color(75bp)=(pgftransparent!100); color(100bp)=(pgftransparent!100)

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : SIOD 5 /M2/2018

Mémoire

présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Système d'Information et Optimisation Décisionnel (SIOD)**

Analyse des textes en utilisant Le Deep Learning

Par :

Belkhiri Khadidja

Soutenu le 25 juin 2018, devant le jury composé de :

Bouguetitiche Amina	M.A.A	Président
Meadi Med Nadjib	M.C.B	Rapporteur
Bendahmane Asma	M.A.A	Examinateur

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GÉNÉRALE

Introduction générale

Aujourd'hui avec le développement d'internet, la quantité d'information électronique est devient énormes. Ces données sont présentes dans des livres, des articles, des blogs et des réseaux sociaux, stocker dans des ordinateurs et échangées sur des réseaux de télécommunication informatiques comme internet. Il est donc très intéressant de trouver un moyen de qui nous aide à extraire et analyser les informations depuis ces grandes bases et trouver des applications efficaces pour conserver, chercher et classer ces informations afin d'assister les utilisateurs à trouver leurs besoins et faciliter leur travail dans certaines tâches qui sont devenues impossible à traiter manuellement[6, 1].

Objectifs :

Il existe beaucoup des techniques d'analyse des textes parmi lesquelles nous avons choisi la classification des textes. L'objectif de la classification des textes est de rassembler les textes similaires selon un certain critère au sein d'une même classe.

Notre but dans ce travail est de classifier des textes en utilisant une approche d'apprentissage profond et plus précisément les réseaux de neurones à convolution .

Organisation du mémoire :

Ce mémoire a été organisé en quatre chapitres :

- Dans le premier chapitre, nous allons présenter les notions de base du réseau de neurone et ces différentes architectures et nous essayons de définir le domaine d'apprentissage profond et spécialement les réseaux de neurones à convolution.
- Le deuxième chapitre est consacré à la description du domaine d'analyse de texte et ces techniques ainsi que les travaux connexes .
- Dans le troisième chapitre, nous allons exposé les architectures globales et détaillées de notre conception du système.
- Dans le dernier chapitre, nous avons montrer la partie expérimentale de notre travail et nous discutons les différentes résultats obtenues et nous terminons avec une conclusion générale.

CHAPITRE

1

APPRENTISSAGE PROFOND

1.1 Apprentissage Automatique

L'apprentissage automatique vise à construire automatiquement des connaissances à partir de grandes quantités d'information, il consiste à élaborer des algorithmes capables d'apprendre à partir des exemples, il ne s'agit pas ici d'apprendre des exemples, mais plutôt de bien généraliser à partir de ces exemples, afin d'effectuer une certaine tâche.
En général, l'apprentissage automatique consiste à apprendre de faire mieux dans le futur en se basant sur ce qui a été vécu dans le passé. Le but est de concevoir des algorithmes capables, à partir d'un nombre important d'exemples, d'assimiler la nature afin de pouvoir appliquer ce qu'ils ont ainsi appris aux cas futurs sans intervention ou assistance humaine.
L'apprentissage automatique peut être appliqué à une grande variété de tâches et de données. Selon le cas, différents cadres d'apprentissage peuvent s'appliquer[55].

1.1.1 Types d'apprentissage Automatique

Dans l'apprentissage automatique, on trouve beaucoup types d'apprentissage, on situe les deux types les plus utilisée : l'apprentissage supervisé et l'apprentissage non supervisé.

1.1.1.1 Apprentissage supervisé

Dans l'apprentissage supervisé, un modèle reliant des données d'apprentissage à un ensemble des valeurs de sortie (la valeur désirée).
On dispose d'un ensemble d'apprentissage qui constitue l'information disponible, chaque exemple de l'ensemble d'apprentissage est associée à une classe. Nous supposons que cette classe est donnée par ce que nous appelons un superviseur. L'apprentissage supervisé permit de trouver un algorithme qui classe correctement les exemples, c'est-à-dire qui leur donne la même classe que le superviseur.
Quelques algorithmes d'apprentissage supervisé
La plupart des algorithmes d'apprentissage supervisés tentent de trouver un modèle qui explique le lien entre des données d'entrée et les classes de sortie. Il existe de nombreuses méthodes d'apprentissage supervisé comme[7] :
-La méthode des k plus proches voisins.
-Les machines à vecteurs de support (SVM) .
-Les réseau de neurones.
-Les Arbres de décision.

-Les classifications naïve bayésienne.

1.1.1.2 Apprentissage non-supervisé

Apprentissage non-supervisé est connu sous nom de « Clustering », est un regroupement de données d'apprentissage en classes homogènes, sans aucun étiquette n'y soit rattaché (il n'y a pas de notion de sortie désirée).
Son objectif, très général, consiste à séparer un ensemble d'objets en différents groupes en fonction d'une certaine notion de similarité, les objets qui sont considérés comme similaires sont ainsi associés au même groupe alors que ceux qui sont considérés comme différents sont associés à des groupes distincts.
Il consiste à représenter un nuage des points d'un espace quelconque en un ensemble de groupes appelé Cluster.
Le regroupement est une problématique de recherche étudiée depuis de nombreuses années dans différentes communautés : apprentissage automatique, fouille de donnée, la reconnaissance des formes, statistiques, ... etc[30].
Quelques algorithmes d'apprentissage non supervisé
On peut voir aussi les méthodes non-supervisées les plus utilisées[23] :
- les centres mobiles (k-means)
- Fuzzy C-means
- Suffix Tree Clustering
- Hierarchical clustering
- Mixture of Gaussians (Expectation maximisation)

1.2 Les réseaux de neurones artificiels

Les réseaux de neurones artificiels sont des systèmes de traitement de l'information, un réseau de neurones est un ensemble de cellules inter-connectées, peut être considéré comme un modèle mathématique de traitement réparti, composé de plusieurs éléments (neurones), opérant en parallèle et connectés entre eux par des poids et le comportement du réseau dépend essentiellement des valeurs de ces poids . L'apprentissage est l'opération par laquelle les poids synaptique sont calculés de telle manière que le système réalise bien la fonction qu'on attend de lui [36, 3].

1.2.1 Historique des réseaux de neurones

•En 1943 , W. Mcculloch et W. Pitts ont proposé des neurones formel mimant les neurones biologique, capables de mémoriser les fonctions booléennes simples.
•En 1949, D. Hebb a mis en évidence l'importance du couplage synaptique dans l'apprentissage par renforcement ou la dégénérescence des liaisons interneuronales lors de l'interaction du cerveau avec le milieu extérieur[2].
•En 1958, F. Rosenblatt a développé le modèle du Perceptron. C'est un réseau de neurones inspiré du système visuel.
•S'inspirant du perceptron, Widrow et Hoff, sont développé dans la même période, le modèle de l'Adaline (Adaptive Linear Element).
•En 1969, M. Minsky et S. Papert ont publié leur livre « Perceptrons » et démontré les limites théoriques du perceptron.
•En 1982 ,les travaux de Hopfield ont montré que les réseaux de neurones artificiels étaient capables de résoudre des problèmes d'optimisation
• En 1984 c'est la découverte des cartes de Kohonen avec un algorithme non supervisé basé sur l'auto-organisation et suivi une année plus tard par la machine de Boltzman (1985).
• En 1986, Werbos a proposé pour la première fois le Perceptron Multi-Couche introduit par Rumelhart, et, simultanément, sous une appellation voisine, chez Le Cun (1985). Ces systèmes reposent sur la rétropropagation du gradient de l'erreur dans des systèmes à plusieurs couches[2, 3].

1.2.2 Neurone formel

Les premières formulations du réseau de neurones remontent à 1943, dans les travaux de Mc Culloch et Pitts . L'idée est de reproduire le fonctionnement d'un neurone humain. Le fonctionnement d'un neurone formel est simple : c'est une somme pondérée d'entrées à quelle on applique une fonction d'activation, les coefficients de pondération sont appelés poids synaptiques et la fonction d'activation utilisée était initialement un seuil[20] .

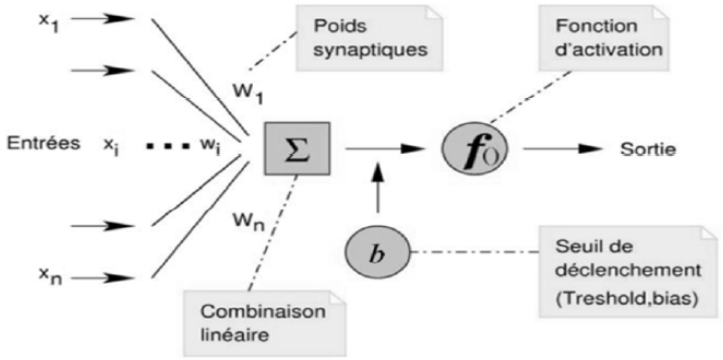

FIGURE 1.1: Structure générale d'un neurone formel

Entrées(X_i) : sont directement les entrées du système ou peuvent provenir d'autres couches précédentes.

Biais ($b = W_0$) : les entrées qui sont toujours mises à 1 qui permet d'ajouter de la flexibilité au réseau en permettant de varier le seuil de déclenchement du neurone par l'ajustement du poids du biais lors de l'apprentissage .

Poids(W_i) : sont les facteurs multiplicateurs qui affectent l'influence de chaque entrée sur la sortie du neurone.

Noyau(Somme pondérée + Fonction d'activation) «$F(x) = \sum(X_i W_i) - W_0$ » Intègre toutes les entrées et le biais et calcule la sortie du neurone selon une fonction d'activation qui est souvent non linéaire pour donner une plus grande flexibilité d'apprentissage. La figure 1.2 définit les principaux types de fonction d'activations.

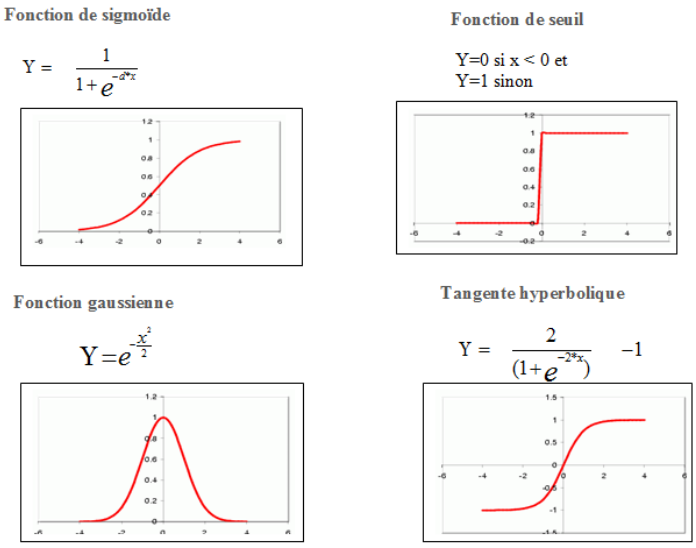

FIGURE 1.2: Quelques types de fonctions d'activations

Sortie(y) Directement une des sorties du système où peut être distribuée vers d'autres neurones($Y = \sum(X_i * W_i) + W_0$) .[2]

- Les réseaux de neurones formels sont connectés entre eux de différentes manières.[5]

1.2.3 Les différentes structures des réseaux de neurones

Dans les architectures des réseaux de neurone, on remarque qu'il existe deux types du structure, structure d'un réseau à couche simple et un réseau multicouches .

1.2.3.1 Structure d'un réseau de neurone à couches simples

Le réseau à couches est un réseau dont les neurones sont organisés en couches, la forme la plus simple est le réseau à une seule couche .Tous les signaux d'entrée sont propagés des nœuds d'entrée vers la couche de neurones de sortie. L'exemple le plus connus de ce type c'est le modèle de F. Rosenblatt (perceptron monocouche)[9].

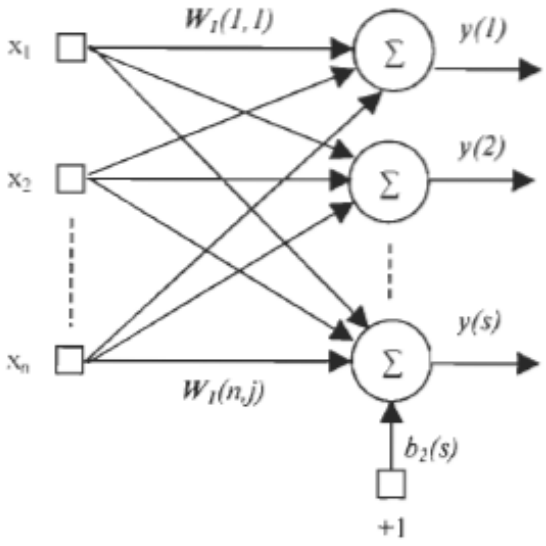

FIGURE 1.3: Réseau de neurones mono-couche

1.2.3.2 Structure d'un réseau de neurones multi-couches

Dans ce cas, les neurones sont arrangés par couches, il n'y a pas de connexion entre les neurones d'une même couches, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante [2]. La première couche est appelée couche d'entrée : recevra les données source que l'on veut utiliser pour l'analyse. Sa taille est donc directement déterminée par le nombre de variables d'entrées. Et les couches du milieu sont appelées couches cachées chaque neurone de cette couche reçoit l'information de plusieurs couches précédentes, les neurones de ces couches cachées appliquent deux traitements : Une combinaison linaire de leurs entrées (dont les poids sont des paramètres du réseau), suivie par une fonction d'activation. Par la suite, il envoie cette réponse aux neurones de la couche suivante. Le choix de sa taille (nombre de neurones) n'est pas automatique et doit être ajusté. Il sera souvent préférable pour obtenir le résultat optimale, d'essayer le plus des tailles possibles. La troisième couche est appelée couche de sortie, elle joue le même rôle que les couches cachées, la seule différence entre ces deux types de couches est que la sortie des neurones de la couche de sortie n'est liée à aucun autre neurone, elle donne le résultat obtenu après compilation par le réseau des données entrée dans la première couche. Sa taille est directement déterminée par le nombre de variables dont on a besoin en sortie. Permet les structures les plus utilisées de ce type on trouve le perceptron

multicouche[3, 5].

