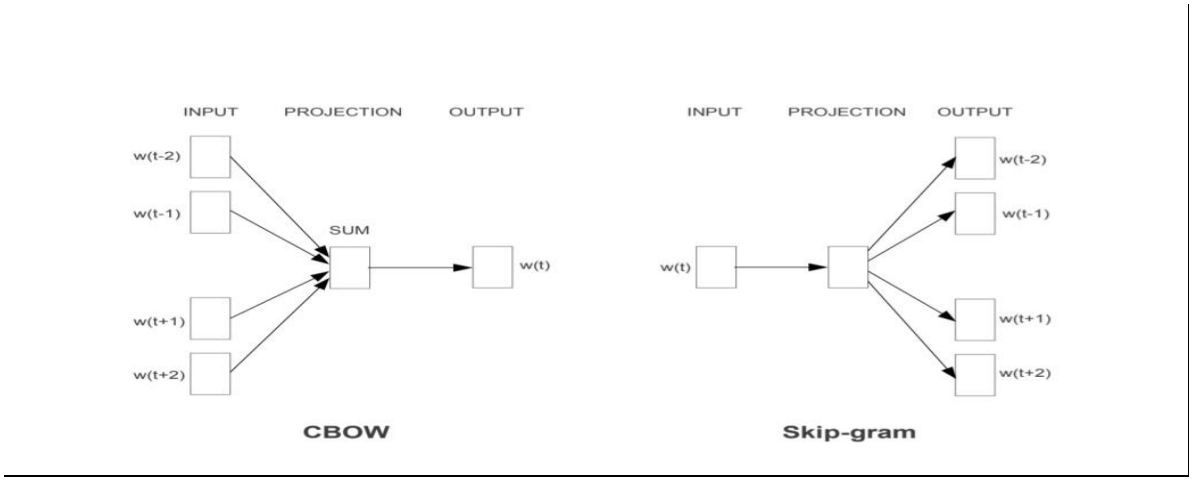

Figure 2.1 : Modèle CBOW vs Modèle Skip-Gram.

Le modèle CBOW (Continous Bag of Words) et le modèle Skip-Gram contiennent trois couches, une couche d'entrée, une couche de projection et une couche de sortie. Le CBOW a pour but de prédire un mot central W(t) à partir des mots de voisinage W(t-n),…, W(t-1), W(t+1),…, W(t+n), ou n est la taille de la fenêtre du contexte. Tandis que le modèle Skip-Gram a pour but de prédire les mots de voisinage à partir d'un mot central [37].
Les deux modèles constituent la représentation des termes dans un nouvel espace.

2.3.1.1 Skip-Gram (SG)

Le modèle Skip-Gram prédit le contexte (les mots voisins) basés sur un mot cible, dans le processus SG trois couches, une couche d'entrée qui correspond au mot cible, une couche de

sortie qui correspond au contexte et la couche cachée qui correspond à la projection du mot cible de la couche d'entrée.

En effet, à partir d'un mot donné, Skip-Gram cherche la prédiction d'un contexte, ensuite il fait une comparaison entre le mot d'entrée avec chaque mot de sortie, a pour but de corriger sa représentation en fonction de la propagation arrière du gradient d'erreur [38], en maximisant la fonction objective suivante :

$$S = \sum_{t=1}^{v} \sum_{j=t-C, j\neq t}^{t+C} \log\, p(m_{t+j} \mid m_t)$$

Où v (et c) est la taille de vocabulaire (contexte).

La structure de Skip-Gram est illustrée dans la figure 2 ci-dessus :

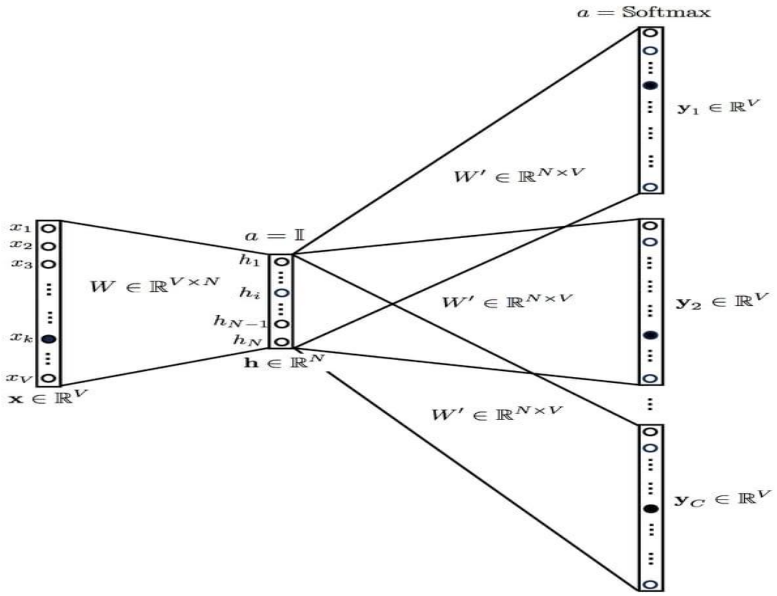

Figure 2.2 : Le modèle Skip-Gram.

2.3.1.2 Continous Bag-of-Word (CBOW)

C'est une combinaison de techniques BOW et N-Gram. L'idée de BOW repose sur l'utilisation d'un certain nombre de mots dans le texte et mise des nombres 1 et 0 en fonction de la présence de chaque mot, et l'idée de N-Gram repose sur l'utilisation d'un mot ou

plusieurs à partir les mots précédents pour déduire le mot suivant. Ainsi, en utilisant CBOW, nous pouvons utiliser plus d'un mot dans la même phrase pour déduire un mot candidat**.**

Le modèle CBOW, c'est le contraire de modèle Skip-Gram, il vise à prédire un mot en fonction de son contexte, Cela se fait en calculant le Embedding Matrix, il se compose de trois couches, la couche d'entrée correspond au contexte, la couche de projection est partagée sur tous les mots d'entrée, chaque mot est projeté sur la même position des autres mots voisins dans la couche de sortie [39]. A la fin, ce modèle corrige la représentation du mot en fonction de la propagation arrière du gradient d'erreur, en faisant une comparaison du mot et sa sortie [38] en maximisant la fonction objective suivant :

$$\mathbf{S=}\ \sum_{i=1}^{v} \mathbf{log\ p}(\boldsymbol{m_i} \mid \boldsymbol{m_{i-n}}, \ldots, \boldsymbol{m_{i-1}}, \boldsymbol{m_{i+1}} \ldots, \boldsymbol{m_{i+n}})$$

Où v : correspond au nombre de mots dans le contexte.

Il faut choisir un nombre précis de mots qui seront utilisés pour extraire le mot candidat.

La structure de CBOW est illustrée dans la figure 2.2 :

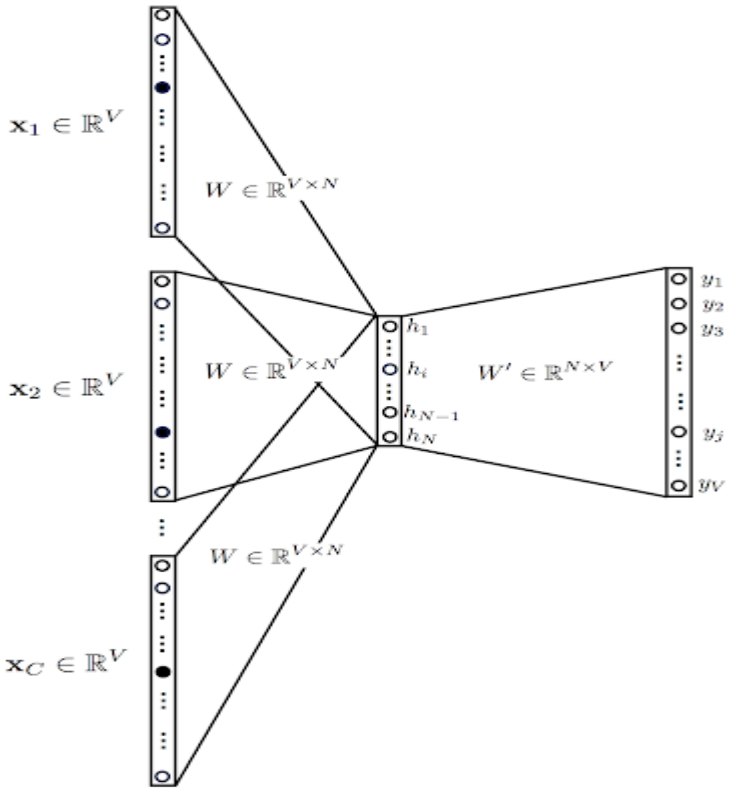

Figure 2.3 : Le modèle CBOW [30].

2.2.2 GLoVe (Global Vectors for Words Representations)

GloVe (Vecteur GLObal) est un modèle d'apprentissage non supervisé, proposée par [Pennington et al., 2014], est une approche récente, qui réunit deux approches : la factorisation de matrice « count-based », et les modèles prédictifs ou neuronaux. Cette approche est considérée comme modèle prédictif par [Levy et al., 2015, Li et Jurafsky, 2015] et comme modèle « count based » par [Arora et al., 2016].

Cette approche repose sur la construction d'une matrice MG de co-occurrence globale des mots, en traitant le corpus en utilisant une fenêtre contextuelle glissante. Ici, chaque élément MG_{ij} représente le nombre d'apparence de mot m_j dans le contexte du mot m_i.

Une fois la matrice MG calculée, Il construit des représentations vectorielles $\overrightarrow{m_i}$ et $\overrightarrow{m_j}$ en entraînant un modèle de régression des moindres carrés. Ces représentations doivent conserver des informations utiles sur la co-occurrence du pair de mots m_i et m_j, telles que :

$$\overrightarrow{m_i}^T \overrightarrow{m_j} + b_i + b_j = logMG_{ij}$$

Où b_i et b_j sont les vecteurs biais associés respectivement pour les mots m_i et m_j. Ceci est accompli en minimisant la fonction du coût, qui évalue la somme des erreurs au carré :

$$E = \sum_{i,j=1}^{nv} f(MG_{ij})(\overrightarrow{m_i}^T \overrightarrow{m_j} + b_i + b_j - logMG_{ij})^2$$

Où *nv* est la taille du vocabulaire, f(.) est une fonction de pondération, qui pondère le coût en fonction de la fréquence du nombre de co-occurrence MG_{ij}. Celle-ci est définie comme suit :

$$f(MG_{ij}) = \begin{cases} (\frac{MG_{ij}}{MG_{max}})^{\alpha} & si\ MG_{ij} < MG_{max} \\ 1 & sinon \end{cases}$$

Où MGmax = 100 et α = 3/4. Lorsque la valeur de co-occurence MG_{ij} d'une paire de mots est très élevée, c'est-à-dire supérieure à une valeur maximale MGmax, la fonction ne prend pas en compte cette valeur et simplement retourne 1. Sinon, pour les autres paires, elle envoie un poids entre (0 et 1), où la distribution de poids dans cette plage est décidée par α [38].