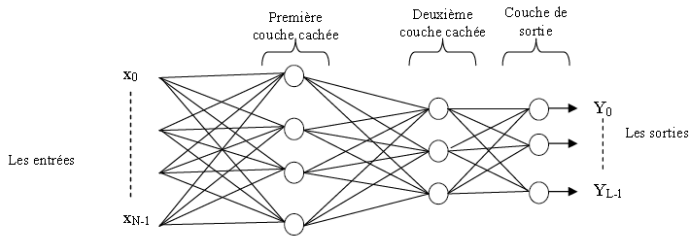

FIGURE 1.4: Réseau de neurones multi-couches

1.2.4 Les types des réseaux de neurones artificiel

Un réseau est défini par sa topologie, qui représente le type de connexion existant entre les divers neurones du réseau[5].

1.2.4.1 Les réseaux de neurones non bouclés

Dans un tel réseau le flux d'information circule des entrées vers les sorties sans retour en arrière. Si l'on représente le réseau comme un graphe dont les nœuds sont les neurones et les arêtes les « connexions » entre ceux-ci, le graphe d'un réseau non bouclé est acyclique. Tout neurone dont la sortie est une sortie du réseau est appelé « neurone de sortie ». Les autres, qui effectuent des calculs intermédiaires, sont des « neurones cachés ». L'un des types de réseaux de neurones c'est les réseaux à couche, le réseau de neurones à une couche cachée et une sortie linéaire est un cas particulier de ce dernier type[3].

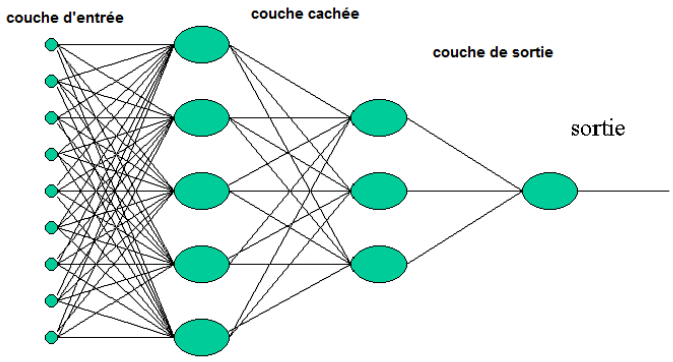

FIGURE 1.5: Réseau de neurone non bouclé

Parmi les réseaux de neurones non bouclé on trouve :

1. **Réseau à connexions locales**

   Il s'agit d'une structure multicouche, conserve une certaine topologie. Chaque neurone a des relations avec un nombre réduit et localisé des neurones de la couche précédent. Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique[51].

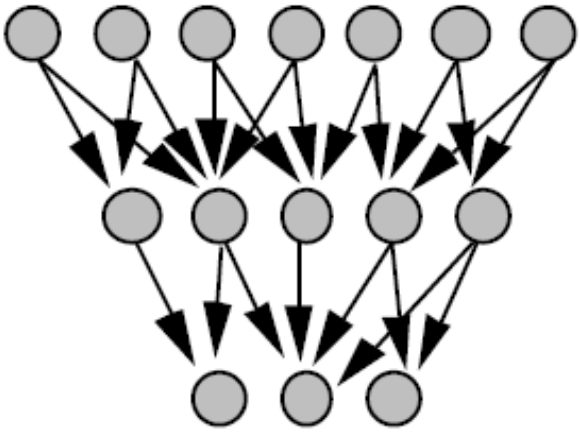

FIGURE 1.6: Réseau à connexions locales

1.2.4.2 Les réseaux de neurones bouclés

L'architecture la plus générale pour un réseau de neurones est le « réseau bouclé», dont le graphe des connexions est cyclique : lorsqu'on se déplace dans le réseau en suivant le sens des connexions, il est possible de trouver au moins un chemin qui revient à son point de départ (un tel chemin est désigné sous le terme de « cycle »). La sortie du neurone peut donc être entrée d'elle-même. L'opérateur non linéaire réalisé par un réseau non bouclé, dépend des valeurs des coefficients de pondération C_{ij} (du neurone j vers le neurone i) du réseau. Pour qu'un réseau effectue une tâche donnée, il faut donc ajuster la valeur de ses coefficients. Une tâche est définie par un ensemble d'exemples, ou couples (valeurs des entrées . Valeurs des sorties désirées correspondantes),ces couples constituent l'ensemble d'apprentissage. La procédure d'ajustement des coefficients de manière que les sorties du réseau soient aussi proches que possibles des sorties désirées est appelée apprentissage. Le principe général des algorithmes d'apprentissage repose sur la minimisation d'une fonction de coût quadratique des différences entre les sorties du réseau et les sorties désirées[3, 46].

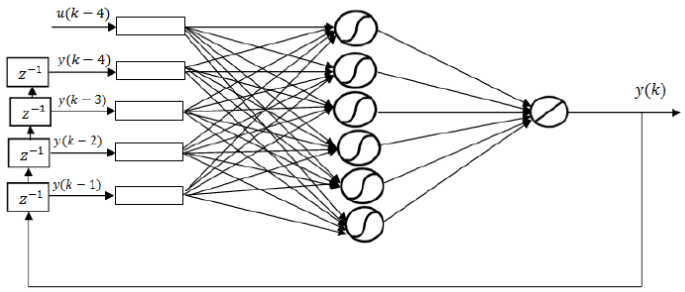

FIGURE 1.7: Réseau de neurones bouclé

Permis les réseaux non boucle on trouve :

1. **Réseau à connexion complète**
   C'est la structure d'interconnexion la plus générale .Chaque neurone est connecté à tous les neurones du réseau et à lui-même[51].

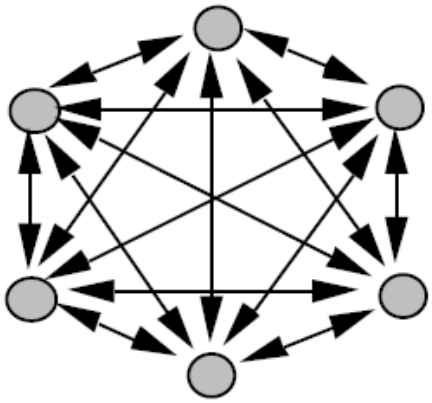

FIGURE 1.8: Réseau à connexions complète

2. **Réseau à connexions récurrentes**
   Les connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouche . Ces connexions sont le plus souvent locales[51].

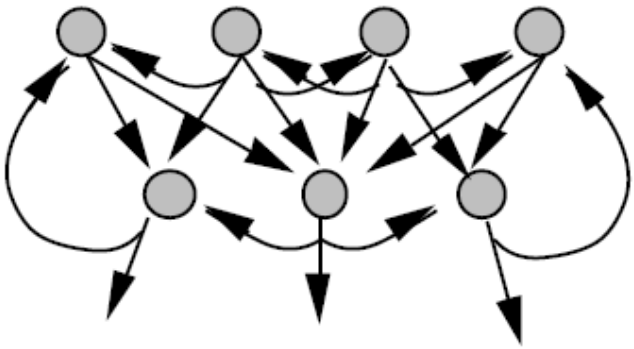

FIGURE 1.9: Réseau de neurone récurrentes

1.2.5 Apprentissage des réseaux de neurones

L'objectif de la phase d'apprentissage des réseaux de neurones est de trouver, parmi toutes les fonctions paramétrées par les poids synaptiques, celle qui s'approche le plus possible de l'optimum. La capacité à apprendre un caractéristique fondamental de l'intelligence, d'un autre côté ce que veut dire l'apprentissage est parfois difficile à décrire, le processus d'apprentissage, dans les contextes des réseaux de neurones peut être vu comme un problème

ajustement et d'adaptation de l'architecture et des poids des réseaux de neurones de façon que le RNA exécute efficacement une tâche spécifique. Au lieu de spécifier un ensemble de règles, les réseaux de neurones apprennent automatiquement à partir de la collection d'exemples, et c'est ce qui les rend très attractives. - Il existe deux types d'apprentissage des réseaux de neurones : l'apprentissage supervisé et l'apprentissage non supervisé[5, 10].

1.2.5.1 Apprentissage par rétro propagation de l'erreur

C'est seulement en 1986 que la généralisation de la règle delta aux réseaux à couches cachées a été formulée. Généralement, la rétro-propagation du gradient de l'erreur consiste à propager l'erreur obtenue à une unité de sortie d'un réseau à couches comportant une ou plusieurs couches cachées à travers le réseau par descente du gradient dans le sens inverse de la propagation des activations, La figure 1.10 montre une illustration du principe de la rétro propagation.

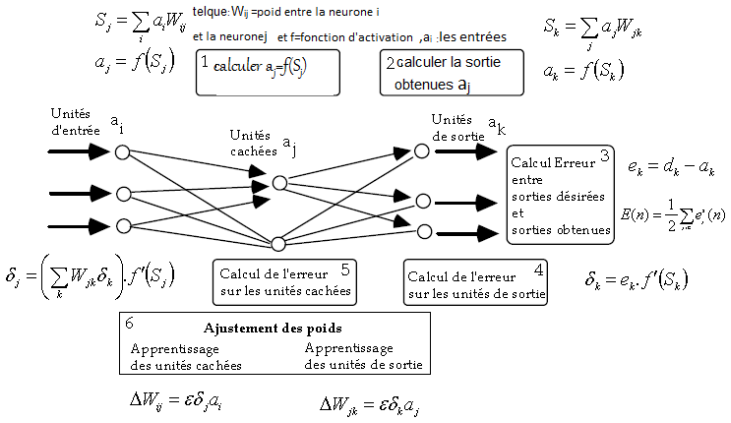

FIGURE 1.10: Rétro-propagation du gradient

1.2.5.1.1 Déroulement de l'algorithme de rétro-propagation On considère un réseau de neurones multicouches : une couche d'entrée, une couche cachée et une couche de sortie. n_0 entrées,n_1 neurones avec une fonction d'activation non linéaire (fonction sigmoïde) dans la couche cachée et n_2 neurones linéaires dans la couche de sortie[2].

1. **Propagation**
   Après l'initialisation des poids, on calcule les sorties du réseau en propageant les valeurs d'entrée X_i de couche en couche , la figure 1.11 illustre cette opération.

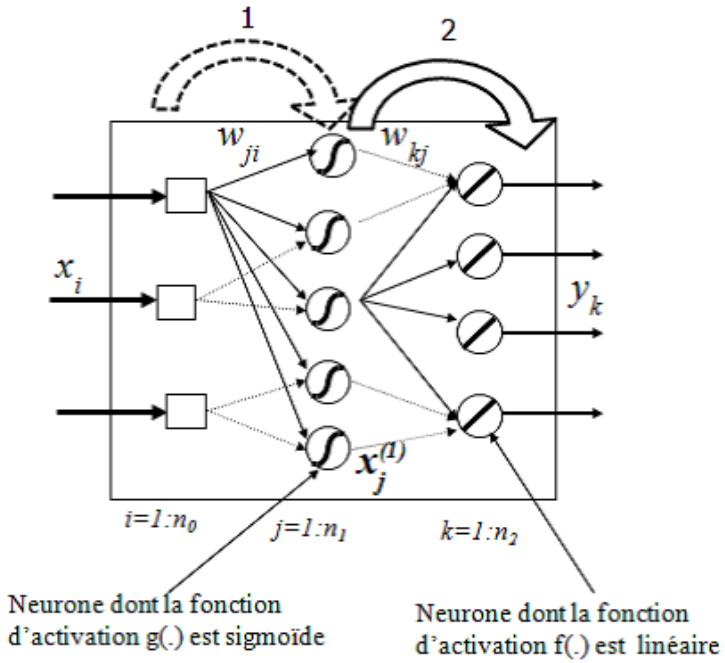

FIGURE 1.11: Propagation des entrées

Etape1 :
La valeur de la sortie de chaque neurone de la couche cachée $X_{j1}(j = 1...n_1)$, dépend de la somme des entrées $x_i(i = 1...n_0)$ pondérées par les poids W_{ji} entre la couche d'entrée et la couche cachée. La fonction d'activation non linéaire g() détermine l'état du neurone

$$\begin{cases} a_j^{(1)} = \sum_{i=0}^{n_0} W_{ji} X_i \\ X_j^1 = g(a_j^{(1)}) \end{cases} \tag{1.1}$$

g(.) étant la fonction d'activation des neurones de la couche cachée.
Etape2 :
De même on calcule la valeur de la sortie de chaque neurone de la couche de sortie $Y_k(k = 1...n_2)$, Elle dépend de la somme des sorties de la couche cachée qu'on a calculée dans l'étape1 $(x_j^(1))$ pondérées par les poids entre la couche cachée et la couche de sortieW_{kj}

$$\begin{cases} a_k^{(2)} = \sum_{j=0}^{n_1} W_{kj} X_j \\ y_k = f(a_k^{(2)}) \end{cases} \tag{1.2}$$

F(.) Etant la fonction d'activation des neurones de la couche de sortie.

2. **Sommaire de l'algorithme (règle du "delta")**

L'algorithme de rétro propagation standard se résume donc à la série d'étapes suivantes :

1. Initialiser tous les poids à des petites valeurs aléatoires dans l'intervalle [-o.5 ;0.5].

2. Normaliser les données d'entraînement.

3. Permuter aléatoirement les données d'entraînement.

4. Pour chaque donnée d'entraînement n :

(a) Calculer les sorties observées en propageant les entrées vers l'avant.

(b) Ajuster les poids en rétro propageant l'erreur observée

$$W_{ji}(n) = w_{ji}(n-1) + \Delta w_{ji}(n) = w_{ji}(n-1)\eta\delta_j(n)y_i(n) \qquad (1.3)$$

Où le "gradient local" est défini par :

$$\delta_j(n) = \begin{cases} e_j(n)y_j(n)[1-y_j(n)] & si \quad j \in couchedesortie \\ y_j(n)[1-y_j(n)]\sum_k \delta_k(n)w_{kj}(n) & si \quad j \in couchecache \end{cases} \qquad (1.4)$$

Avec $0 \leq \eta \leq 1$ représentant le taux d'apprentissage et $y_i(n)$ représentant soit La sortie du neurone i sur la couche précédente, si celui-ci existe, soit l'entrée i autrement.
5. Répéter les étapes 3 et 4 jusqu'à un nombre maximum d'itérations ou jusqu'à ce que la racine de l'erreur quadratique moyenne (EQM) soit inférieure à un certain seuil.

1.3 L'apprentissage profond

Jusqu'à tout récemment, la grande majorité de la recherche en apprentissage automatique se concentrait sur des architectures peu profondes.
Y. Lecun et al [33] Indiquent que les architectures peu profondes comme les

Perceptrons multicouches ou les machines à vecteurs de supports sont mal adaptées pour la résolution de problèmes complexes comme la vision artificielle. Le problème des architectures peu profondes (au plus deux couches) est qu'elles peuvent nécessiter un nombre exponentiel d'éléments sur la couche cachée. Les réseaux profonds sont composés de plusieurs couches caché avec un nombre élevée de neurones, l'apprentissage du réseau de neurones complet par rétro-propagation du gradient fait converger le réseau vers un minimum local. Le Deep Learning, ou « apprentissage profond », correspond donc aux recherches permettant d'entraîner un réseau comprenant plusieurs couches internes. En général, une architecture profonde est composée de plusieurs couches non-linéaires paramétrées, possèdent un meilleur pouvoir de représentation que les architectures peu profondes. Chaque couche permet une représentation de plus haut niveau que la précédente. On espère donc apprendre une représentation abstraite des données, dans laquelle la tâche à résoudre sera plus facile[23, 15, 24].

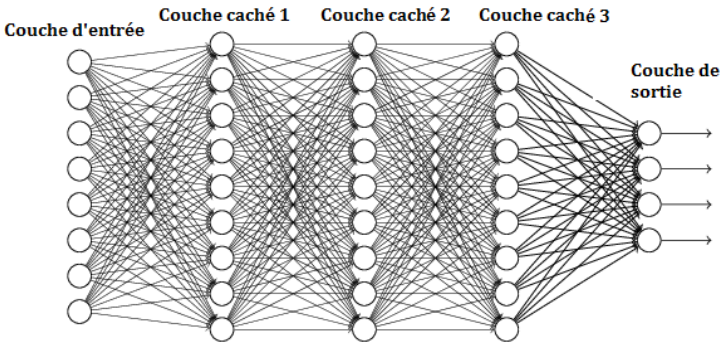

FIGURE 1.12: L'apprentissage en profondeur

1.3.1 Définition de réseau de neurone profond

Le Deep Learning ou « apprentissage profond » est une famille d'apprentissage automatique, est un réseau de neurone non bouclé avec de nombreuses couches cachées. La théorie est que si le réseau de neurone est autorisé à trouver des représentations significatives à plusieurs niveaux, la fonction l'apprentissage en profondeur est le mieux. Le premier niveau caché pourrait représenter des bords ou des traits, le second serait une combinaison de bords / traits, c'est-à-dire des coins / cercles, etc., chaque couche voyant des modèles dans les niveaux inférieurs et représentant de plus en plus des concepts

abstraits.[33]
Les architectures majeures des réseaux profonds[44, 57] :
• Réseau Auto encoder
• Réseaux de croyances profondes (Deep belief networks)
• Les machines de Boltzmann restreintes (RBM)
• Réseaux de neurones à convolutions
• Réseaux de neurones récurrents .

1.3.2 Réseaux de neurones à convolutions

Les réseaux de neurones à convolution sont à ce jour les modèles les plus performants pour la classification. Désignés par l'acronyme CNN, de l'anglais Convolutional Neural Network, son objectif est d'apprendre des fonctionnalités d'ordre supérieur dans les données via des convolutions[40]. Leur conception suit la découverte de mécanismes visuels dans les organismes vivants. Ces réseaux de neurones à convolution sont capables de catégoriser les informations des plus simples aux plus complexes. Ils consistent en un empilage multicouche de neurones, des fonctions mathématiques à plusieurs paramètres ajustables, qui pré-traitent de petites quantités d'informations. Les réseaux convolutifs sont caractérisé par leurs premières couches convolutionelles (généralement une à trois). Une couche convolutive, est basée comme son nom l'indique sur le principe mathématique de convolution, et cherche à repérer la présence d'un caractéristique (dans un signal ou dans une image par exemple). Les réseaux neuronaux convolutifs ont de nombreuses applications dans la reconnaissance d'images, de vidéos ou le traitement du langage naturel[12, 44].

1.3.2.1 Architecture générale du CNN

Une architecture CNN est formée par un empilement de couches de traitement indépendantes :
-La couche de convolution (CONV) qui traite les données d'un champ récepteur.
-La couche de pooling (POOL), qui permet de compresser l'information en réduisant la taille de l'image intermédiaire (souvent par sous-échantillonnage).
-La couche de correction (ReLU), souvent appelée par abus 'ReLU' en référence à la fonction d'activation (Unité de rectification linéaire).
-a couche "entièrement connectée" (FC), qui est une couche de type perceptron.
-La couche de perte (LOSS).

Un exemple : En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a 2 dimensions pour une image en niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu], une image est passée de couche d'entrée jusqu'à la couche de sortie à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques. Ce vecteur de caractéristique est ensuite branché en entrée d'une deuxième partie, constituée de couches entièrement connectées (perceptron multicouche). Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image. La sortie est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories[44, 12, 40] .

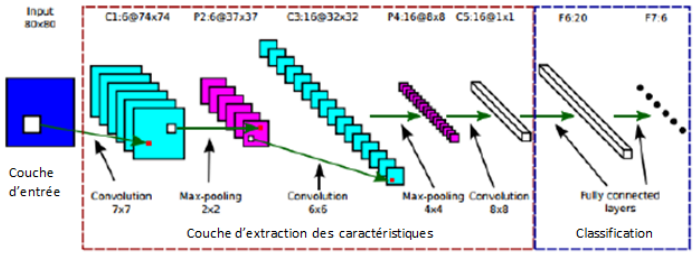

FIGURE 1.13: Architecture générale d'un réseau de neurone à convolution

Dans ce dessous, un explication détaillée des parties de réseaux de neurone à convolution.

1.3.2.1.1 Couches d'entrée :

La couche d'entrée accepte un entrée généralement sous la forme spatiale de la taille (largeur hauteur), si l'entrée est un image un champ de profondeur supplémentaire représente les canaux de couleurs [44].

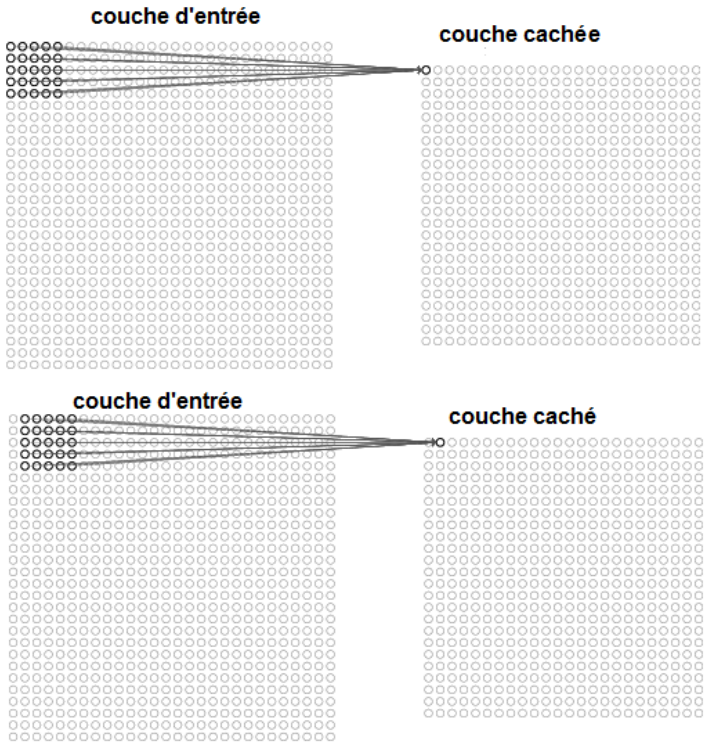

FIGURE 1.14: Représentation de la connexion entre la matrice de la couche entrée et La première couche caché

1.3.2.1.2 Couches d'extraction des caractéristiques :

1. **Couches convolutives** :
   . Les couches de convolution sont considérées comme les éléments de base des architectures CNN . Ces couches contient un bloc de construction commencer avec un ensemble de filtres qui était étendu à travers toute la profondeur du volume d'entrée [44].
   Une convolution est définie comme une opération mathématique décrivant une règle pour la fusion de deux ensembles d'informations. Il applique un noyau de convolution appelé Kernel ou bien filtres sur le volume d'entrée pour donne comme un sortie une carte de caractéristiques . L'opération de convolution est illustrée dans la figure 1.15, est connue sous le nom de détecteur des caractéristiques d'un CNN. L'entrée d'une convolution peut être une donnée brute ou une sortie de carte des caractéristiques d'une autre convolution [40].

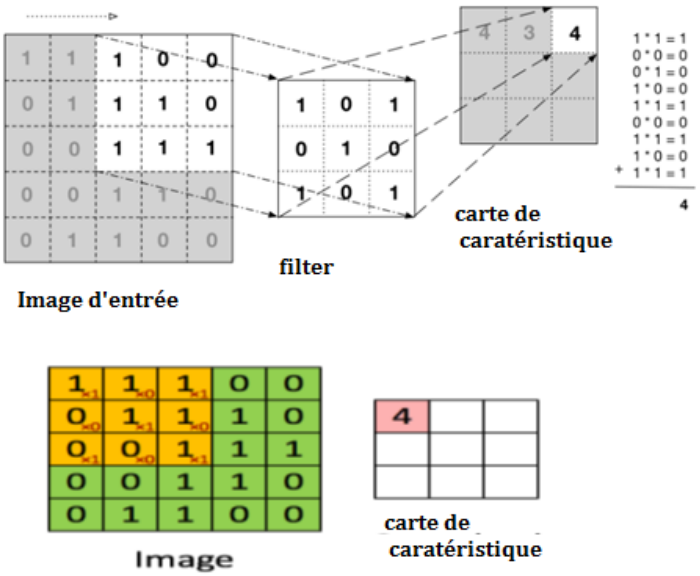

FIGURE 1.15: L'opération de convolution

La figure 1.15 illustre comment le noyau est glissé à travers les données d'entrée pour produire les cartes des caractéristiques (sortie). A chaque étape, le noyau est multiplié par les valeurs de données d'entrée dans ses limites, créant une seule entrée dans la carte des caractéristiques en sortie. En pratique, la sortie est grande si la caractéristique que nous recherchons est détectée dans l'entrée. Les couches de convolution ont des paramètres et des hyperparamètres supplémentaires. La descente en gradient est utilisée pour entraîner les paramètres de cette couche de sorte que les scores de classe soient cohérents avec les étiquettes de l'ensemble d'apprentissage.
Voici les principaux composants des couches convolutives [44] :
• Filtres
• Cartes d'activation
• Hyperparamètres spécifiques aux couches

(1) **Filtres**
Les filtres sont une matrice de poids dans laquelle sa largeur et sa hauteur sont inférieure à la largeur et à la hauteur du volume d'entrée.
Des filtres sont appliqués sur la largeur et la hauteur du volume d'entrée d'une manière glissante, comme le montre la figure(1.16), sont également appliqués pour chaque profondeur du volume d'en-

trée. Un produit scalaire entre le filtre et la région d'entrée pour obtenir la carte d'activation.