2.2.3 LSA (Latent Semantic Analysis)

LSA est l'une des méthodes les plus utilisées pour la représentation sémantique, basé sur un type d'analyse factorielle, signifie analyser des documents pour trouver leurs significations. Dans cette approche, nous transmettons un ensemble de documents d'apprentissage et définissons un nombre possible de concepts qui pourraient exister dans ces documents. Le résultat de cette LSA est essentiellement une matrice de termes en concepts [40].

2.2.4 BERT (Bidirectional Encoder Representations from Transformers)

BERT (Devlin et al.,2018) est un modèle neuronal de plongements lexicaux, vise à produire des représentations de mots qui dépendent du contexte, pré-entrainé sur deux types de tâches :

- Une tâche de Modalisation du Langage Masquée (Masked Language Model MLM).
- Une tâche de Prévision de la Phrase Suivante (Next Sentence Prediction NSP).

Dans deux phases différentes :

- La phase de pré-entrainement : via les deux taches ci-dessus, il permet au modèle d'apprendre à produire des plongements contextualisés.
- La phase d'adaptation a une tache : BERT génère des plongements au sein d'un modèle plus large qui est intégralement entrainé sur une tâche cible [41].

2.2.5 ELMO (Embeddings Encoder Representations from Transformers)

ELMO (Peters et al.,2018), c'est une modèle de nouvelle génération plus spécifique que BERT, vise à représenter les mots en fonction de leurs contexte général (dans le corpus) et le contexte local (dans une phrase particulier), ce modèle basé sur un réseau de neurone LSTM bidirectionnel a pour but de prédire le mot suivant dans la phrase, et quand donné la fin d'une phrase prédire le mot qui venant juste avant. Le but de ELMO est d'utiliser des plongements lexicaux des phrases en entré de modèles neuronaux spécifiques à certaines tâches, n'est pas pour le produire.

Les vecteurs de mots qui directement issus de la première ou la deuxième couche testent sa performance pour but de chercher du premier proche voisin [42].

2.3 Comparaison entre les modèles de plongement lexical

2.3.1 BOW vs Word2vec

La principale différence entre BOW et Word2Vec est que BOW produit un seul nombre par mot, le contraire de Word2Vec qui produit un vecteur par mot. Word2Vec est idéal pour fouiller dans des documents et identifier son contenu, ses vecteurs représentent le contexte de chaque mot.

BOW est une simple méthode pour classer les documents dans leur ensemble [43].

2.3.2 TF vs TF-IDF

La différence entre TF et TF-IDF est de savoir si les fréquences des mots sont utilisées ou non. Le TF / IDF est de loin un meilleur choix, indépendamment du classificateur. Quand en utilisant uniquement TF, nous ne nous soucions pas vraiment de savoir si un mot est commun ou non. Ainsi, des mots courants comme par exemple, les articles définis et indéfinis (un, une, le, la, etc.) reçoivent un poids important même s'ils ne fournissent aucune information réelle [43].

2.3.3 LSA vs GLoVe

LSA une matrice de co-occurrence construite à partir des vecteurs de mots, elle est essentiellement basée sur le corpus global réalisé en utilisant la décomposition matricielle SVD, SVD mais de grande complexité.

GLoVe peut être considéré comme une matrice LSA à efficacité optimisée [43].

2.3.4 Word2vec vs BERT

Glove et Word2vec sont des modèles non supervisés pour générer des vecteurs de mots. La différence entre eux est le mécanisme de génération de vecteurs de mots. Les vecteurs de mots générés par l'un ou l'autre de ces modèles peuvent être utilisés pour une grande variété de tâches [43].

2.3.5 Word2Vec vs LSA

Word2vec permet de créer des représentations vectorielles de mots dans un espace sémantique qui capture en partie la sémantique du mot. Partant de l'hypothèse que le sens général d'un texte est défini par la combinaison du sens des mots qui le composent, l'idée générale consiste à considérer la représentation vectorielle d'un texte sous la forme d'une combinaison des vecteurs des mots qui le forment. L'approche naïve se borne à effectuer la moyenne des vecteurs de mots obtenus par une représentation word2vec. Cette moyenne est une estimation de l'idée générale du texte. Cependant, toute approche basée sur un moyennage est sensible aux valeurs extrêmes, peu pertinentes dans le cas de données très asymétriques et souvent, la moyenne calculée ne correspond à aucune des valeurs observées. Par ailleurs, à partir de la moyenne word2vec, il est difficile, voire impossible, de remonter aux mots qui forment le texte. En contrepartie, une représentation vectorielle type LSA conserve la connaissance des occurrences des mots importants qui caractérisent les textes. C'est cette complémentarité des deux approches qui motive l'approche proposée. En concaténant la moyenne des vecteurs word2vec et la représentation vectorielle des textes fournis par LSA, nous obtenons une représentation vectorielle en basse dimension au niveau du document qui capture la sémantique générale d'un texte tout en conservant une description lexicale/conceptuelle du document. A notre connaissance, cette combinaison n'a pas été explorée [43].

2.4 Conclusion

Dans ce chapitre nous avons présenté les techniques de pondération et les méthodes de plongement lexical, parmi ces méthodes de plongement lexical nous avons examiné word2vec que nous considérons comme un point de départ de notre travail.

Les autres étapes seront expliquées dans le chapitre suivant, où nous allons créer notre propre système de sélection de caractéristiques.

Chapitre 3

Sélection de Caractéristiques par Plongement Lexical

3.1 Introduction

La sélection de caractéristiques est généralement définie comme un processus de recherche permettant de trouver un sous-ensemble de caractéristiques parmi celles de l'ensemble de départ. Il existe Plusieurs techniques classiques de sélection de caractéristiques comme Information Gain (IG), Gini Index (GI), CH2 (Chi-Square), IMGI (Improved Gini Index), etc., qui ont prouvé leur efficacité de trouver des bons espaces représentatifs pour les corpus textuels. Mais malheureusement, ces techniques traitent le mot comme une unité isolée sans prendre en considération sa sémantique en considérant seulement sa fréquence.

Dans ce chapitre, nous allons proposer une architecture en se basant sur le freamwork word2vec pour transformer les mots en vecteurs de valeurs réels, puis nous effectuons un clusternig de ces vecteurs pour écarter ceux qui augmentent la fonction objective de l'algorithme de clustering, où les vecteurs qui assurent une fonction objective optimale correspondent aux mots sélectionnés pour construire le modèle.

3.2 Prétraitement du Dataset

La construction d'un modèle word2vec est basée principalement sur un jeu d'apprentissages. Dans notre cas le dataset est un corpus textuel (des données catégorielles). Il est nécessaire de le prétraiter, c'est l'étape la plus importante pour développer le modèle avec de bonne performance, il consiste à préparer les données avant de les fournir à la machine pour l'apprentissage, pour but de mettre les données dans un format propice mais également d'avoir le dataset le plus propre possible afin d'améliorer la performance de notre modèle, pour ça il existe une tonne d'opération possibles parmi eux :

- Normalisation : qui permet de mettre sur une même échelle toute les variables quantitatives ce qui facilite beaucoup l'apprentissage de la machine.
- Les mots outils sont supprimés : avec, encore, etc.
- Les mots de la même racine sont concaténés (par exemple, réduit, réduite, réduire, etc).
- Supprimer les signes de ponctuation (point, virgule, point-virgule, etc).
- Supprimer les stop-words.
- Supprimer les mots non-ascii.
- Les conjugaisons des verbes sont concaténées.
- Changer les nombres en lettres.

- Changer les lettres majuscules en minuscules.

Ce sont les différents types d'étapes de prétraitement de texte que nous pouvons effectuer sur nos données de texte.

3.3 Construction du Modèle Word2Vec (CBOW)

Pour rappel, un modèle est une **représentation** simplifiée de la réalité, le modèle qui nous avons construire est en quelque sorte le cœur de notre travail. Nous considérons le Wor2Vec (CBOW) le modèle que nous allons construire, qui prend d'abord les textes d'apprentissage pour construire le vocabulaire, puis apprend la représentation vectorielle des mots pour capturer la similarité entre les eux par des mesures de distance, et cela pour prédire le mot candidat à partir d'un ensemble des mots de contexte. Néanmoins, les deux versions de word2vec, Skip-Gram et CBOW ont des performances similaires, la seule différence est que lorsque le corpus textuel est petit Skip-Gram fonctionne mieux contrairement à CBOW qui donne des bons résultats pour des gros corpus textuels, et puisque dans notre cas on a un grand corpus textuel alors nous avons choisi le modèle CBOW.

3.4 Sélection des Caractéristiques par le biais de Word2Vec

Maintenant que nous savons comment transformer les mots en vecteurs de valeurs réels dans un espace vectoriel grâce à Word2Vec et générer notre modèle, nous pouvons potentiellement utiliser l'algorithme de Clustering pour effectuer la sélection.

3.4.1 Clusternig de l'espace vectoriel

Comme nous l'avons déjà mentionné dans le chapitre 1, le clustering est un type d'apprentissage non supervisée. Il consiste à regrouper le jeu de données en K groupes, en fonction des attributs (caractéristiques) qui les décrivent. C'est-à-dire regrouper les données qui sont similaires et séparer les données qui sont dissimilaires, comme ça on a en sortie des groupes appelés clusters**,** qui vont contenir des éléments qui partagent les mêmes caractéristiques. L'idée est donc de découvrir des groupes au sein des données, de façon automatique [44].

Il existe plusieurs méthodes de clustering et pour chaque méthode, il est nécessaire de choisir comment mesurer la similarité entre deux individus, il nous faut donc une fonction de distance.

3.4.1.1 Mesures de Distance

La distance joue un rôle important dans l'apprentissage automatique, il fournisse la base de l'algorithme de clustering K-means, c'est une application qui associe un couple de vecteurs à un nombre réel positif. C'est-à-dire la longueur qui sépare entre deux vecteurs.