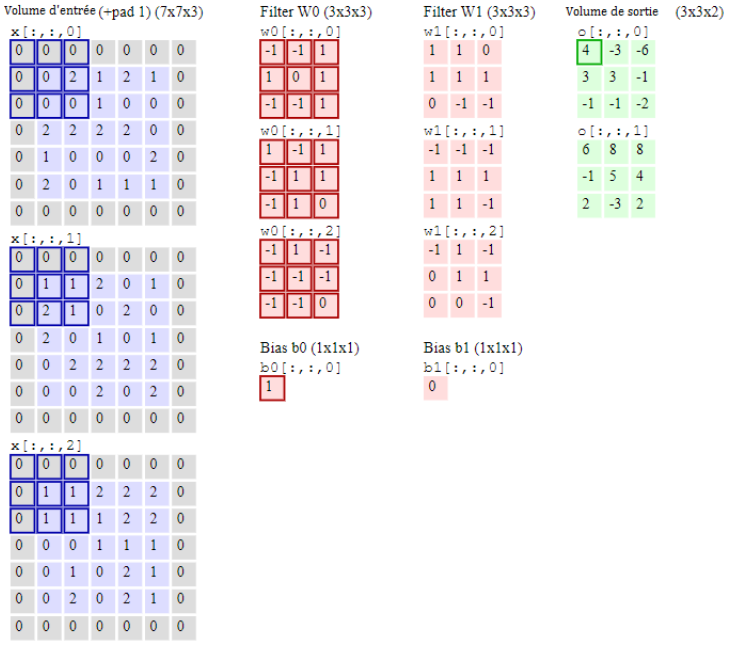

FIGURE 1.16: Filtre de convolution

(2) **Les Cartes d'activation**
La carte d'activation est un matrice contient les sorties de produit scalaire entre le filtre et le volume d'entrée .

(3) **Hyperparamètres spécifiques aux couches**
Voici les hyperparamètres qui organiser l'arrangement spatial et la taille du volume de sortie d'une couche de convolution :

- Taille du filtre (ou du noyau)

- Profondeur de la couche

- Le pas

-La marge (à 0) ou zéro padding

a- Taille du filtre

Chaque filtre est petit dans l'espace par rapport à la largeur et la hauteur des données d'entrée.

b-Profondeur de la couche

Profondeur de la couche c'est le sortie de la convolution c'est-à-dire le nombre des cartes de caractéristiques.

c- Le pas

contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.

d-La marge (à 0) ou zero padding
parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. La taille de ce 'zero-padding' est le troisième hyper paramètre. Cette marge permet de contrôler la dimension spatiale du volume de sortie.

2. **Maxpooling / subsampling**

En plus des couches de convolution, les réseaux de neurones à convolution contiennent également des couches de pooling. Les couches de pooling sont généralement utilisées immédiatement après les couches de convolution. Ce que les couches de pooling font est de simplifier l'information dans la sortie de la couche de convolution. En détail, une couche de pooling prend chaque sortie de carte des caractéristiques de la couche de convolution et prépare une carte des caractéristiques condensée. Il y a plusieurs façons de faire cette mise en commun, comme prendre la moyenne ou le maximum, ou une combinaison linéaire apprise des neurones dans le bloc. Par exemple, la figure(1.17) montre max pooling sur une fenêtre 2 2. Un grand avantage est qu'il y a beaucoup moins de fonctionnalités groupées, ce qui permet de réduire le nombre de paramètres nécessaires dans les couches ultérieures. Max-pooling n'est pas la seule technique utilisée pour pooling. Une autre approche courante est connue sous le nom de l2pooling. Ici, au lieu de prendre l'activation maximale d'une région de neurones, on prend la racine carrée de la somme des carrés des activations dans la région ,alors que les détails sont différents, l'intuition est similaire à

max-pooling : l2pooling est un moyen de condenser les informations de la couche convolutionnelle. En pratique, les deux techniques ont été largement utilisées [22, 21].

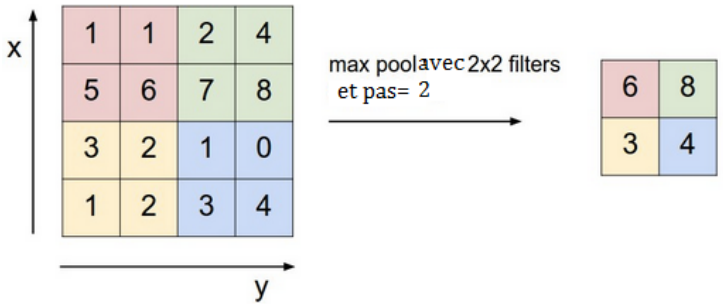

FIGURE 1.17: l'operation de Maxpooling

1.3.2.1.3 La fonction d'activation Il est possible d'améliorer l'efficacité du traitement en intercalant entre les couches de traitement et une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie. La fonction ReLU (abréviation de Unités Rectifié linéaires) : F(x)=max(0,x) Cette fonction force les neurones à retourner des valeurs positives[44].

1.3.2.1.4 Couches classification La sortie de la partie convolutive est ensuite branchée en entrée d'une deuxième partie, constituée de couches entièrement connectées .
La structure d'une couche entièrement connectée est la même que celle d'une couche dans un réseau de neurones simple, utiliser cette couche pour calculer les scores de classe utilisent comme sortie du réseau. les couches entièrement connectées effectuent des transformations sur le volume de données d'entrée qui sont fonction des activations dans le volume d'entrée et des paramètres (poids et biais des neurones)[44].

1.3.2.1.5 Dropout Les couches "FC" (Fully Connected) occupent la majeure partie de la mémoire du CNN. D'ailleurs le concept de FC crée un problème exponentiel de mémoire appelé "overfitting" ("sur-connexion" conduisant au sur-apprentissage) ralentissant le traitement de l'information. Pour prévenir cela, la méthode du dropout est utilisée pour "éteindre" les neurones aléatoirement (avec une probabilité prédéfinie, souvent un neurone

sur deux) ainsi que les neurones périphériques. Ainsi, avec moins de neurones, le réseau est plus réactif et peut donc apprendre plus rapidement. À la fin de la séance d'apprentissage, les neurones "éteints" sont "rallumés" (avec leurs poids originaux). Plus la couche FC est proche de l'image source, moins on éteindra de neurones[40].

1.3.2.1.6 Couche de perte (LOSS) La couche de perte spécifie comment l'entrainement du réseau pénalise l'écart entre le signal prévu et réel. Elle est normalement la dernière couche dans le réseau. Diverses fonctions de perte adaptées à différentes tâches peuvent y être utilisées. La fonction « Softmax » permet de calculer la distribution de probabilités sur les classes de sortie[40].

1.4 Conclusion

Grâce à des bases de données colossales, l'apprentissage à profondeur permet de produire des performances exceptionnelles, bien meilleures que celles réalisées jusqu'à présent, en réduisant considérablement le taux d'erreurs. Ses applications se sont multipliées, notamment en matière de reconnaissance du visage, de voiture autonome, ou encore de traitement de l'imagerie médicale. L'utilisation du l'apprentissage à profondeur dans ce dernier domaine va entraîner une révolution en médecine dans les années à venir. On commence également à assister à une révolution similaire concernant la compréhension et le traitement de la langue naturelle. L'apprentissage à profondeur est surtout devenu la pièce maîtresse de l'industrie de l'Internet, qui l'utilise notamment pour le filtrage des contenus et des résultats de recherche.
Dans le chapitre suivant on va définir le domaine d'analyse de texte , le processus et les techniques utiliser ainsi que des exemples du domaine d'application .

CHAPITRE

2

ANALYSE DE TEXTE

2.1 Introduction

L'analyse de textes ou bien la fouille de texte (text mining) est l'héritière directe de la fouille de données (data-mining), introduite au milieu des années quatre-vingt-dix sous le terme Knowledge Discovery in Textual Databases (KDT) ou Text Data Mining (TDM), puis traduit en français par Extraction des Connaissances à partir des Textes (ECT).
A cette époque, les ordinateurs personnels se généralisent, leur capacité de calcul et de mémorisation atteignent des seuils tels qu'ils commencent à pouvoir traiter des grandes quantités d'informations. La complexité inhérente à la quantité énorme et croissante de données textuelles peut maintenant être approchée efficacement, ce qui permet la découverte des relations intéressantes dans les documents, e-mails, pages Web et autres sources d'informations non structurées. Le besoin des approches de l'analyse des textes efficaces devient plus important si l'on considère que, la plupart des informations organisationnelles disponibles dans les entreprises modernes sont présentées sous forme textuelle.
les systèmes des analyses de texte et les opérations de pré-traitement sont centrées sur l'identification et l'extraction des informations représentatives pour les documents en langage naturel. Ces opérations de pré-traitement sont responsables de la transformation des données non structurées stockées dans les collections de documents en un format intermédiaire plus explicitement structuré, ce qui n'est pas pertinent pour la plupart des systèmes de fouille du donnée [26, 18, 16].
Dans ce chapitre , on va définir le domaine d'analyse du texte et le processus, les techniques qu'on va suivre ainsi que les types des problèmes résolus et quelques exemples d'analyses de texte et on va terminer avec les travaux connexes.

2.1.1 Définition Analyse des textes

: L'ère de l'information a conduit au développement d'une grande variété d'outils et d'infrastructures pour capturer et stocker des quantités massives de données textuelles. Il n'est pas pratique pour un individu (ou un groupe d'individus) de traiter des grandes données textuelles et d'extraire des informations, des sentiments ... etc.
L'analyse de textes est un ensemble de processus permettant, à partir d'un ensemble des ressources textuelles, de construire des connaissances pouvant être représentées dans un langage formel et appliquer sur des textes en différent format : classification, regroupement, recherche dans des document, ... etc [50, 4, 13].

2.1.2 Processus d'analyse des textes

L' d'analyse des textes est un processus itératif comprenant une liste de six processus successifs (Figure 2.1)

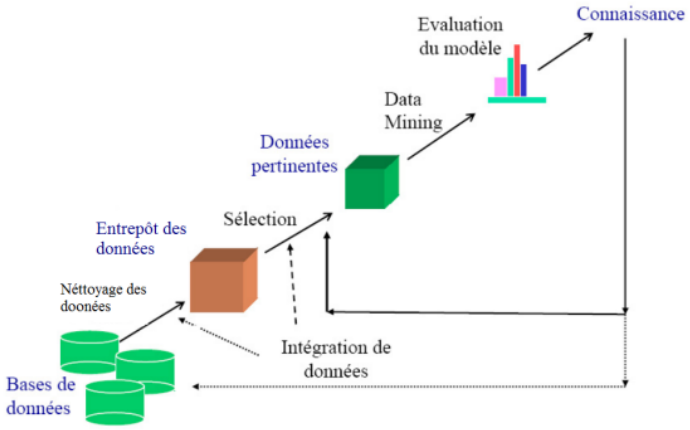

FIGURE 2.1: Processus d'analyse des textes

1. **Pré-traitement(Nettoyage des données)** : Après la première opération que doit effectuer un système d'analyse des textes à savoir la reconnaissance des termes utilisés, expurger le plus possible les informations inutiles des documents afin que les connaissances gardées soient aussi pertinentes qu'il se peut.
   La phase de pré-traitement commence par une tokenisation qui est la division d'un document en termes. En effet dans les documents textuels des nombreux mots apportent peu d'informations sur le document concerné. Les algorithmes dits de "Stop Words" s'occupent de les éliminer. Un autre traitement nommé "Stemming" permet également de simplifier les textes tout en augmentant leurs caractères informatifs .
   Les mots restants sont saisis pour la sélection et la réduction des caractéristiques. Une fois caractéristiques(termes) réduites, les documents sont représentés en format vectoriel et sont prêts pour l'analyse[37, 17, 53].
   L'étape de prétraitement du texte est en outre divisée en :
   a.tokenisation

b.Suppression de stopword
c.lemmatisation
d.stemming.

2. **Intégration de données** : Le processus d'intégration de données combine les données provenant de différentes sources. Les données d'entrées généralement sont collectées à partir de plusieurs bases de données ayant chacune une structure et une définition de données distincte. Dans ce cas, le processus d'intégration de données insère les données dans un seul magasin de données (entrepôt des données) cohérent à partir de ces sources de données multiples[42].
3. **Sélection de données** :Tente de trouver le sous-ensemble optimal d'un ensemble de données. A partir des sources de données le processus de sélection de données récupère les termes jugées pertinentes pour les utilisées dans les étapes suivantes du processus d'analyse de données des données[42, 19].

4. **Transformation de données** : Dans la phase de transformation, les données sources vont être converties en format approprié pour le processus d'extraction des connaissances[42].
5. **Analyse de données** : Dans le processus d'analyse de données, des méthodes intelligentes sont appliquées afin d'extraire les modèles de données, par exemple : pour résoudre un tel problème de prédiction, les données sont d'abord converties en instances d'une représentation définie, puis transmises à un algorithme d'apprentissage automatique. Les modèles trouvés sont ensuite généralisés, de sorte qu'ils peuvent être appliqués pour déduire de nouvelles informations à partir de données non vues[56, 42].

6. **Évaluation de modèle** : L'évaluation du modèle est la tâche de découvrir des modèles intéressants parmi un ensemble des modèles et des connaissances extraites[42].
7. **Présentation de la connaissance** : La représentation des connaissances comprend des techniques de visualisation, qui sont utilisées pour interpréter la connaissance découverte à l'utilisateur. généralement, l'objectif de l'analyse de donnée est d'aider à la prise de décision en fournissant des modèles compréhensibles aux utilisateurs. En effet, les utilisateurs ne demandent pas des pages et des pages de chiffres, mais des interprétations des modèles obtenus. Les expériences montrent que les modèles simples sont plus compréhensibles

mais moins précis, alors que ceux complexes sont plus précis mais difficiles à interpréter[35, 42].

2.1.3 Représentation des textes

Les textes en langage naturel ne peuvent pas être directement interprétés par un classifieur ou par les algorithmes de classification. Ces derniers ne sont pas capables de traiter directement ces textes. Ainsi une étape de représentation numérique est nécessaire. La représentation mathématique généralement utilisée est l'utilisation d'un espace vectoriel comme espace de représentation cible. La caractéristique principale de cette représentation est que chaque terme est associée à une dimension propre au sein de l'espace vectoriel[4].