Dans ce qui suit, nous allons passer en revue les principales mesures de distance utilisées en clustering. Elles sont illustrées dans le tableau suivant [45]:

Nom	**Paramètres**	**Fonction**
Distance de Manhattan	1-distance	$\sum_{i=1}^{n} \lvert x_i - y_i \rvert$
Distance euclidienne	2-distance	$\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$
Distance en Minkowski	p-distance	$\sqrt[p]{\sum_{i=1}^{n} \lvert x_i - y_i \rvert^p}$
Distance de Tchebychev	∞-distance	$\lim_{p\to\infty} \sqrt[p]{\sum_{i=1}^{n} \lvert x_i - y_i \rvert^p} = \sup_{1\leq=i\leq=} \lvert x_i - y_i \rvert$

Tableau 3.1 : Principales mesures de distance.

Selon les types ci-dessus, la plus connue et plus couramment utilisé est la distance euclidienne qu'il s'agit d'une distance géométrique dans un espace multidimensionnel, Ce qui nous amené de le choisir dans notre étude.

La distance euclidienne entre deux vecteurs $v(x_v, y_v)$ et $u(x_u, y_u)$ est définie par la formule suivante :

$$d(uv) = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2}$$

3.4.1.2 Méthodes de Clustering

Il existe plusieurs grandes catégories de clustering :

a) Les méthodes hiérarchiques

Ces méthodes partitionnent récursivement les instances de manière ascendante ou descendante pour but de construire les clusters [46], ils utilisent une matrice de distances entre les individus pour trouver le cluster le plus proche d'un autre.

On commence d'abord d'un ensemble de n individus en tant que singletons, alors les deux premiers individus les plus proches se font la première connexion.

Pour la deuxième étape, il faut d'abord mettre à jour la matrice des distances en supprimant une case à cause de regroupement de deux individus. Où la distance est calculée dans ce cas en choisissant la distance minimale entre individus, maximale ou bien moyenne.

Et après avoir terminé cette étape, on connecte les deux groupes les proches, et ainsi de suite pour les étapes suivantes, jusqu'à ou se fait la dernière connexion des deux derniers groupes qui recouvrent tous les individus [47].

Le résultat des méthodes de clustering hiérarchiques est un dendrogramme, représentant les niveaux de similarité auxquels les regroupements changent et le regroupement imbriqué d'objets. Et en coupant le dendrogramme au niveau de similarité souhaité on obtient le regroupement des objets de données [46].

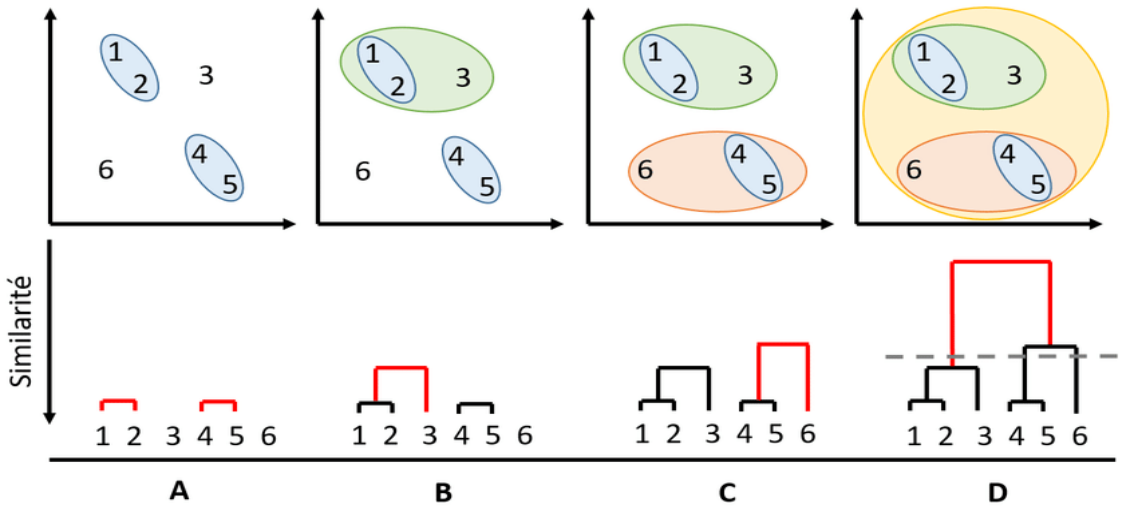

Figure 3.1 : Exemple de Classification Ascendante Hiérarchique (CAH).

b) Les méthodes à densité

Les méthodes à densité considèrent les classes qui correspondent aux zones denses dans l'espace d'objet. C'est-à-dire le nombre de ses voisins dépasse un certain seuil fixé à priori. Ils essayent alors d'identifier les classes en se basant sur la densité des objets dans une région. Les objets sont alors groupés non pas sur la base d'une distance mais sur la base de la densité de voisinage qui excède une certaine limite.

Parmi les algorithmes les plus connus dans cette catégorie, nous citons DBSCAN (Density Based Spatial Clustering of Application with Noise) et OPTICS [47].

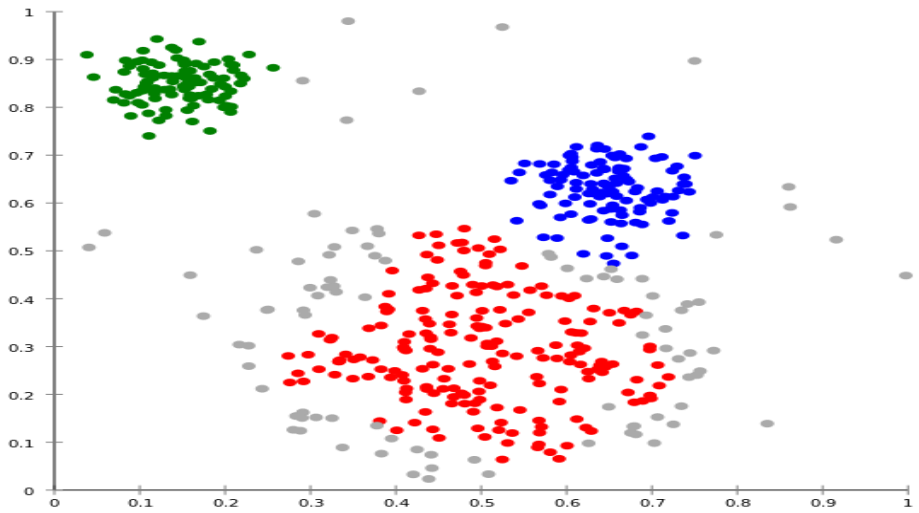

Figure 3.2: Exemple de clustering avec l'algorithme DBSCAN.

c) Les méthodes centroïdes

La méthode centroïde la plus classique et populaire dans la science de données est la méthode des K-moyennes, en anglais K-means (MacQeen 1967) est un algorithme populaire en Maching Learning qui permet d'obtenir un partitionnement de l'espace du jeu de données. Etant donné k, où k représente le nombre de groupes prédéfinis par l'analyste. Il classe les objets en plusieurs clusters, de sorte que les objets au sein d'un même cluster ont une similitude intra-classe élevée, tandis que les objets de différents clusters sont aussi différents

que possible (similitude inter-classe faible). Dans k-means, chaque cluster est représenté par son centre (centroïde) qui correspond à la moyenne des points attribués au cluster. Chaque cluster est représenté par son centre de gravité [47].

Notre approche s'appuie sur l'utilisation de l'algorithme de K-means car notre objectif est de regrouper les mots les plus similaires en des petits clusters pour représenter chaque groupe (cluster) par un seul mot qui est au centre de cluster.

Maintenant la question qui se pose, comment on va choisir le nombre de clusters K ?

Le choix de nombre de clusters K n'est pas forcément aléatoire, spécialement quand on a un jeu de données plus grand, et puisque nous ne faisons aucune hypothèse sur les données à l'avance, si nous choisissons un grand nombre de K, cela peut conduire à un partitionnement trop fragmenté des données (overfitting). En revanche, si on choisit un nombre de clusters trop petit, cela nous donnera des groupes très généraux contenant beaucoup de données (underfitting), et dans les deux cas, cela nous empêchera de détecter les features intéressants dans les données.

Donc la difficulté est de savoir comment choisir le nombre de clusters K, mais malheureusement il n'existe pas de procédé automatisé pour trouver le bon nombre de clusters.

La méthode qui nous utilisons pour choisir le nombre optimal de clusters K est la méthode Elbow (la méthode du coude). Dans cette méthode nous lançons k-means avec différentes valeurs de K pour calculer la variance de différents clusters. La variance est la somme des distances entre chaque centroide d'un cluster et les différents points inclus dans le même cluster.

Généralement quand nous plotons le graphe et nous mettons les différents nombres de clusters K en fonction de la variance, nous retrouvons un graphique similaire à celui-ci :

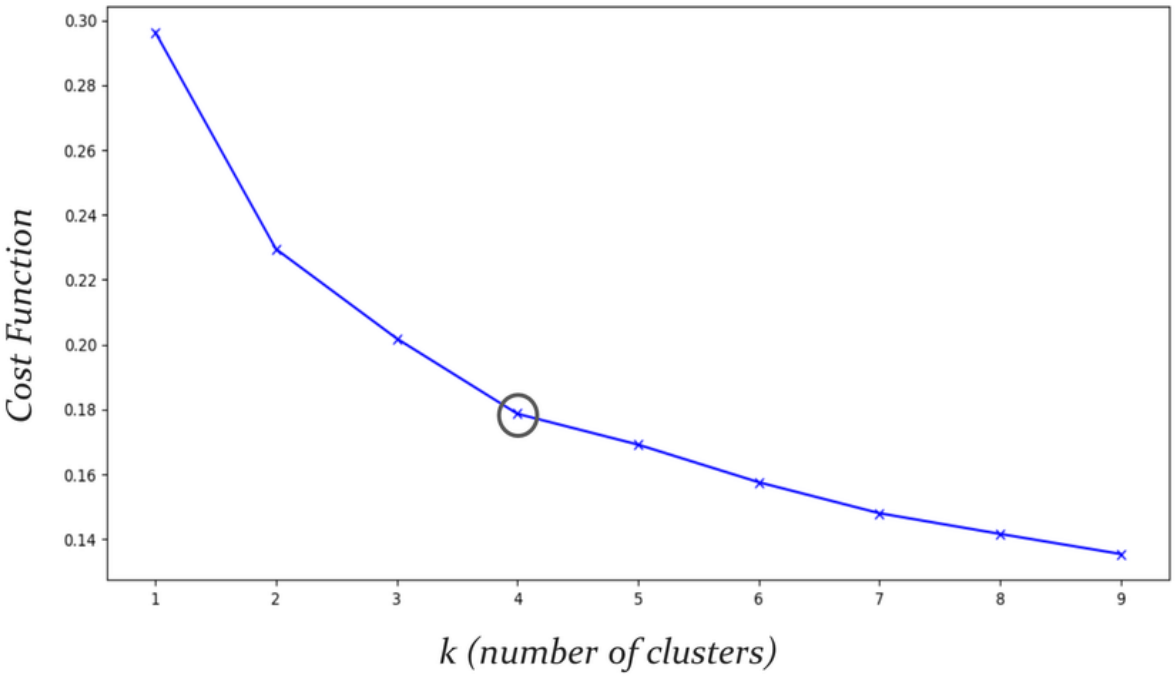

Figure 3.3 : Exemple d'un graphe de la méthode Elbow.

Nous remarquons à travers ce graphique la forme du bras, le point le plus haut représente l'épaule, et le point 9 représente la main, alors le nombre optimal de clusters K est le point qui représente le coude dans cet exemple K=4 c'est le nombre optimal de clusters.

Maintenant, pour choisir les centres des clusters, il existe plusieurs méthodes comme la méthode médïodes, mais ces méthodes ne répondent pas quand la taille de la base de teste est trop grande, alors nous choisissons un point virtuel au centre et à chaque fois nous essayons de minimiser l'erreur pour l'obtenir au centre de cluster. La figure 3.4 montre en exemple:

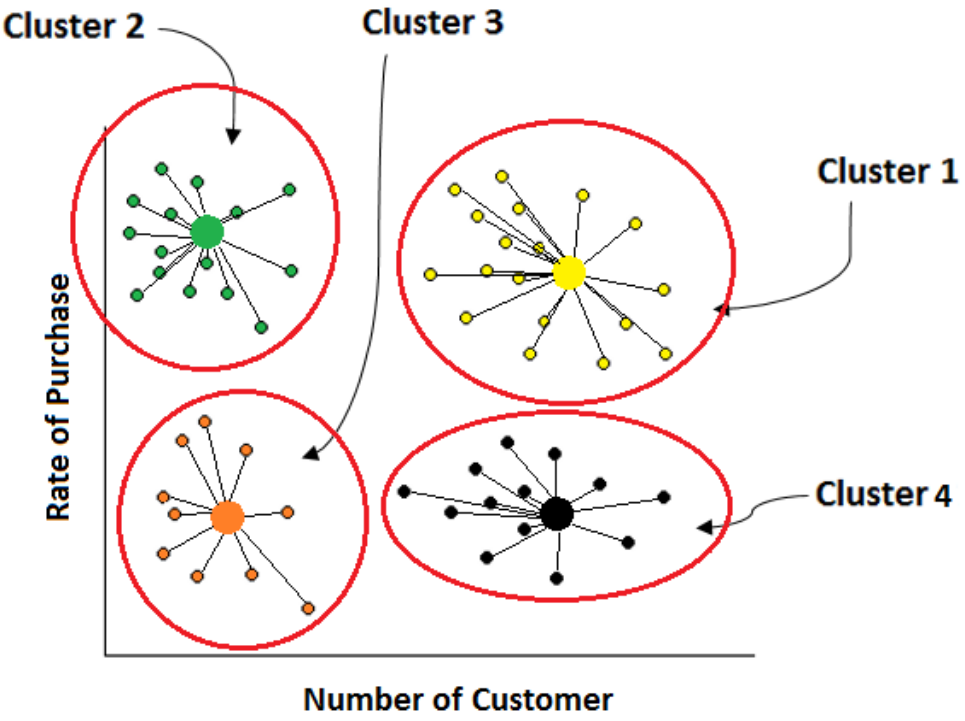

Figure 3.4: Exemple de k-means.

3.5 Approche Proposée

Notre travail proposé vise à étudier l'impact de la sélection de mot par plongement lexical (word embedding) en considérant la sémantique du mot, ou chaque mot est représenté par un vecteur de valeurs réels, où l'objectif est d'améliorer les performances des techniques de classification par la représentation des connaissances dans un espace plus réduit.

3.5.1 Structures de travail

Ce travail est organisé comme suit :

- Nous choisissons un dataset textuel et le traitons.
- Nous construisons notre modèle word2vec (CBOW).
- Nous faisons une classification sur le corpus par les méthodes de pondération par exemple : TF-IDF et les méthodes de sélection de caractéristiques les plus connues comme IG, CH2, MI, etc, avec les classifieurs : Random Rorest (RFC), k-Nearest-Neighbors (KNN) et Decision Tree (DT).
- Sur le même corpus nous choisissons les caractéristiques par clustering.

- Réduction du vocabulaire (supprimer les mots moins informatifs), et à chaque fois nous ferons un nouveau teste sur les nouveaux mots.
- Nous changeons le nombre de clusters et nous ferons une nouvelle classification pour trouver s'il y a une amélioration dans la précession ou pas, s'il y a une amélioration nous continuons jusqu'à nous trouvons les meilleurs points (bonne classification).
- Nous faisons une comparaison entre notre approche et les bases line existantes pour trouver que les meilleurs scores.

3.5.2 Architecture

Dans cette section, nous présentons l'architecture de notre approche que nous avons nommée SCPL *(Sélection de Caractéristiques par Plongement Lexical)*.

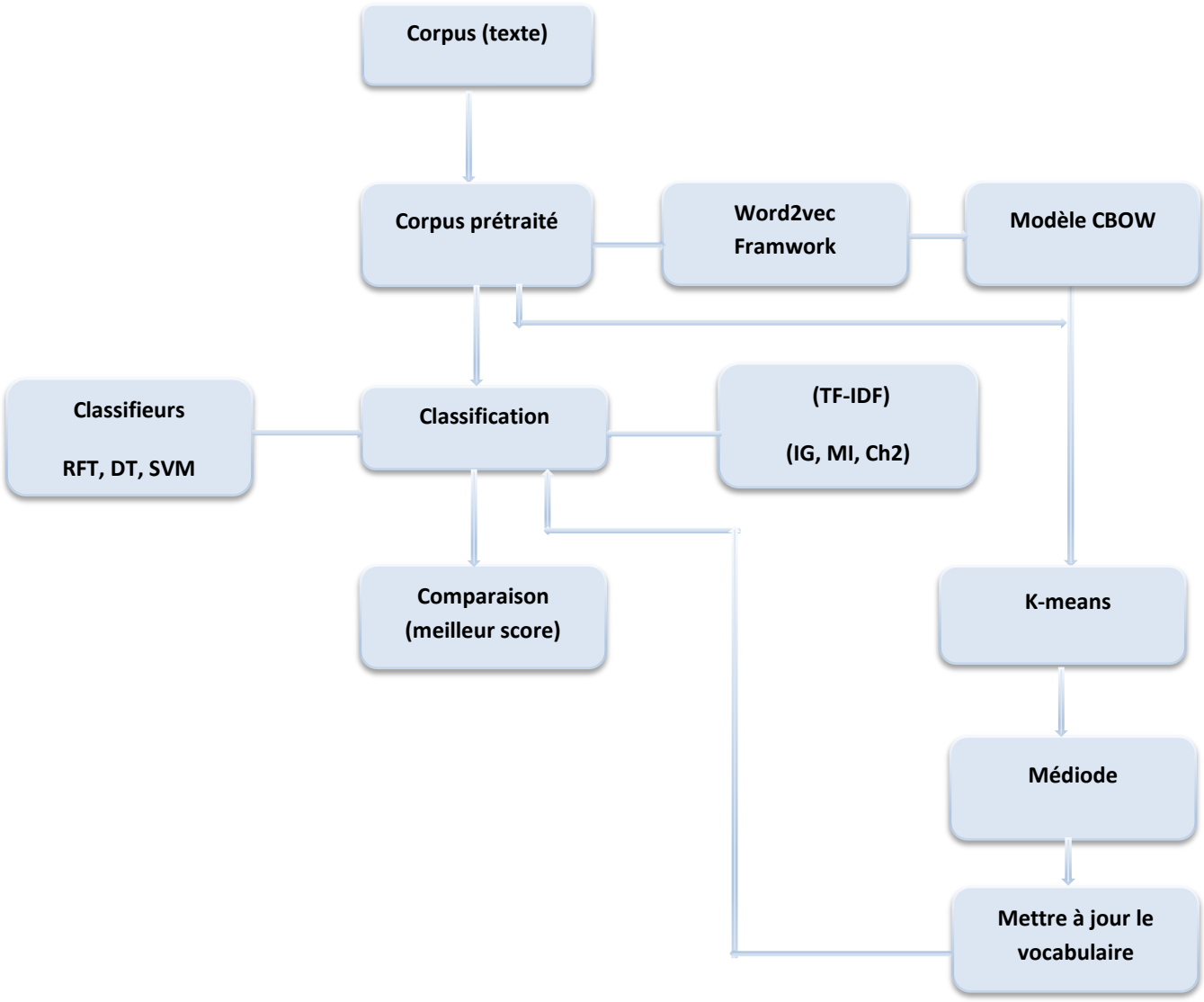

Figure 3.5 : Architecture de l'approche SCPL.

Le pseudo-algorithme suivant résume l'ensemble des étapes de notre approche :

Algorithme SCPL

Entrée : C : Corpus Textuel, CLF : Ensemble de classifieurs, **FS** : Méthodes de sélection de caractéristiques (TF-IDF, IG, MI, CH2)

Sortie : **best_features** : Ensemble de caractéristiques sélectionnées, **best_score** : Réel

Fonction prétraitement (C) :

Début

Pour tout document **d** dans **C faire**

Transfert majuscule en minuscule (d)

Supprimer ponctuation (d)

Supprimer mots vides (d)

Supprimer nombre (d)

Return Corpus Prétraité (CP)

Fin

Fonction construire modèle CBOW (CP : Corpus Prétraité)

V= {} # Ensemble de vecteurs générés

Début

Répéter :

Pour tout document **d** dans CP **faire**

Pour tout mot **m** de **d faire**

v ← générer vecteur (m)

V = V + v

Return V

Fin

Fonction classification ()

Scores = {}

Début

Pour tout classifier **cl** dans **CLF faire**

Pour tout **fs** dans **FS faire**

Score ← classification(cl, fs)

Scores = Scores + score

Return Scores

Fin

Fonction FS_par_Kmeans (seuils : ensemble d'entiers positifs)

Scores = {}

Début

Pour tout **s** dans **seuils faire**

clusters = Kmeans(k=s)

features ={}

Pour tout **cluster** dans **clusters faire**

m = medoide(cluster)

features = features + m

Pour tout **clf** in **CLF faire**

score = classification(clf, features)

Scores = Scores + score

best_score = Comparer (max (scores)) avec FS_scores

best_features = selectionner_features (best_score)

Return best_features, best_score

Fin.

Fin Algorithme.