2.1.3.1 Représentation en « sac de mots »

L'idée est de transformer les textes en vecteurs dont chaque composante représente un mot. Les mots ont l'avantage de posséder un sens explicite. Il faut alors gérer les sigles, ainsi que les mots composés , ceci nécessite un pré-traitement linguistique. On peut choisir de conserver les majuscules pour aider, par exemple, à la reconnaissance de noms propres, mais il faut alors résoudre le problème des débuts de phrases. Les composantes du vecteur sont une fonction de l'occurrence des mots dans le texte. Cette représentation des textes exclut toute analyse grammaticale et toute notion de distance entre les mots : c'est pourquoi cette représentation est appelée « sac de mots » (voir la figure 2.2)[27].

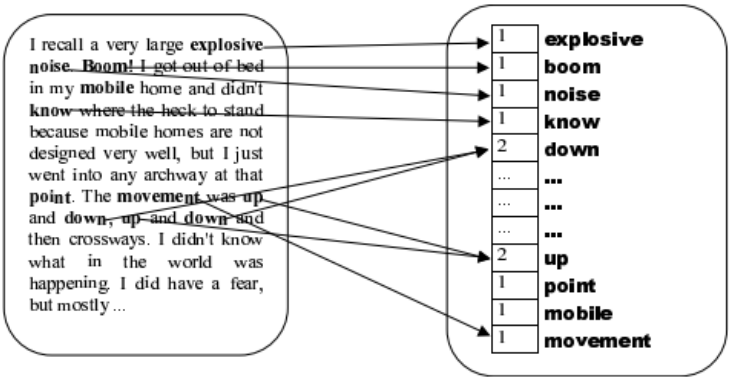

FIGURE 2.2: Exemple de la représentation d'un texte en sac de mots

2.1.3.2 Représentation des textes avec des racines lexicales et des lemmes

Dans le modèle précédent (représentation en « sac de mots »), chaque flexion d'un mot est considérée comme un descripteur différent et donc une dimension de plus, ainsi, les différentes formes d'un verbe constituent autant de mots. Par exemple : les mots « déménageur, déménageurs, déménagement, déménagements, déménager, dé-ménage, déménagera, etc. » sont considérés comme des descripteurs différents alors qu'il s'agit de la même racine « déménage », les techniques de désuffixation (ou stemming), qui consistent à rechercher les racines lexicales, et de lemmatisation cherchent à résoudre cette difficulté.
Pour la recherche des racines lexicales, plusieurs algorithmes ont été proposés, l'un des plus connus pour la langue anglaise est l'algorithme de Porter [39]. La lemmatisation consiste à remplacer les verbes par leur forme infinitive, et les noms par leur forme au singulier. Un algorithme efficace, nommé Tree-Tagger [47], a été développé pour les langues anglaise, française, allemande et italienne[27].

2.1.3.3 Représentation des textes par des phrases

Malgré la simplicité de l'utilisation des mots comme unité de représentation, certains auteurs proposent plutôt d'utiliser les phrases comme unité. Les phrases sont plus informatives que les mots seuls, par exemple : « machine learning » ou « world wide web » car les phrases ont l'avantage de conserver l'information relative à la position du mot dans la phrase.
Logiquement, une telle représentation doit obtenir de meilleurs résultats que ceux obtenus via les mots. Mais les expériences présentées ne sont pas concluantes car, si les qualités sémantiques sont conservées, les qualités statistiques sont largement dégradées : le grand nombre de combinaisons possibles entraîne des fréquences faibles et
On ne considère pas toutes les séquences possibles mais on tente d'effectuer une sélection des phrases, en privilégiant celles qui sont sémantique ment riche. Dans la phrase "Le gentil lapin orange mange la carotte bleue" par exemple, on peut dire que des séquences comme "gentil lapin orange","carotte bleue","lapin orange", ... sont porteuses de sens. Alors que les séquences "orange mange","le gentil" ...etc, sont insignifiantes. Une autre approche de Caropreso et les autres [8] qui proposent d'utiliser des phrases statistiques comme descripteurs au lieu des phrases grammaticales qui ont amélioré considérablement la performance du classifieur. Une phrase statistique est définie comme une collection de mots adjacents qui apparaissent ensembles mais qui

ne respectent pas forcement les règles grammaticales.
Une autre étude menée par Scott et les autres [48] démontre que ce type de représentation améliore la qualité des résultats par rapport aux méthodes de type « sac de mots » lorsque les documents étudiés sont limités en nombre et taille. De plus, ce type de représentation présente de grandes variations dans la qualité de ses résultats en fonction du type de documents à classifier[27, 37].

2.1.4 Techniques d'analyse des textes

L'analyse de Textes est utilisée pour extraire des informations, des connaissances ou des modèles intéressants à partir des documents non structurés provenant de différentes sources. Il convertit les mots et les phrases en informations non structurées en valeurs numériques qui peuvent être liées à des informations structurées dans la base de données et analysées avec des techniques anciennes de la fouille de données, c'est l'analyse des données contenues dans texte en langage naturel. La figure montre les techniques utilisent dans l'analyse de textes[54].

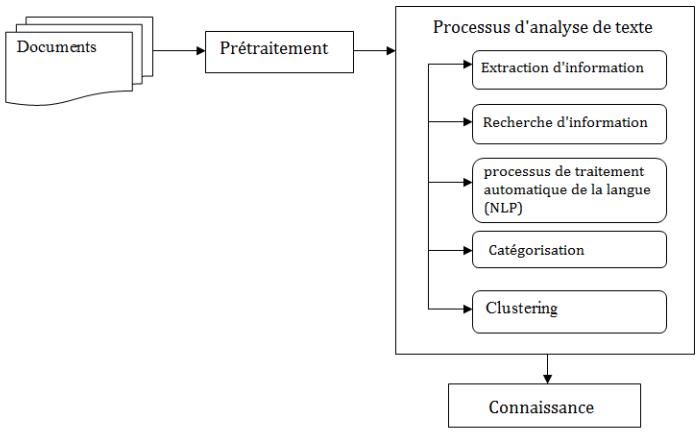

FIGURE 2.3: Les techniques utilisent pour l'analyse des textes

2.1.4.1 L'extraction d'information

Extraction d'information (EI) implique directement avec le processus de fouille du texte, permet extraire des informations utiles à partir des textes. EI traite de l'extraction d'entités, d'événements et des relations spécifiés à partir des sources de texte non restreintes. L'IE peut être décrit comme la création d'une représentation structurée d'informations sélectionnées tirées

de textes[58].

Les informations extraites fournissent des données plus concises et plus précises, telles que celles utilisées pour la catégorisation des textes, et les informations tendent à représenter les concepts et les relations, qui sont plus significatives et se rapportent directement au domaine du document examiné.
Les éléments pouvant être extraits du texte
Il existe quatre types d'éléments de base qui peuvent actuellement être extraits du texte.

Entités :Les entités sont les éléments de base qui peuvent être trouvés dans les documents texte, par exemple : les personnes, les entreprises, les lieux, les gènes et les médicaments,...etc.

Les attributs : Les attributs sont des caractéristiques des entités extraites, par exemple : les attributs sont le titre d'une personne, l'âge d'une personne et le type d'organisation.

Faits :Les faits sont les relations qui existent entre les entités, exemple : un relation de travail entre une personne et une entreprise .

Événements : Un événement est une activité ou une occurrence d'intérêt dans laquelle des entités participent comme un acte terroriste, une fusion entre deux entreprises, un anniversaire...etc[18].

2.1.4.2 La Recherche d'Information (RI)

La Recherche d'Information est une discipline de recherche qui intégré des modèles et des techniques dont le but est de faciliter l'accès à l'information pertinente pour un utilisateur ayant un besoin en information. Ce besoin en information est souvent formulé en langage naturel par une requête décrite par un ensemble de mots-clés. L'objectif de tout Système de RI (SRI) est alors de retrouver dans une collection de documents ceux qui sont susceptibles d'être pertinents à une requête . L'enjeu majeur ne se situe donc pas dans la « compréhension » d'un document mais sur la capacité du système à distinguer un document d'un autre. La présence d'un mot, d'un mot-clé ou d'un terme dans un document peut ainsi être pondérée, par exemple, par le TF-IDF (Term Frequency - Inverse Document Frequency) reflétant son pouvoir discriminant. Les temps de réponse doivent être courts et les méthodes de représentation vectorielle des documents sont très performantes.

Un concept de base pour la recherche d'information est de mesurer la similarité : une comparaison est faite entre deux documents, mesurant à quel point les documents sont similaire. une mesure de similarité raisonnable est la distance de cosinus, qui met l'accent sur les propriétés des textes qui se produisent réellement .
Les utilisateurs peuvent soumettre des requêtes de recherche sur l'index pour récupérer les informations les plus pertinentes pour les termes de la requête. Les champs de méta-données générés par les analyseurs peut être utilisé dans les index pour activer différents types de fonctionnalités pour l'interrogation, un index de recherche contient un entrée pour chaque propriété mesurée. Chaque entrée pointe vers tous les textes pertinents en ce qui concerne la propriété.

2.1.4.3 Le Processus de traitement automatique de la langue

Le traitement du langage naturel est le problème le plus difficile dans le domaine de l'intelligence artificielle. C'est l'étude du langage humain pour que les ordinateurs puissent comprendre des langages naturels semblables à ceux des humains. Le traitement du langage naturel est concerné par la génération du langage naturel (NLG) et la compréhension du langage naturel (NLU). NLG s'assure que le texte généré est grammaticalement correct et fluide. La plupart des systèmes NLG incluent un réalisateur syntaxique pour s'assurer que les règles grammaticales telles que l'accord du verbe sujet sont respectées et le planificateur de texte pour décider comment organiser les phrases, le paragraphe et les autres parties de manière cohérente. Le meilleur NLG connu application est la traduction automatique. NLU est constitué d'au moins un des composants suivants : tokenization, analyseur lexical, analyseur syntaxique et analyseur sémantique[41, 11].

2.1.4.4 Catégorisation

La Catégorisation de Textes (ou Text Categorization - CT) est une tâche particulière de l'analyse des textes. Son objectif est d'affecter automatiquement un texte à une catégorie définie au préalable en se basant sur le contenu du texte. C'est une tâche qui remonte au début des années 60. En raison de l'augmentation de la disponibilité des documents sous forme numérique et de la nécessité d'accéder en souplesse, celle-ci a acquis un statut important dans le domaine de la recherche d'information au début des années 90.

L'avantage de la CT pour ces méthodes est qu'il existe souvent un corpus d'apprentissage contenant des textes déjà classés. C'est typiquement le cas où un expert a déjà élaboré les catégories, a indexé et classé quelques documents.

Nous pouvons distinguer trois types de catégorisation de textes :
-Catégorisation binaire : ce type de catégorisation correspond au filtrage, elle permet, par exemple, de répondre aux questions suivantes : « le document est pertinent ou non ? », « le courriel est un spam ou non » ?
-Catégorisation multi catégories : c'est le cas le plus général de la catégorisation à n classes. Le système doit affecter 0, 1 ou plusieurs catégories à un même document.
- Catégorisation multi catégories disjointes : c'est une catégorisation à n classes mais le document doit être affecté à une et une seule catégorie[4].

2.1.4.5 Clustering

Le clustering (regroupement) est une technique utilisée pour grouper des documents similaires, mais diffère de la catégorisation en ce que les documents sont regroupés sans utilisation des sujets prédéfinis.
Dans le contexte de l'analyse de texte, le regroupement divise la collection des documents en groupes mutuellement exclusifs en fonction de la présence des thèmes similaires. Dans la plupart des applications métier impliquant des grandes quantités de données textuelles, il est souvent difficile de profiler chaque groupe en lisant et en examinant manuellement tout le texte d'un groupe. Au lieu de cela, le thème d'un groupe est identifié en utilisant un ensemble de termes descriptifs que chaque groupe contient. Ce vecteur de termes représente les poids mesurant la façon dont le document s'inscrit dans chaque groupe. Les thèmes aident à mieux comprendre le client, les concepts ou les événements. Le nombre de groupe identifiées peut être contrôlé par l'analyste.

2.1.5 Des Exemples sur les applications d'analyses de texte

1. **Bioinformatique**
   Les industries des sciences de la vie et des soins de santé génèrent une grande quantité de données textuelles et numériques concernant les dossiers des patients, les maladies, les médicaments, les symptômes et les traitements des maladies et bien d'autres encore. C'est un grand

défi de filtrer un texte approprié et pertinent pour prendre une décision à partir d'un grand dépôt biologique. Les dossiers médicaux sont de nature variable, le vocabulaire complexe, long et technique est utilisé pour rendre le processus de découverte des connaissances très difficile. Les outils de fouille de texte dans le domaine biomédical fournissent une occasion d'extraire des informations précieuses, leur association et inférer la relation entre diverses maladies, espèces et gènes. L'utilisation d'outils d'analyse de texte appropriés dans le domaine médical aide à évaluer l'efficacité des traitements médicaux qui démontrent l'efficacité en comparant différentes maladies, symptômes et leur traitement[29, 14, 49].

2. **Intelligence d'affaires :**

   L'analyse de texte joue un rôle important dans l'intelligence d'affaires qui aide les organisations et les entreprises à analyser leurs clients et leurs concurrents pour prendre de meilleures décisions . Le risque de faire une mauvaise prédiction devrait être réduit. La plupart des techniques de fouille de données sont créées pour gérer la prédiction . Le problème avec la fouille de données est qu'elle ne peut aider que jusqu'à un certain point, puisque la plupart des données sont disponible dans les textes (rapports, notes de service, courriels, document de planification, etc.). Les techniques de fouille de données et de fouille de texte peuvent se compléter les unes les autres. Il donne un aperçu plus approfondi des affaires et donne des informations sur comment améliorer la satisfaction du client et obtenir des avantages compétitifs[29, 49].