3.6 Conclusion

Dans ce chapitre, nous avons expliqué les étapes détaillées de notre approche, où nous avons fait une combinaison entre le framework word2vec avec la technique de clusternig K-means, afin de nous permettre de résoudre le problème posé.

Après avoir réalisé la conception, qui aboutit à la présentation de notre architecture, il serait judicieux de penser à mettre en œuvre notre projet. En effet, cela nous amène au quatrième chapitre, ce dernier est consacré à la mise en œuvre de notre architecture.

Chapitre 4

Expérimentation et Evaluation des Résultats

4.1 Introduction

Après avoir terminé l'étape de formalisation de notre approche, nous abordons dans ce chapitre la phase d'implémentation qui consiste le dernier volet de ce rapport et qui a pour objectifs de mettre en œuvre notre approche. Nous allons commencer tout d'abord par une description des ressources matérielles et préciser l'environnement de développement du système. Nous présentons dans la suite, une description des étapes menés dans le processus d'expérimentation. Par la suite, nous discutons les résultats obtenus.

4.2 Description des ressources matérielles et logicielles

4.2.1 Ressources Matérielles

Processeur : intel (R), Celeron (R).

Capacité Mémoire : 4 MO.

Vitesse d'horloge : 2.16 GHz.

4.2.2 Ressources Logicielles

a) Enivrements de développement

Durant l'implémentation de notre projet nous avons utilisé Python version 3.8.5 comme langage de programmation, et Jupyter Notebook comme IDE, dans lequel nous pouvons écrire et exécuter notre code, où nous les définissons comme suit :

Python : est un langage de programmation open source, puissant et facile à apprendre. Il est interprété qui ne nécessite pas d'être compilé pour fonctionner, Python permet aux programmeurs de se focaliser sur ce qu'ils font plutôt que sur la façon dont ils le font. Ainsi, écrire des programmes prend moins de temps que dans un autre langage [48].

Jupyter : est un notebook de calcul open source, c'est une application web basée client permettant de créer et de partager du code, des équations, des visualisations ou du texte. Son nom est en fait une référence à trois langages : **Ju**lia, **Pyt**hon et **R**. Avec cet outil, il est possible de visualiser le code et l'exécuter depuis la même interface utilisateur. On peut mm apporter des changements au code et vérifier les résultats de ces modifications instantanément. Des données peuvent également être ajoutées. En outre, Jupyter Notebook s'exécute via un navigateur web [49].

c) Les bibliothèques nécessaires

Nous avons également adopté certaines bibliothèques nécessaires pour mettre en œuvre notre approche :

Gensim: est une bibliothèque libre de Python conçue pour extraire automatiquement des sujets sémantiques des documents, pour traiter les textes numériques bruts et non-structurés. Les algorithmes de gensim, tels que Word2vec, n'ont besoin que d'un corpus de documents en texte brut [50].

Scikit-learn : est une bibliothèque Python pour les algorithmes d'apprentissage automatique, elle se concentre sur l'apprentissage de la machine, dédié aux non-spécialistes en utilisant un langage général de haut niveau. L'accent est mis sur la facilité d'utilisation, la performance, la documentation et la cohérence de l'API. Il a des dépendances minimales et est attribué sous la licence BSD simplifiée, enchaînant son utilisation dans les milieux académiques et commerciaux [51].

Pandas : est une bibliothèque écrite pour le langage de programmation python permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles [52].

Numpy : est une bibliothèque pour le langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux. NumPy est la base de SciPy, regroupement de bibliothèques Python autour du calcul scientifique [53].

4.3 Démarche expérimental

Après avoir défini les grandes lignes de l'approche méthodologique que nous avons nommé **SCPL** *(Sélection de Caractéristiques par Plongement Lexical)* dans le chapitre 3, nous allons voir à présent plus en détails le processus d'application de cette dernière :

4.3.1 Jeux de données et prétraitement

a) Jeux de données

Nous avons utilisé différents jeux de données de taille et de complexité variables sous forme CSV pour évaluer notre approche, et la bibliothèque Pandas pour les manipuler. Dans cette section nous présentons les deux datasets utilisés :

- La base IMDB composé de 50.000 critiques de films pour le traitement du langage naturel et l'analyse des textes, de taille 129.19 Mo. Il s'agit d'un ensemble de données pour la classification des sentiments contenant 25.000 critiques d'avis positifs et 25.000 critiques d'avis négatifs.

	category	text
0	0	Once again Mr. Costner has dragged out a movie...
1	0	This is an example of why the majority of acti...
2	0	First of all I hate those moronic rappers, who...
3	0	Not even the Beatles could write songs everyon...
4	0	Brass pictures (movies is not a fitting word f...
5	0	A funny thing happened to me while watching "M...
6	0	This German horror film has to be one of the w...
7	0	Being a long-time fan of Japanese film, I expe...
8	0	"Tokyo Eyes" tells of a 17 year old Japanese g...
9	0	Wealthy horse ranchers in Buenos Aires have a ...

Figure 4.1: Le dataset IMDB

- La base BBC News, est composé de 2.225 articles de taille 4,8 Mo, chacun est classé dans l'une des 5 catégories suivantes : business, entrainement, politics, sport ou tech. L'ensemble de données est divisé en 1490 articles pour l'apprentissage et 735 pour le test. La figure suivante présente le dataset BBC après avoir choisi seulement les deux catégories : bussiness et tech.

	category	text
0	tech	tv future in the hands of viewers with home th...
1	business	worldcom boss left books alone former worldc...
11	business	virgin blue shares plummet 20% shares in austr...
12	business	crude oil prices back above $50 cold weather a...
15	business	s korean credit card firm rescued south korea ...
18	business	japanese banking battle at an end japan s sumi...
19	tech	games maker fights for survival one of britain...
20	tech	security warning over fbi virus the us feder...
21	tech	halo 2 heralds traffic explosion the growing p...
24	tech	mobile audio enters new dimension as mobile ph...

Figure 4.2: Le dataset BBC News

b) Prétraitement

Pour la première étape est de faire passer les données de leur format brut et lisible à un format avec lequel l'ordinateur peut travailler plus facilement les opérations de prétraitement qui nous affections sur les deux datasets sont :

1- Tokenization : tous les textes sont tokenisés pour extraire des mots uniques à l'aide de l'outil de traitement du langage naturel (NLTK) pour python.
2- Supprimer les ponctuations.
3- Transfert la majuscule en minuscule.
4- Supprimer les mots vides.

Les figures suivantes présentent les deux datasets après prétraitement.

	category	text
0	0	mr costner dragged movie far longer necessary ...
1	0	example majority action films generic boring r...
2	0	first hate moronic rappers couldnt act gun pre...
3	0	even beatles could write songs everyone liked ...
4	0	brass pictures movies fitting word really some...
5	0	funny thing happened watching mosquito one han...
6	0	german horror film one weirdest seen br br awa...
7	0	longtime fan japanese film expected ca nt real...
8	0	tokyo eyes tells year old japanese girl falls ...
9	0	wealthy horse ranchers buenos aires longstandi...

Figure 4.3: IMDB après prétraitement

	category	text
0	tech	tv future hands viewers home theatre systems p...
1	business	worldcom boss left books alone former worldcom...
11	business	virgin blue shares plummet shares australian b...
12	business	crude oil prices back cold weather across part...
15	business	korean credit card firm rescued south korea la...
18	business	japanese banking battle end japan sumitomo mit...
19	tech	games maker fights survival one britain larges...
20	tech	security warning fbi virus us federal bureau i...
21	tech	halo heralds traffic explosion growing popular...
24	tech	mobile audio enters new dimension mobile phone...

Figure 4.4 : BBC News après prétraitement

4.3.2 Création d'un modèle word2vec

Afin de générer un modèle word2vec nous avons collecté les données de deux datasets utilisés dans une seule, et nous avons utilisé la bibliothèque Gensim avec les paramètres suivants :

• Taille du vecteur = 200
• Taille de la fenêtre (contexte) = 5

• Nombre d'entrainement sur le dataset = 30

• Nombre de processeurs parallèles = 4

En plus, on ne prend pas en considération les mots apparaissent mois que trois fois dans notre modèle Word2vec, chaque mot est représenté par un vecteur de 200 dimensions de nombres réels. Le calcule de similarité entre deux mots est obtenu par le cosinus de leurs vecteurs.

Par la suite, nous avons obtenu un vocabulaire de taille 119138 mots.

4.3.3 Test du modèle

Après avoir construit le modèle, nous avons vérifié la capacité de notre modèle de capturer la similarité sémantique des mots. Les figures suivantes montrent les dix premiers mots sémantiquement similaires aux mots 'man', 'actor', 'film', 'nice', respectivement:

```
man : ['woman', 'guy', 'men', 'person', 'lady', 'mans', 'boy', 'chap',
'son', 'dude']

actor : ['actress', 'actors', 'role', 'comedian', 'performer', 'performa
nce', 'roles', 'cast', 'actresses', 'comedienne']

film : ['movie', 'films', 'movies', 'flick', 'documentary', 'story', 'b
r', 'picture', 'cinema', 'thriller']

nice : ['good', 'neat', 'great', 'interesting', 'cool', 'wonderful', 'li
ked', 'excellent', 'refreshing', 'fine']
```

Figure 4.5 : Les dix premiers mots similaires aux mots 'man', 'actor', 'film', 'nice' capturés par notre modèle.

4.3.4 Résultats de classification et comparaison

L'objectif des expérimentations effectuées sur les jeux de données est de tester l'effet du non sélection de caractéristiques et les méthodes de sélection de caractéristiques (MI, CH2) pour les comparer avec notre approche SCPL (*Sélection de Caractéristiques par Plongement Lexical*) en utilisant trois classifiers et consignant les résultats obtenus de précision de classification et taux d'erreur. Cela permettra de déduire lequel des trois type d'algorithmes

d'apprentissage utilisés offre des meilleures précisions de classification par rapport aux classifiers utilisés.

L'évaluation de la précision de classification est faite par une validation croisée pour les data-sets. Nous avons utilisé les exemples d'entraînement comme montré dans la figure suivante:

```
print (X_train)
```

```
1408    media gadgets get moving pocketsized devices l...
420     small firms hit rising costs rising fuel mater...
983     india seeks boost construction india cleared p...
351     new consoles promise big problems making games...
430     korea spending boost economy south korea boost...
                            ...
270     building giant asbestos payout australian buil...
672     microsoft plans safer id system microsoft plan...
2124    tate lyle boss bags top award tate lyle chief ...
1079    ore costs hit global steel firms shares steel ...
262     digital uk driven net tv uk adoption digital t...
Name: text, Length: 610, dtype: object
```

```
print(y_train)
```

```
1408         tech
420      business
983      business
351         tech
430      business
           ...
270      business
672         tech
2124     business
1079     business
262         tech
Name: category, Length: 610, dtype: object
```

Figure 4.6 : Exemples d'entrainement.

4.3.3.1 Classification sans sélection de caractéristiques

Les méthodes statistiques permettent d'identifier la fréquence de mot dans le texte, nous avons décrit le principe de ces derniers dans le deuxième chapitre. Ainsi que, la génération des fréquences est basée sur des formules mathématiques qui nécessite des données numériques, en s'aident de package scikit-learn de python.

Dans le cadre de notre travail, nous avons utilisé la pondération TF-IDF et nous précisons que les algorithmes de classification utilisés sont: Random Rorest (RFC), K Nearest Neighbor (KNN) et Decision Tree (DT) pour faire la classification sur les corpus, les résultats des précisions de classification illustrés dans le tableau suivant :

Classifiers	**Précision de classification avec TF-IDF**
Random Rorest (RFC)	97.67%
k-Nearest-Neighbors (k-NN)	96.01%
Decision Tree (DT)	87.71%

Tableau 4.1 : Résultats obtenus de classification par la pondération TF-IDF.

4.3.3.2 Classification avec sélection de caractéristiques

Parmi les algorithmes de sélection de caractéristiques dont nous avons parlé dans le premier chapitre, nous avons choisi Mutel Informtion (MI) et CH-Square (CH2) pour faire la classification avec les mêmes classifiers Random Rorest (RFC), k-Nearest Neighbors (k-NN) et Decision Tree (DT), les résultats de classification illustrés dans le tableau suivante :

Classifiers	**Précision de classification avec MI**	**Précision de classification avec CH2**
Random Rorest (RFC)	96.68%	97.67%
k-Nearest-Neighbor (k-NN)	92.36%	80.07%
Decision Tree (DT)	89.37%	88.70%

Tableau 4.2 : Résultats de classification obtenus après sélection de caractéristiques par MI et CH2.

Les meilleurs résultats obtenus de classifications avec et sans sélection de caractéristiques, nous les avons stockés dans un dictionnaire que nous avons nommée ‘best_scores’, pour les comparer ensuite avec les résultats obtenus par notre modèle.

4.3.4 Implémentation de SCPL

Nous avons implémenté notre approche SCPL, où nous avons fait un clustering sur les mots du vocabulaire représentés par leurs vecteurs générés par CBOW, et cela pour tester la performance de K-means dans la génération des groupes de mots portant une similarité sémantique. L’exemple suivant, illustré dans la figure 4.7, montre le résultat de clustering de 500 mots regroupés en quatre clusters :

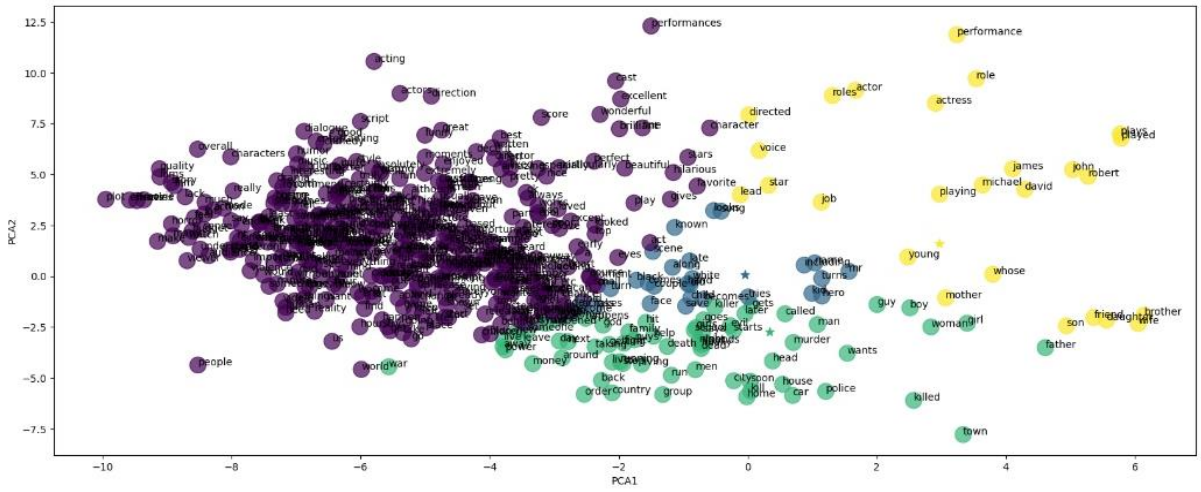

Figure 4.7 : Exemple de clustering de 500 mots issus du modèle CBOW construit.

Nous remarquons à travers ce graphique que les mots qui sont proche sémantiquement sont proches les uns des autres dans le graphique, par contre les mots qui sont loin sémantiquement sont loins les uns des autres dans le graphique. Par exemples, les mots ‘roles’, ‘actor’,’star’,’job’ sont des mots proches sémantiquement, donc on les retrouve dans le même cluster.

4.3.5 Sélection des caractéristiques par SCPL

La méthode SCPL (*Sélection de Caractéristiques par Plongement Lexical*) que nous avons proposée est basée sur la combinaison d’une méthode de plongement lexical word2vec et K-means comme technique de clustering. Pour tester la performance de SCPL, les précisions des classifiers obtenues après sélection des caractéristiques avec SCPL doivent être comparées à celles obtenues par les mêmes classifiers après sélections des caractéristiques avec les méthodes communes MI et CH2.

Pour cela, nous avons fait une classification sur le dataset avec les trois algorithmes de classification: Random Rorest (RFC), k-Nearest Neighbors (k-NN) et Decision Tree (DT), où nous avons fait des clusterings (k-means), on changeant la valeur de k (nombres de clusters) avec des grandes valeurs, et à chaque itération nous faisons un réduction sur le vocabulaire par la suppression des mots redondants et inutiles, et à chaque fois nous ferons un nouveau test sur les nouveaux mots, et pour chaque classification nous comparons les scores.

Pour le mot à choisir comme feature dans un cluster, il doit vérifier que la distance entre ce dernier et les autres mots du même cluster soit minimale, comme il peut aussi vérifier la distance minimale avec le centre du cluster, c'est à dire on choisit le mot qui est le plus proche au centre du cluster comme feature (caractéristique).

Les résultats des expérimentations sont exposés dans les figures suivantes :

```
classifier: RFC
cluster #: 4
#selected features:  17985 / 17997
0.9867
cluster #: 9
#selected features:  17965 / 17997
0.9734
cluster #: 14
#selected features:  17937 / 17997
0.9801
cluster #: 19
#selected features:  17910 / 17997
0.9767
cluster #: 24
#selected features:  17862 / 17997
0.9734
cluster #: 29
#selected features:  17794 / 17997
0.9801
cluster #: 34
#selected features:  17742 / 17997
0.9834
cluster #: 39
#selected features:  17696 / 17997
0.9801
cluster #: 44
#selected features:  17680 / 17997
0.9934
cluster #: 49
#selected features:  17602 / 17997
0.9734
cluster #: 54
#selected features:  17535 / 17997
0.9734
cluster #: 59
#selected features:  17434 / 17997
0.9801
cluster #: 64
#selected features:  17406 / 17997
0.9834
cluster #: 69
#selected features:  17386 / 17997
0.9834
cluster #: 74
#selected features:  17369 / 17997
0.9867
cluster #: 79
#selected features:  17338 / 17997
0.9701
cluster #: 84
#selected features:  17296 / 17997
0.9734
cluster #: 89
#selected features:  17199 / 17997
0.9867
cluster #: 94
#selected features:  17180 / 17997
0.9867
cluster #: 99
#selected features:  17152 / 17997
0.9834
cluster #: 104
#selected features:  17119 / 17997
0.9867
cluster #: 109
#selected features:  17087 / 17997
0.9867
cluster #: 114
#selected features:  17063 / 17997
0.9900
cluster #: 119
#selected features:  16899 / 17997
0.9767
cluster #: 124
#selected features:  16720 / 17997
0.9767
cluster #: 129
#selected features:  16699 / 17997
0.9801
cluster #: 134
#selected features:  16625 / 17997
0.9801
cluster #: 139
#selected features:  16593 / 17997
0.9767
cluster #: 144
#selected features:  16569 / 17997
0.9801
cluster #: 149
#selected features:  16407 / 17997
0.9701
cluster #: 154
#selected features:  16389 / 17997
0.9767
```

Figure 4.8: Résultats classification de RFC avec sélection de caractéristiques (mots) par SCPL.

```
classifier: KNN
cluster #: 4
#selected features:  17985 / 17997
0.9601
cluster #: 9
#selected features:  17965 / 17997
0.9601
cluster #: 14
#selected features:  17937 / 17997
0.9601
cluster #: 19
#selected features:  17910 / 17997
0.9635
cluster #: 24
#selected features:  17862 / 17997
0.9635
cluster #: 29
#selected features:  17794 / 17997
0.9635
cluster #: 34
#selected features:  17742 / 17997
0.9601
cluster #: 39
#selected features:  17696 / 17997
0.9601
cluster #: 44
#selected features:  17680 / 17997
0.9601
cluster #: 49
#selected features:  17602 / 17997
0.9601
cluster #: 54
#selected features:  17535 / 17997
0.9601
cluster #: 59
#selected features:  17434 / 17997
0.9601
cluster #: 64
#selected features:  17406 / 17997
0.9601
cluster #: 69
#selected features:  17386 / 17997
0.9601
cluster #: 74
#selected features:  17369 / 17997
0.9601
cluster #: 79
#selected features:  17338 / 17997
0.9601
cluster #: 84
#selected features:  17296 / 17997
0.9601
cluster #: 89
#selected features:  17199 / 17997
0.9568
cluster #: 94
#selected features:  17180 / 17997
0.9568
cluster #: 99
#selected features:  17152 / 17997
0.9568
cluster #: 104
#selected features:  17119 / 17997
0.9535
cluster #: 109
#selected features:  17087 / 17997
0.9535
cluster #: 114
#selected features:  17063 / 17997
0.9535
cluster #: 119
#selected features:  16899 / 17997
0.9568
cluster #: 124
#selected features:  16720 / 17997
0.9535
cluster #: 129
#selected features:  16699 / 17997
0.9535
cluster #: 134
#selected features:  16625 / 17997
0.9601
cluster #: 139
#selected features:  16593 / 17997
0.9601
cluster #: 144
#selected features:  16569 / 17997
0.9635
cluster #: 149
#selected features:  16407 / 17997
0.9635
cluster #: 154
#selected features:  16389 / 17997
0.9635
```