3. **la sécurité sur Internet**
   L'utilisation de l'outil du fouille de texte dans le domaine de la sécurité est devenue fait important. Un grand nombre de logiciels de la fouille de texte est commercialisé pour des applications de sécurité, en particulier la surveillance et l'analyse des sources de texte en ligne, telles que internet nouvelles, blogs, courrier électronique,... etc. à des fins de sécurité . C'est aussi impliqué dans l'étude du cryptage / décryptage de texte[29, 14, 49].

4. **Bibliothèques numériques**

   Des nombreuses techniques et outils d'analyse des textes sont utilisés pour déterminer les modèles et les tendances des journaux et des pro-

cédures à partir d'une quantité immense de référentiels. Ces sources d'information aident dans le domaine de la recherche et du développement. Les bibliothèques sont une excellente source d'information pour les chercheurs, les bibliothèques numériques tentent de comprendre l'importance de leur collection. Il fournit une nouvelle méthode d'organisation de l'information de telle sorte qu'il est possible de disposer des trillions des documents disponibles en ligne. Il fournit une nouvelle façon d'organiser l'information et permet d'accéder à des millions de documents en ligne. Bibliothèque numérique internationale GreenStone qui prend en charge plusieurs langues et interfaces multilingues fournissent une méthode élastique pour extraire des documents qui traitent plusieurs formats, à savoir Microsoft Word, PDF, PostScript, HTML, langages de script et messages électroniques[29].

5. **Domaine académique et de recherche**

   Dans le domaine de l'éducation, divers outils et techniques d'analyse des textes sont utilisés pour analyser les tendances de l'éducation dans une région spécifique, l'intérêt des étudiants pour un domaine spécifique et le taux d'emploi. L'utilisation de l'analyse des textes dans le domaine de la recherche aide à trouver et classifier documents de recherche et matériel pertinent de différents domaines en un seul endroit. L'utilisation de k-means clustering et d'autres techniques aident à identifier les attributs des informations pertinentes. La performance des élèves dans différents sujets peut être consulté et comment différents attributs affectent la sélection des sujets[49].

2.1.6 L'apprentissage profond et Analyse de texte

L'apprentissage profond est une nouvelle méthode d'apprentissage dans l'analyse de texte, elle peut améliorer la vitesse d'extraction des informations textuelles afin d'accéder rapidement aux connaissances souhaitées.
L'utilisation des techniques d'apprentissage profond pour extraire des représentations de données significatives permet d'obtenir des caractéristiques sémantiques à partir de telles données textuelles de grande dimension, ce qui conduit également à la réduction des dimensions des représentations des données .
Hinton et les autres [25] , écrivent un modèle génératif d' apprentissage profond pour apprendre les codes binaires des documents. La couche inférieure

du réseau d'apprentissage profond représente le vecteur de comptage des mots du document qui représente des données de grande dimension, tandis que la couche supérieure représente le code binaire appris du document.
Dans le traitement du langage naturel, une grande partie du travail avec des méthodes d'apprentissage profond a impliqué l'apprentissage de représentations des vecteurs de mots à travers des modèles de langage neuronal et d'effectuer la composition sur les vecteurs des mots appris pour la classification.
L'outil "word2vec" de Google est une autre technique d'extraction automatisée de représentations sémantiques à partir de Big Data. Cet outil prend en entrée un corpus de texte à grande échelle et produit les vecteurs de mots en sortie. Il construit d'abord un vocabulaire à partir des données de texte d'apprentissage et apprend ensuite la représentation vectorielle des mots, sur laquelle le fichier vectoriel peut être utilisé comme caractéristiques dans des nombreuses applications de traitement du langage naturel (NLP) et d'apprentissage automatique. Les réseaux de neurones à convolution (CNN) utilisent des couches avec des filtres de convolution qui sont appliqués aux fonctionnalités locales, les modèles CNN ont ensuite montré être efficace pour la PNL et ont obtenu d'excellents résultats dans l'analyse sémantique analyse syntaxique , recherche des requêtes , modélisation des phrases , et d'autres tâches PNL[43, 31].

2.1.7 Les Travaux Connexes

La classification des documents est nécessaire pour organiser les documents pour la récupération, l'analyse, la conservation et l'annotation. Les chercheurs ont étudié et développé diverses méthodes de classification des documents. Travailler dans la communauté de recherche d'information a mis l'accent sur les fondamentaux des moteurs de recherche tels que l'indexation et les dictionnaires qui sont considérés comme des technologies de base dans ce domaine . Des travaux plus récents ont employé des méthodes de fouille de données et d'apprentissage automatique. Parmi les plus précises de ces techniques, la machine à vecteurs de support (SVM) .
Les SVM utilisent les fonctions du noyau pour trouver des hyperplans de séparation dans des espaces de grande dimension. SVM et les méthodes associées sont difficiles à interpréter. Pour cette raison, de nombreux systèmes de recherche d'information utilisent des arbres de décision et des méthodes naïve Bayes. Ces méthodes sont plus faciles à comprendre et, en tant que telles, peuvent prendre en charge la réformulation des requêtes, mais elles manquent de précision. Certains travaux récents ont étudié la modélisation des sujets pour fournir des interprétations similaires aux méthodes de naïve

Bayes mais avec une précision améliorée . L'apprentissage en profondeur a été largement utilisé pour le traitement d'images, mais de nombreuses études récentes ont appliqué dans d'autres domaines tels que l'analyse de texte et des données. L'architecture de base dans un réseau de neurones est un réseau entièrement connecté de non-linéaire les nœuds de traitement organisés en couches. La première couche est la couche d'entrée, la couche finale est la couche de sortie et toutes les autres couches sont caché. Dans notre travail, nous intéressons sur les réseaux de neurone à convolution (CNN) . CNN fournissent d'excellents résultats dans la généralisation de la classification des objets dans les images . Des travaux plus récents ont utilisé des CNN pour l'analyse texte. Zhang et al. utilisaient des CNN pour la classification de texte. Indépendamment d'application, les réseaux CNN requièrent des grands ensembles d'apprentissage. Une autre architecture fondamentale d'apprentissage profond utilisée est le réseau de neurone récurrent (RNN). Les RNN connectent la sortie d'une couche à son entrée. Cette architecture est particulièrement importante pour apprendre . Ces méthodes d'apprentissage en profondeur ont la promesse de fournir une plus grande précision que SVM et les méthodes connexes[39].

2.2 Conclusion

Des nombreuse des tâches de la fouille du textes ont étés proposées dans la littérature parmi elles on trouve la catégorisation automatique de textes qui s'impose comme un clé de technologie dans la recherche et l'extraction d'information, où de nos jours, la demande d'outils de recherche permettant de gérer la surcharge d'information a conduit à un intérêt pour la catégorisation automatique des textes dans l'espoir de réduire le travail humain de façon significative, et à résoudre des problèmes d'accès à l'information voulu .

Dans le chapitre suivant, nous allons décrire l'architecture de notre système.

CHAPITRE

3

CONCEPTION

3.0.1 Introduction

La catégorisation du texte est la tâche de l'analyse des textes, qui a de nombreuses applications importantes telles que l'analyse des sentiments et la catégorisation des sujets, regroupement ,...etc, Récemment, plusieurs variantes des réseaux neuronaux convolutifs (CNN) ont été montré pour atteindre une grande précision sur la catégorisation de texte pour cela nous allons proposer un processus d'analyse de texte.

3.0.2 Objectif de notre travail

Notre objectif est de réaliser un système capable de classifier les textes dans des grandes collections des documents en utilisant le réseau de neurones à convolution profond.

3.0.3 Conception globale du système

Globalement, notre système suit les étapes suivantes pour construire un modèle de classification des textes :
la première étape est un pré-traitement des données textuelle, le résultat de cette étape est un représentation vectorielle des textes qui était l'entrée de réseau du neurone à convolution, finalement, on construire un modèle de classification du texte .
On peut schématiser la conception est schématisé comme suit :

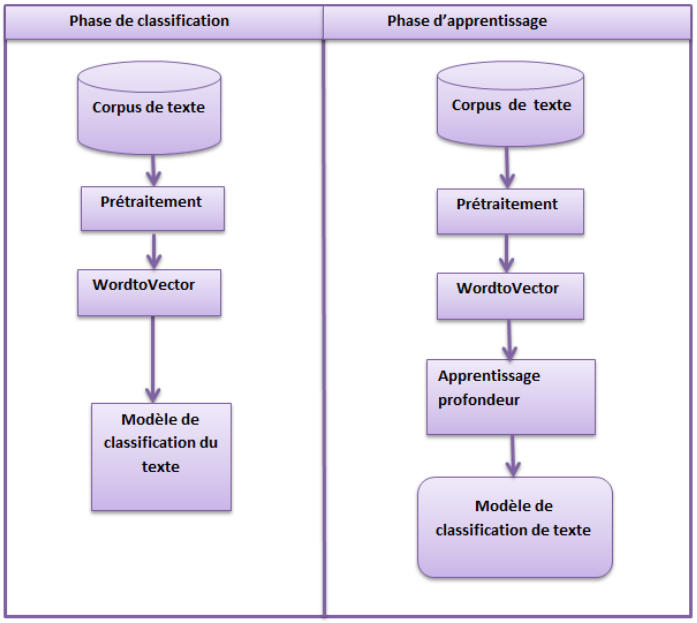

FIGURE 3.1: La conception globale du système de classification

La conception globale comprend deux phase : la phase d'apprentissage et la phase de classification.
Les deux phases contient les étapes suivant : la collecte des textes nécessaires à l'analyse, la deuxième étape c'est le prétraitement qui consiste à extraire, nettoyer, analyser les mots extraits pour identifier les entités, supprimer les mots d'arrêt et vérifier l'orthographe. L'étape la plus importante après l'analyse(prétraitement) est la transformation de texte en vecteur, cette étape traite la représentation numérique du texte, généralement la méthode la plus utilisé pour la représentation vectorielle du texte est Word2Vector.
La phase d'apprentissage contient un étape d'apprentissage de CNN qui consiste à prendre en entrés les vecteurs et produire en sortie un modèle de classification du texte.

3.0.4 Conception détaillée du système

Pour la conception détaillée, Notre système comprend quatre étapes principales qui sont : Préparation des données textuelle, pré-traitement, l'apprentissage de CNN et finalement le modèle de classification.

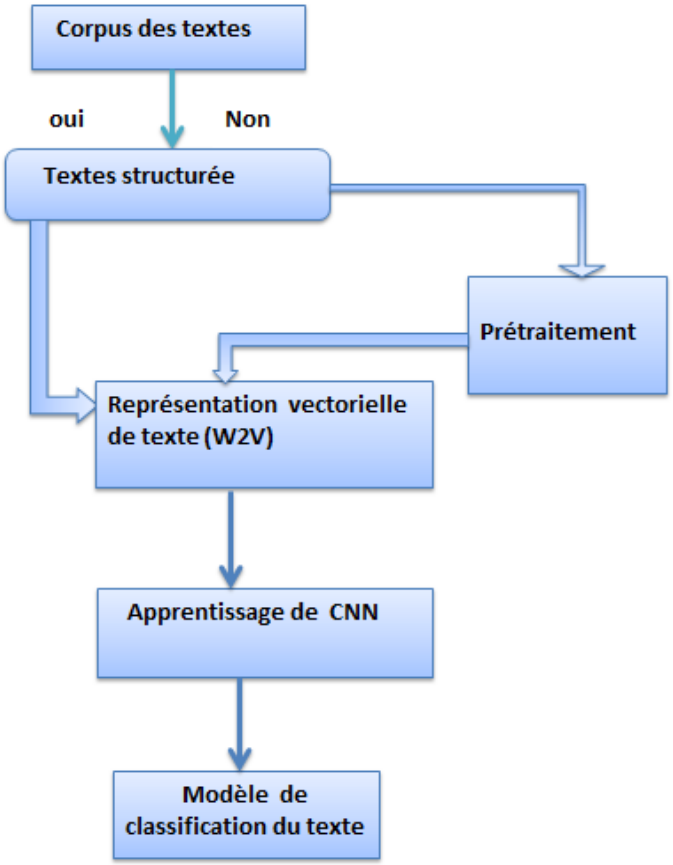

FIGURE 3.2: Architecture détaillée du système

3.0.4.1 Prétraitement de texte

L'entrée du système est un ensemble de documents. Un document est un texte, par exemple un e-mail, chapitre dans un livre, réponse à une question ouverte, ... etc). Le prétraitement est très important dans le processus de classification car la construction du modèle est basée sur les données préparées. En effet les textes qui ne sont pas préparées correctement peut de donner un modèle non performant[34].
Ainsi le prétraitement comprend plusieurs étapes illustrer dans la figure(3.3)

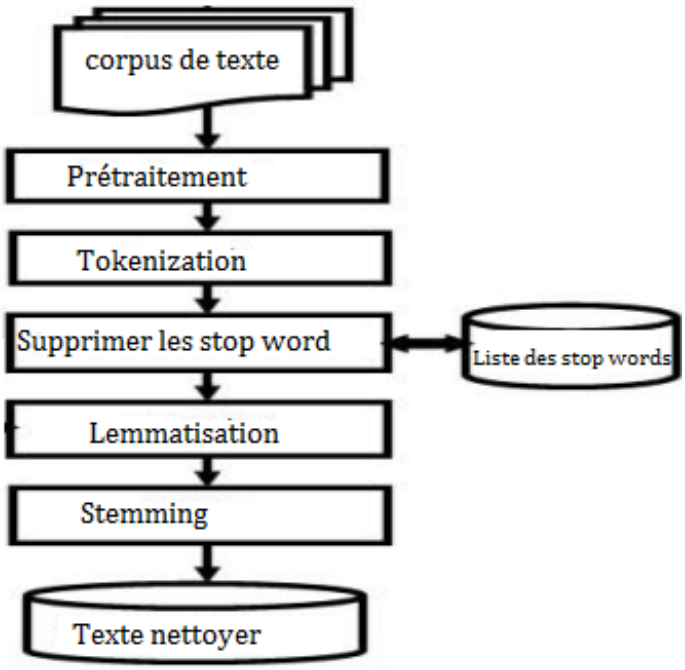

FIGURE 3.3: Processus de prétraitement

1. **Tokenization**

   Tokenization est la tâche de l'analyse du contenu d'un document consiste à identifier les éléments qui le composent. Cela peut être fait par un lexer. Un lexer prend une séquence de caractères en entrée et la divise en jetons qui sont généralement les mots. Une approche simple pour ce faire est de simplement diviser tous les caractères non alphanumériques. Les signes de ponctuation sont un autre problème. Le pire est le chinois où il n'y a pas d'espaces blancs. Une autre façon d'effectuer la tokenisation est d'analyser le texte en fonction d'un ensemble prédéfini des règles. Un exemple d'une telle application est flex qui génère la source code pour un lexer en C[34].