Figure 4.9 : Résultats classification de k-NN avec sélection de caractéristiques (mots) par SCPL.

```
classifier: DT
cluster #: 4
#selected features:  17985 / 17997
0.8837
cluster #: 9
#selected features:  17965 / 17997
0.8804
cluster #: 14
#selected features:  17937 / 17997
0.8605
cluster #: 19
#selected features:  17910 / 17997
0.8837
cluster #: 24
#selected features:  17862 / 17997
0.8837
cluster #: 29
#selected features:  17794 / 17997
0.8870
cluster #: 34
#selected features:  17742 / 17997
0.8837
cluster #: 39
#selected features:  17696 / 17997
0.8804
cluster #: 44
#selected features:  17680 / 17997
0.8671
cluster #: 49
#selected features:  17602 / 17997
0.8704
cluster #: 54
#selected features:  17535 / 17997
0.8804
cluster #: 59
#selected features:  17434 / 17997
0.8870
cluster #: 64
#selected features:  17406 / 17997
0.8870
cluster #: 69
#selected features:  17386 / 17997
0.8837
cluster #: 74
#selected features:  17369 / 17997
0.8837
cluster #: 79
#selected features:  17338 / 17997
0.8804
cluster #: 84
#selected features:  17296 / 17997
0.8837
cluster #: 89
#selected features:  17199 / 17997
0.8605
cluster #: 94
#selected features:  17180 / 17997
0.8704
cluster #: 99
#selected features:  17152 / 17997
0.8837
cluster #: 104
#selected features:  17119 / 17997
0.8771
cluster #: 109
#selected features:  17087 / 17997
0.8804
cluster #: 114
#selected features:  17063 / 17997
0.8771
cluster #: 119
#selected features:  16899 / 17997
0.9169
cluster #: 124
#selected features:  16720 / 17997
0.9103
cluster #: 129
#selected features:  16699 / 17997
0.9136
cluster #: 134
#selected features:  16625 / 17997
0.9136
cluster #: 139
#selected features:  16593 / 17997
0.9070
cluster #: 144
#selected features:  16569 / 17997
0.9070
cluster #: 149
#selected features:  16407 / 17997
0.9103
cluster #: 154
#selected features:  16389 / 17997
```

Figure 4.10: Résultats classification de DT avec sélection de caractéristiques (mots) par SCPL.

Une fois ces calculs terminés nous avons pu implémenter les résultats obtenus dans des graphiques.

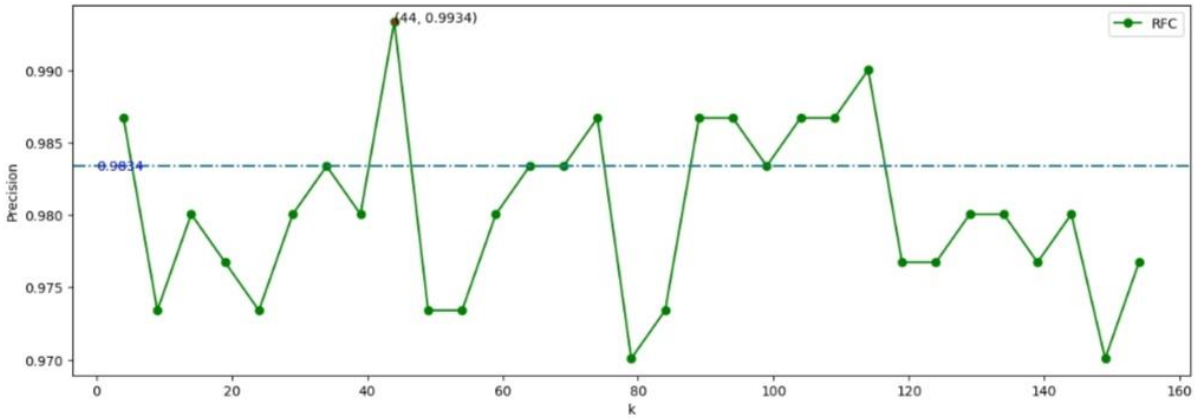

Figure 4.11 : Représentation graphique des précisions obtenues par RFC.

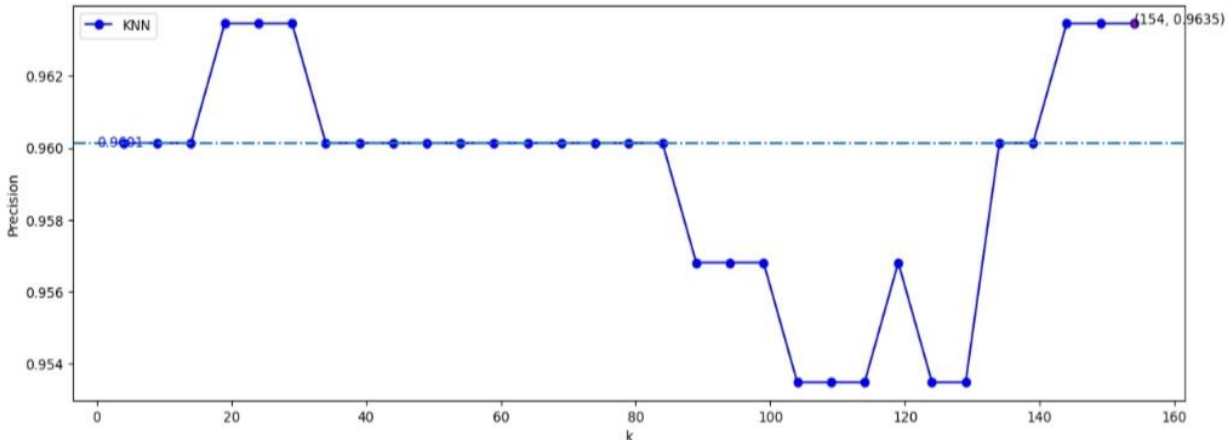

Figure 4.12 : Représentation graphique des précisions obtenues par k-NN.

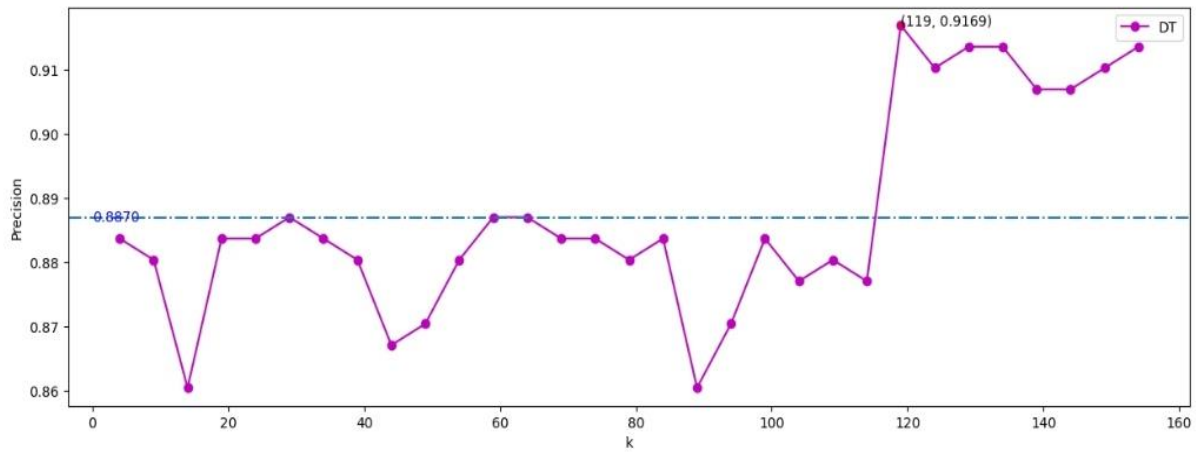

Figure 4.13 : Représentation graphique des précisions obtenues par DT.

Le tableau récapitulatif suivant, résume les résultats obtenus par notre approche SCPL :

Classifier	Nombre de Clusters (Nombre de caractéristiques)	Meilleure Précision
RFC	44	99,34%
k-NN	154	96,35%
DT)	119	91,69%

Tableau 4.3 : les résultats classification obtenus par notre approche SCPL.

Les expérimentations ont montré que notre méthode est plus performante en termes de précisions achevées par les trois classifiers, avec un nombre optimal de caractéristiques (mots) comparé avec MI et CH2.

4.4 Conclusion

Dans le dernier chapitre de notre projet, nous avons implémenté notre approche qui répond parfaitement aux objectifs fixés au début.

Notre méthode de sélection de caractéristiques SCPL est basée sur la combinaison de deux techniques qui sont : le framework word2vec pour capturer la sémantique des mots et le clustering avec k-means pour extraire les mots les plus représentatifs qui conduisent à une meilleure performance de classification. Nous avons fait une évaluation approfondie de la nouvelle méthode par comparaison des résultats obtenus avec ceux obtenus par les méthodes les plus couramment utilisés dans la littérature (MI et CH2). Cette évaluation confirme la performance de notre approche en considérant que la sélection des mots en se basant sur la sémantique conduit à une classification meilleure en terme de précision que la sélection basée sur la fréquence des mots.

Conclusion Générale et Perspectives

La classification des textes est un sujet fastidieux mais il a beaucoup de bénéfice pour l'avancement des technologies de l'information, pour cela nous avons axé notre recherche sur la sélection de caractéristiques par plongement lexical pour tester l'impact de ce dernier sur la précision de classification des textes.

Notre thème est très vaste et liée à plusieurs domaines de recherche. Pour cela, nous avons commencé notre recherche par un état de l'art sur la sélection de caractéristiques et les algorithmes existants dans la littérature, et les travaux connexes effectués par les chercheurs. Après, nous avons parlé des différentes méthodes de plongement lexical en citant les caractéristiques de chacune. Ensuite, nous avons proposé notre approche qui est une combinaison du framework word2vec avec la méthode de clustering K-means, en résultant une méthode de sélection de caractéristiques par plongement lexical qui nous avons nommée SCPL.

Nous avons évalué notre système avec deux datasets, et nous avons obtenu des résultats meilleurs par rapport aux algorithmes existants dans la littérature. SCPL peut être appliquée sur divers, et ceci exige bien sûr une adaptation selon la langue en question comme la disponibilité des ressources (datasets), et le prétraitement nécessaire pour la langue choisie.

Comme perspectives de recherche future, nous envisageons de compléter le cadre formel proposé par une étude plus approfondie en profitant des technologies boostées par l'intelligence notamment en Deep-Learning.

Les connaissances acquises au cours de ce projet de fin d'études couvrent de nombreuses dimensions parmi lesquelles nous avons pu maîtriser le domaine de classification de textes et son importance. Sur le plan pratique, c'est l'approfondissement de nos connaissances dans le langage Python et la maîtrise de certaines bibliothèques qu'il contient.

Méthodologiquement, nous avons mené la démarche de la recherche et mis en œuvre toutes ses étapes, qui consistent à comprendre le domaine, définir le problème, lui proposer une solution et la mettre en œuvre.

Références Bibiographiques

[1] Dash, M. et Liu, H. (1997). **« Feature selection for classification »,** Intelligent Data Analysis**,** Volume 1(1-4), pp-pp 131–156.

[2] Pudil, P., Novovi covà, J., et Kittle.,J,.1994. **« Floating search methods in feature selection »**, Pattern Recognition Letters, Volume. 15, pp-pp 11198-1125.

[3] Y. Yuan, H. Liu, G. Qiu, Décembre 20, 2013 **« A new approach for HIV-1 protease cleavage site prediction combined with feature selection»**, Journal of Biomedical Science and Engineering**,** Volume 06**,** pp-pp 1155- 1160.

[4] Abdelkader Abdelmalek et Walid Hebbar, 2013/ 2014, **«Evaluation des méthodes de Sélection de Variables en Apprentissage supervisé »,** Thèse Pour obtenir le diplôme de Master,.

[5] Bilal Belainine, avril 2017, «**classification super visée de texte courts et bruités** », Thèse Pour obtenir le diplôme de mastère.

[6] M. Zaiz Faouzi, 15 juillet 2010, « **Les Supports Vecteurs Machines (SVM) pour la reconnaissance des caractères manuscrits arabes**», Thèse Pour obtenir le diplôme de magister, Université Mohamed Khider – BISKRA.

[7] Bilal Belainine , Avril 2017, «**Classification super visée de texte courts et bruités** », Mémoire

[8] Philippe Lucidarme, Novembre 2003, « **Apprentissage et adaptation pour des ensembles de robots réactifs coopérants** », Thèse Pour obtenir le grade de docteur de l'université Montpellier II.

[9] Khodja Fouad, Octobre 2011, «**Conception d'un système intellegent à base de réseaux de neurones artificiels pour l'étude de la dynamique des streamers à la surface des polyméres** », Thèse Pour obtenir le diplôme de magister.

[10] Bernard, G., 1996. « **Application de réseaux de neurones artificiels à la reconnaissance automatique de caractères manuscrits** ». Thèse de doctorat en sciences appliquées, faculté polytechnique de Mons.

[11] Alain Girard, Avril 2007, **« Exploration d'un algorithme génétique et d'un arbre de décision à des fins de catégorisation** », Université du Québec.

[12] M. Belgiu, L. Dragut, 2016, « **Random forest in remote sensing: A review of applications and future directions**», ISPRS Journal of Photogrammetry and Remote Sensing. Volume 114, pp-pp 24-31.

[13] Article, Kai-Yan Feng, Yu-Dong Cai, Kuo-Chen Chou, le 27 juin 2005, **« Boosting classifier for predicting protien domain structural class »**, Biochemical and Biophysical Research Communications disponible en ligne, Volume 334-1, pp-pp 213- 217.

[14] Sadegh Bafandeh Imandoust et Mohammad Bolandraftar, Septembre-Octobre 2013,« **Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background** », Payame Noor University, Tehran, Iran.

[15] Bilal Belainine , avril 2017 **«classification super visée de texte courts et bruités »**.

[16] X. Deng, Y. Li, J. Weng , J. Zhang, 2018, **«Feature selection for text classification: A review»**, Multimedia Tools and Applications, Volume 78- 3, pp-pp 3797- 3816.

[17] J. Li, K. Cheng, S. Wang, F.Morstatter, R.P. Trevino, J. Tang, H. Liu, 2018, **«Feature Selection: A Data Perspective »,** ACM Computing Surveys, Volume 50- 6 **,**pp-pp 1-45.

[18] H. Liu, S. Wang, J. Tang, janvier 2016 **« Feature selection »** Encyclopedia of Machine Learning and Data Mining (pp.1-9).

[19] J. Yang, Y. Liu, X. Zhu, et al, 2012, « **A new feature selection based on comprehensive measurement both in inter-category and intra-category for text categorization** », Information Processing & Management, Volume 48-4 , pp-pp 741-754.

[20] G. Kou , P. Yang , Y. Peng , F. Xiao , Y. Chen , F.E. Alsaadi, 2020, « **Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods** », Applied Soft Computing, Volume 86, pp-pp 105836.

[21] M.H. Aghdam, N. Ghasem-Aghaee, M.E. Basiri, 2009 « **Text feature selection using ant colony optimization** », Expert Syst. Appl. Volume 36-3, pp-pp 6843–6853.

[22] Y. Yang, J.O. Pedersen, 1997, « **A comparative study on feature selection in text categorization, in: Fourteenth International Conference on Machine Learning** », Morgan Kaufmann Publishers Inc, pp-pp 412–420.

[23] M. Rahmaninia, P. Moradi, 2017, « **OSFSMI: online stream feature selection method based on mutual information** ». Appl Soft Comput, Volume 68, pp-pp 733-746.

[24] J. Che, Y. Yang, L. Li, X. Bai, S. Zhang, C. Deng, 2017, « **Maximum relevance minimum common redundancy feature selection for nonlinear data**». Information Sciences, Volume 409, pp-pp 68- 86.

[25] H. Sanz, C. Valim, E. Vegas, JM. Oller, F. Reverter, 2018 « **SVM-RFE: selection and visualization of the most relevant features through non-linear kernels** ». BMC bioinformatics, Volume 19-1, pp-pp 432.

[26] F .Viegas, L . Rocha, M. Gonçalves, F. Mourão, G. Sá, T. Salles, G. Andrade, I. A Sandin, 2017 « **genetic programming approach for feature selection in highly dimensional skewed data** ». Neurocomputing, Volume 273, pp-pp 544-569.

[27] J. Izetta, PF . Verdes, PM . Granitto, 2017. « **Improved multiclass feature selection via list combination** », Expert Systems with Applications, Volume 88, pp-pp 205–216.

[28] J. Xiao, H. Cao, X . Jiang, X. Gu, L . Xie. 2017 « **GMDH-based semi-supervised feature selection for customer classification** », Knowledge-Based Systems, Volume 132, pp-pp 236- 248.

[29] J. Liu, Y. Lin, M. Lin, S . Wu, J . Zhang. 2017 « **Feature selection based on quality of information** ». Neurocomputing.; Volume 225, pp-pp 11–22.

[31] Ricco Rakotomalala , **« Introduction au text mining principe et application»**, Université Lyon 2.

[33] P. Huilgol, 2020 **« Quick Introduction to Bag-of-Words (BoW) and TF-IDF for Creating Features from Text»,** Analytics Vidhya**.**

[35] Tomas Mikolov Tomas, sutskever, Ilya, Chen, Kai, Corrado, Greg S. et Dean, Jeff, 2013, « **Efficient Estimation of word Representation in vector Space** », Arxiv.

[36] Atul Dhingra ODH , 2017 « **word2vec** ».

[37] Lei Zhu, Guijun Wang et Xianchun Zou, **« Improved Information Gain Feature Selection Method For Chinese Text Classification Based On Word Embedding »,** School of Computer and Information Science, Southwest University.

[38] Sahar Ghannay, 2017, **«Etude sur les représentations continues de mots appliquées à la détection automatique des erreurs de reconnaissance de la parole »**, Université du Maine**.**

[39] Bnetekkouka Abderrahman, Chiek Abdellah, 9 janvier 2020 « **généralisation d'approches basées word embedding pour un système d'évaluation des réponses courtes adapté à la langue anglaise** », Thèse Pour obtenir le diplôme de master, université Saad Dahlab Belida.

[40] Philippe Dessus, Septembre 1999 « **Vérification sémantique de liens hypertextes avec LSA. 5ème Conférence** », internationale Hypertextes, hypermédias et internet (H2PTM'99), Paris, France. pp.119-129.

[41] Hicham El Boukkouri, « **Ré-entraîner ou entraîner soi-même ? Stratégies de pré-entraînement de BERT en domaine médical** », Université Paris-Saclay, CNRS, LIMSI, 91400, Orsay, France

[42] Bétrice Mazoyer, Nicoles hervé, céline hudelot, julia cagé , « **représentation lexicales pour la détection non supervisée d'événements dans un flux de tweets : étude des corpus français et anglais** », centrale supélec (université paris-scalay), MICS, Gif-sur-Yvette, France.

[44] J.A.Hartigan, 1975 « **Clustering Algorithms** » John Wiley and Sons Inc. New York.

[46] Ahmed Yahia , 2018 /2019 « **Clustering des données de puces à ADN** », mémoire de master.

Webographie

[38] J. Brownlee, 2019 **« A Gentle Introduction to the Bag-of-Words Model »,** Machine Learning Mastery, https://machinelearningmastery.com/gentle-introduction-bag-words-model/ consulté le 18/04/2021.

[40] **« Tf-idf :: A Single-Page Tutorial - Information Retrieval and Text Mining »** http://www.tfidf.com/ consulté le 18/05/2021.

[42] **« A Beginner's Guide to Bag of Words & TF-IDF »**, Pathmind, https://wiki.pathmind.com/bagofwords-tf-idf, consulté le 18/05/2021.

[43] « **What are the main differences between the word embeddings of ELMo, BERT, Word2vec, and GloVe? – Quora** », https://www.quora.com/What-are-the-main-differences-between-the-word-embeddings-of-ELMo-BERT-Word2vec-and-GloVe , consulté le 26/04/2021

[45] 2021, **« 9 Mesures de distance en science des données** », ICHI.PRO, https://ichi.pro/fr/9-mesures-de-distance-en-science-des-donnees-159983401462266, consulté le 22/05/2021.

[47] L. Eluère, 2021 « **Qu'est-ce que le clustering ? Les 3 méthodes à connaitre** », La revue IA, https://larevueia.fr/clustering-les-3-methodes-a-connaitre/, consulter 21/08/2021.

[48] « **Python.org** », https://www.python.org/ consulté le 22/07/2021.

[49] « **Project Jupyter** », https://jupyter.org/ consulté le 22/07/2021.

[50] « **GenSim** », https://gensim.org/home consulté le 23/07/2021.

[51] « **scikit-learn: machine learning in Python — scikit-learn 0.16.1 documentation** », http://scikit-learn.org/ consulté le 23/07/2021.

[52] « **pandas - Python Data Analysis Library** », https://pandas.pydata.org/ consulté le 24/07/2021.

[53] « **NumPy** », https://numpy.org/ consulté le 24/07/2021.