2. **Elimination de stop words**

   Elimination de stop words ou le filtrage consiste à supprimer tous les mots standards dans un texte, ce sont des mots très communs et utilisés dans pratiquement tous les textes. Leur présence peut dégrader la performance de l'algorithme de classification en termes de coût et en termes de précision de la classification. Nous pouvons prendre comme exemple pour l'anglais I, am, with, what, you,...etc.

Et pour le français il s'agit d'éliminer par exemple :
- Les pronoms pronominaux : il, elle, je, lui, ceux....
- Les auxiliaires : avoir, être et toutes leurs conjugaisons possibles .
- Les opérateurs de conjonction : avec, et, ou, aux...
- L', c' ...

3. **Lemmatisation :**
   La lemmatisation est la tâche qui considère le l'analyse morphologique des mots, c'est-à-dire regrouper les différentes formes d'un mot afin qu'elles puissent être analysées comme un seul article. En d'autres termes, les méthodes de lemmatisation tentent de cartographier les formes verbales infinitive tendu et noms à une seule forme.

4. **Stemming**
   Les méthodes de stemming visent à obtenir la racine de mots dérivés. Les algorithmes d'étalement sont en effet dépendants du langage. L'algorithme le plus couramment utilisé est l'algorithme de Porter.

3.0.4.2 word2vector

Word2vec est une méthode fondée sur des réseaux de neurones artificiels. Cette méthode propose deux architectures de réseaux de neurones : l'architecture en "sac-de-mots" continus (CBOW) et l'architecture Skip-gram. Ces deux architectures se présentent sous la forme de réseaux de neurones artificiels simples. Ils sont constitués de trois couches : une couche d'entrée, une couche cachée et une couche de sortie. La couche d'entrée contient soit un "sac-de-mots" (CBOW), soit un mot seul (Skip-gram). La couche cachée correspond à la projection des mots d'entrée dans la matrice des poids. Cette matrice est partagée par tous les mots (matrice globale). Enfin, la couche de sortie est composée de neurones “softmax”. Pour des raisons de complexité algorithmique due à la couche "softmax". Le couplage de ces fonctions avec la simplicité de ces réseaux leur permettent d'être entraînés sur des très grandes quantités de textes, et ainsi d'obtenir des modélisations de meilleure qualité que les modèles plus complexes à base de récurrence ou de convolution[28].

1. **Approche par sac-de-mots continus (CBOW)**

   L'architecture CBOW est un réseau de neurones devant prédire un mot à partir de son contexte. La couche d'entrée représente la pré-

sence ou l'absence des mots dans le contexte de manière binaire (i.e. 1 pour la présence, 0 pour l'absence).

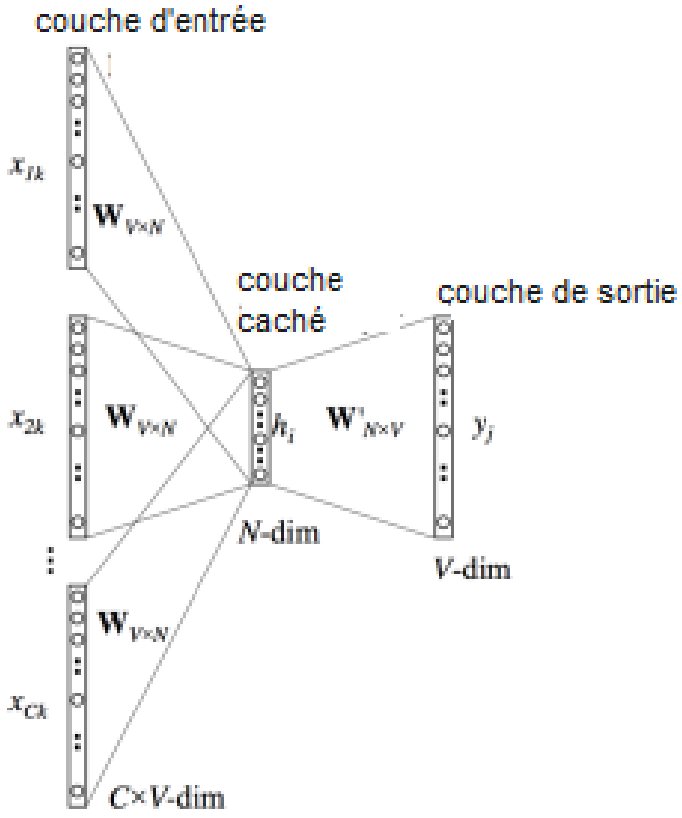

FIGURE 3.4: Modèle général de CBOW

Chaque mot dans le contexte est projeté dans la matrice des poids du modèle. La somme (ou la moyenne) de ces représentations passe ensuite par la couche de sortie. Enfin, le modèle compare sa sortie avec le mot seul et corrige sa représentation par rétro-propagation du gradient. Ce modèle cherche à maximiser l'équation :

$$\frac{1}{T}\sum_{t=1}^{T}\log p(m_{t-\frac{c}{2}}...m_{t+\frac{c}{2}}) \tag{3.1}$$

où T correspond à l'ensemble des mots dans le corpus et c correspond à la taille de la fenêtre du contexte de chaque mot.

2. **Approche par Skip-Gram**
   L'architecture Skip-Gram tente de prédire, pour un mot donné, le contexte dont il est issu. La couche d'entrée de ce réseau est alors un vecteur ne contenant qu'un seul mot. Le mot est projeté dans la couche cachée puis dans la couche de sortie. Le contexte est ensuite réduit de façon aléatoire à chaque itération. Le vecteur de sortie est ensuite comparé à chacun des mots du contexte réduit et le réseau se corrige par rétro-propagation du gradient. De cette manière, la représentation du mot d'entrée va se rapprocher de chacun des mots présents dans le contexte. Le réseau de neurones Skip-gram essaie de maximiser une variation de l'équation de CBOW comme suit :

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{j=t-c,j\neq t}^{t+c}\log p(m_j|m_t) \tag{3.2}$$

   Comparativement au CBOW, cette architecture permet une meilleure modélisation des mots peu fréquents et permet de mieux capturer les relations sémantiques.

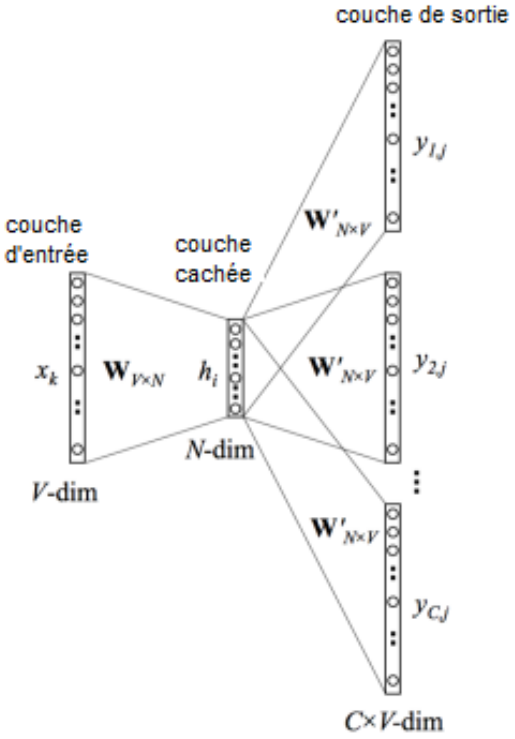

FIGURE 3.5: Modèle général de Skip-Gram

3.0.4.3 Apprentissage de CNN

À l'étape suivante, un réseau de neurone profond est formé avec les données appropriées, c'est le réseau de neurone à convolution a défini le format d'entrée et de sortie. L'entrée est un espace vectoriel du texte et la sortie est la classe de texte. Un CNN a été implémenté parce qu'il a été prouvé qu'il est possible d'obtenir des résultats impressionnants en utilisant des convolutions sur la couche d'entrée pour calculer la sortie. Il a différentes couches :

1. Embedding
2. Dropout
3. Block de convolution
4. Concatenate

5. Dropout

6. Fonction d'activation

Dans la couche d'embedding, les poids sont appliqués dans le réseau . Dropout aide à éviter le surajustement en mettant des poids aléatoires à zéro. Ensuite, un bloc de convolution est exécuté plusieurs fois et comprend trois couches :

1. Convolution1D

2. Max pooling

3. Flatten

Le premier, Convolution1D, crée un noyau de convolution (filtre ou détecteur de caractéristiques) qui est convolué avec l'entrée de la couche sur une seule dimension spatiale pour produire un tenseur des sorties. Il a différents paramètres qui spécifient ce qui suit :

— Taille du noyau :liste d'un nombre entier unique, spécifiant la longueur de la fenêtre du convolution 1D.

— Nombre de filtres : Entier, la dimensionnalité de l'espace de sortie (c'est-à-dire le nombre de cartes du caractéristique).

— Unité linéaire rectifiée (ReLU) :fonction d'activation non linéaire .

Ce bloc convolutif est exécuté en parallèle. Ensuite, les résultats sont concaténés puis une fonction d'activation "softmax" est appliquée, elle réduit le nombre de neurones au nombre de sorties possibles.

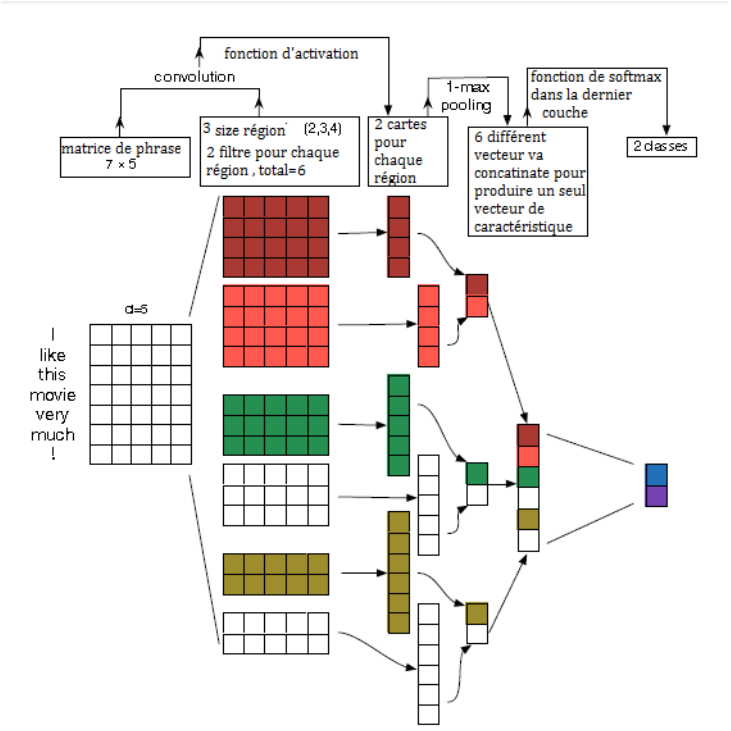

FIGURE 3.6: Schéma de CNN pour la classification des phrases

La figure 3.6 est représente un exemple d'un classification d'une phrase par CNN, la phrase est représenté dans un matrice de taille (75), On trouve dans la partie de convolution trois tailles de régions(2,3,4) chacune ayant 2 filtres. Les filtres sont glisser sur la matrice de phrases et génèrent des cartes de caractéristiques, La mise en commun (max pooling) 1-max est effectuée sur chaque carte. Ainsi, une concaténation est appliquer sur les six cartes et produire un vecteur de caractéristique. La dernière softmax finale reçoit ensuite ce vecteur de caractéristiques en entrée et l'utilise pour classer la phrase, c'est une classification binaire et donc deux états de sortie possibles.

3.1 Conclusion

Dans ce chapitre, nous avons appliqué la méthode du réseau de neurone à convolution pour la classification des textes en proposant une conception générale et une conception détaillée du système. Dans le prochain chapitre, nous présentons l'implémentation de notre système et les résultat obtenus.

CHAPITRE

4

IMPLÉMENTATION

4.1 Introduction

Dans le chapitre précédent, nous avons présenté la conception globale et détaillée d'un système de classification des textes par CNN.
Dans ce chapitre, nous allons présenter en premier temps, le langage de programmation, l'environnement du développement avec les différentes outils utilisées. Ensuite, nous allons exposer l'expérimentation que nous avons appliquées sur la méthode CNN et nous expliquons les résultats obtenus.

4.2 Environnement et outils de programmation

Python : est un langage portable, dynamique, extensible, gratuit, qui permet (sans l'imposer) une approche modulaire et orientée objet de la programmation. Python est développé depuis 1989 par Guido van Rossum [**?**] et de nombreux contributeurs bénévoles.

Nous avons choisi le langage de programmation Python dans notre projet a propos de ses critères :

1. **Python est portable** :il fonctionne non seulement sur Linux, mais aussi sur MacOS, et les différentes variantes de Windows.
2. **Python :** convient aussi bien à des scripts d'une dizaine de lignes qu'à des projets complexes de plusieurs dizaines de milliers de lignes.
3. **La syntaxe de Python :** est très simple et, combinée à des types de données évolués (listes, dictionnaires,...), conduit à des programmes à la fois très compacts et très lisibles. À fonctionnalités égales, un programme Python (abondamment commenté et présenté selon les canons standards) est souvent de 3 à 5 fois plus court qu'un programme C ou C++ (ou même Java) équivalent, ce qui représente en général un

temps de développement de 5 à 10 fois plus court et une facilité de maintenance largement accrue.

4.3 les outils utilisés (package)

Pour implémenter notre projet on besoin de quelques outils :
numpy : Python numérique, contenant la plupart des routines numériques de base telles que la manipulation matricielle, l'algèbre linéaire, l'échantillonnage aléatoire, l'intégration numérique, etc.

nltk : est une plate-forme pour le traitement du langage naturel dans Python, contenant un modèle de sac de mots, un tokenizer, des stemmers, un chunker, des lemmatisateurs, des taggers de parties du discours, etc.

Gensim est une bibliothèque Python pour la modélisation de sujets, l'indexation de documents et la recherche de similarité avec de grands corpus, inclut les implémentations de tf-idf, projections aléatoires, algorithmes word2vec et document2vec,...etc.
TensorFlow ™ est une bibliothèque de logiciels open source pour le calcul numérique haute performance. Son architecture flexible permet un déploiement facile du calcul sur une variété de plates-formes (CPU, GPU, TPU), et des ordinateurs de bureau aux clusters de serveurs aux périphériques mobiles et périphériques. Développé à l'origine par des chercheurs et des ingénieurs de l'équipe Google Brain au sein de l'organisation AI de Google, il bénéficie d'un fort soutien pour l'apprentissage automatique et l'apprentissage en profondeur et le cœur de calcul numérique flexible est utilisé dans de nombreux autres domaines scientifiques.

Keras : est une bibliothèque Python minimaliste pour l'apprentissage profond qui peut fonctionner sur Theano ou TensorFlow. Il a été développé pour rendre la mise en œuvre des modèles d'apprentissage en profondeur aussi rapide et facile que possible pour la recherche et le développement. Keras a été développé et maintenu par François Chollet, un ingénieur de Google utilisant quatre principes directeurs :
Modularité : Un modèle peut être compris comme une séquence ou un graphique seul. Toutes les préoccupations d'un modèle d'apprentissage en profondeur sont des composants discrets qui peuvent être combinés de manière arbitraire.
Minimalisme : La bibliothèque fournit juste assez pour obtenir un résultat, sans fioritures et en maximisant la lisibilité.

Extensibilité : Les nouveaux composants sont intentionnellement faciles à ajouter et à utiliser dans le cadre, destinés aux chercheurs pour tester et explorer de nouvelles idées.
Python : aucun fichier modèle séparé avec des formats de fichiers personnalisés. Tout est Python natif.

ANACONDA est une distribution Python libre qui intègre directement un grand nombre de packages (il n'est donc plus nécessaire de les installer, mais on peut en ajouter d'autres si nécessaire avec le gestionnaire de packages Conda).

4.4 Les bases de textes utilisées pour l'évaluation

4.4.0.1 Reuters

Reuters est un corpus de dépêches en langue anglaise qui a été proposé par l'agence de presse Reuters en 1987. Il correspond à une problématique de classification en plusieurs classes. Deux qualités principales caractérisent les documents de Reuters c'est qu'ils sont courts et plutôt homogènes avec un vocabulaire riche (environ 17000 mots), et la disponibilité gratuitement de la base dans le Web. Du fait que la distribution des classes pour ces documents est très faussée, deux sous-collections sont généralement considérées pour les tâches de classification de texte :
R10 - L'ensemble des 10 classes ayant le plus grand nombre d'exemples d'entraînement positif.
R90 - L'ensemble des 90 classes avec au moins un exemple positif d'entraînement et de test.
De plus, beaucoup de ces documents sont classés comme n'ayant aucun sujet ou avec plus d'un sujet. Comme le but est de considérer les ensembles de données à étiquetage unique, tous les documents comportant moins d'un ou plus d'un sujet ont été éliminés. Avec cela, certaines des classes dans R10 et R90 ont été laissés sans train ou documents de test. En ne considérant que les documents avec un seul sujet et les classes qui ont encore au moins un train et un exemple de test, il nous reste 8 des 10 classes les plus fréquentes et 52 des 90 originales. Suivant la convention habituelle.

1. **Reuter(R8)** L'ensemble Reuters-21578 R8 se compose de 7674 documents regroupés en 08 classes de sujets (acq, crude, earn, grain, interest, money-fx, ship et trade), 5485 document pour l'apprentissage et 2189 pour le testes.

Catégorie	Document d'apprentissage	Document de teste	Total
acq	1596	696	2292
crude	253	121	374
earn	2840	1083	3923
grain	41	10	51
interest	190	81	271
money-fx	206	87	293
ship	108	36	144
trade	251	75	326

TABLE 4.1: Répartition des documents par catégorie, avec le nombre de documents associés pour l'apprentissage et le test de la base Reuters R8

2. **Reuter(R52)** L'ensemble Reuters-21578 R52 est un ensemble ds document contient 9100 documents concernant 52 catégories de sujets (acq, earn, etc ...), contient 6532 document pour l'apprentissage et 2568 pour le testes.

Catégorie	Document d'apprentissage	Document de teste	Total
acq	1596	696	2292
alum	31	19	50
bop	22	9	31
carcass	6	5	11
cocoa	46	15	61
coffee	90	22	112
copper	31	13	44
cotton	15	9	24
cpi	54	17	71
cpu	3	1	4
crude	253	121	374
dlr	3	3	6
earn	2840	1083	3923
fuel	4	7	11
gas	10	8	18
gnp	58	15	73
gold	70	20	90
grain	41	10	51
heat	6	4	10
housing	15	2	17
income	7	4	11

Catégorie	Document d'apprentissage	Document de teste	Total
instal-debt	5	1	6
interest	190	81	271
ipi	33	11	44
iron-steel	26	12	38
jet	2	1	3
jobs	37	12	49
lead	4	4	8
lei	11	3	14
livestock	13	5	18
lumber	7	4	11
meal-feed	6	1	7
money-fx	206	87	293
money-supply	123	28	151
nat-gas	24	12	36
nickel	3	1	4
orange	13	9	22
pet-chem	13	6	19
platinum	1	2	3
potato	2	3	5
reserves	37	12	49
retail	19	1	20
rubber	31	9	40
ship	108	36	144
strategic-metal	9	6	15
sugar	97	25	122
tea	2	3	5
tin	17	10	27
trade	251	75	326
veg-oil	19	11	30
wpi	14	9	23
zinc	8	5	13

TABLE 4.2: Répartition des documents par catégorie, avec le nombre de documents associés pour l'apprentissage et le test de la base Reuters R52

4.4.0.2 20 Newsgroups

Cet ensemble de données est une collection de documents de newsgroup, il est devenue un ensemble de données populaires pour les expériences dans les

applications de texte des techniques d'apprentissage automatique, telles que la classification de texte et le regroupement de texte. Il contient 18821 document regroupé en 20 classes (alt.atheism,comp.graphics,comp.os.ms-windows.misc,...etc) , il contient 11293 documents pour l'apprentissage et 7528 documents pour le testes.

Catégorie	Document d'apprentissage	Document de teste	Total
alt.atheism	480	319	799
comp.graphics	584	389	973
comp.os.ms-windows.misc	572	394	966
comp.sys.mac.hardware	578	385	963
comp.windows.x	593	392	985
misc.forsale	585	390	975
rec.autos	594	395	989
rec.motorcycles	598	398	996
rec.sport.baseball	597	397	994
rec.sport.hockey	600	399	999
sci.crypt	595	396	991
sci.electronics	591	393	984
sci.med	594	396	990
sci.space	593	394	987
soc.religion.christian	598	398	996
talk.politics.guns	545	364	909
talk.politics.mideast	564	376	940
talk.politics.misc	465	310	775
talk.religion.misc	377	251	628

TABLE 4.3: Répartition des documents par catégorie, avec le nombre de documents associés pour l'apprentissage et le test de la base 20 Newsgroups

4.4.0.3 WebKB

Les documents du WebKB sont des pages Web collectées par le projet World Wide Knowledge Base (Web-> Kb) du groupe d'apprentissage de texte CMU, et ont été téléchargés à partir de la page d'accueil The 4 Universities Data Set(Cornell, Texas, Washington et Wisconsin) en Janvier 1997. il contient 4199 documents regroupé en 04 catégories : student, faculty, course, project, 2803 documents pour l'apprentissage et 1396 documents pour le

testes.

Catégorie	Document d'apprentissage	Document de teste	Total
project	336	168	504
course	620	310	930
faculty	750	374	1124
student	1097	544	1641

TABLE 4.4: Répartition des documents par catégorie, avec le nombre de documents associés pour l'apprentissage et le test de la base WebKB

- Le tableau suivant montrant des informations générales sur les différentes bases que nous avons utilisées dans notre expérience.

Les bases	Base d'apprentissage	Base de teste	Total
Reuters R8	5485	2189	7674
Reuters R52	6532	2568	9100
20Newsgroups	11293	7528	18821
WebKB	2803	1396	4199

TABLE 4.5: Description des bases d'apprentissage et de tests des différents corpus utilisés dans nos expérimentations

4.5 Expérimentation et la discussion des résultats

1. **Prétraitements :**
   1-Nous avons commencer notre prétraitement avec la tokenizarion :
   Premièrement, nous avons appliqué la tokenization sur notre document(texte) où nous avons appelé notre méthode en utilisant l'outil ntlk.
   On a X représente la liste qui reçoit les documents après la tokenization et all[i] c'est la matrice contient tous les documents.

```
X.append(nltk.word_tokenize(all[i][1]))
```

   2-Supprimer les stops word exister dans le texte.
   importer la méthode de stop words à partir de l'outil nltk.

```
from nltk.corpus import stopwords
```

```
stop_word = []
stopset = set(stopwords.words('english'))
for w in doc:
    for word in w:
        if not word in stopset:
            stop_word.append(word)
```

Où stopset contient la liste des tous les stopwords existe dans la langue anglais puis nous avons parcouru notre document et supprimer les stopwords.
3-Appliquer la lemmatisation sur notre texte après l'éliminations des stopwords.
Où nous appelons d'abord la méthode en utilisant l'outil nltk.

```
from nltk.stem.wordnet import WordNetLemmatizer

lemmatiser = WordNetLemmatizer()
lemmatisation=[]
for lem in stop_word:
    lemmatisation.append(lemmatiser.lemmatize(lem))
```

5- finalement, un stemming est appliqué sur le document.

```
from nltk.stem import PorterStemmer

stemming=[]
stemmer = PorterStemmer()
for stem in lemmatisation:
    stemming.append(stemmer.stem(stem))
```

-Construire la liste du vocabulaire(dictionnaire) de tous les termes distincts qui apparaissent dans tous le corpus d'apprentissage.

2. **codage des textes**
- Dans nos expériences, nous avons utilisé deux types de codage pour les corpus où nous donnons pour chaque mot dans le dictionnaire un numéro unique :

(1) Le premier type de codage : Le premier type est le codage avec word2vector où nous avons codé chaque mot dans le dictionnaire en cinq numéro de type float.
Les paramètres principaux à régler le modèle de word2vector :
- Phrases : textes d'entrées.
- Taille : Spécifie la taille du vecteur qui représentera chaque mot, par défaut=100.
- Min-count : Nombre minimal d'apparitions d'un mot à prendre en compte pour l'apprentissage du modèle. On donne 1 pour mettre tous les mots pris en compte, par défaut=100.

```
print(" modele de word2vector")
model = Word2Vec(doc, min_count=1, size=5)
```

FIGURE 4.1: Modèle de Word2vector

(2) Le deuxième type : Le deuxième type est le codage en entier où nous avons utilisé des nombres entier pour chaque mot dans le dictionnaire. Le but d'utilisation de deux codages est de faire une comparaison pour savoir ce qui est mieux et efficace pour l'utilisé dans la classification.

-On applique les mêmes codages pour la corpus de teste.
-pour les classes, on créions un fichier contient toutes les classes de corpus puis nous donnons un indice pour chaque catégorie.
-Enfin, les textes ont été nettoyés, coder et prêts à être utilisés dans le modèle CNN.

3. **Architecture du réseau**
Au cours de nos expérimentations, nous avons créé un modèle de cnn avec trois couches convolutives et un couche de fully connected.
Le modèle de CNN :
Nous utilisons le modèle suivant dans notre expérience :
Notre texte en entrée est de taille varier car nous avons utilisé des différentes bases de texte par exemple : si nous prenons la base reuters r8 avec une taille 1500 pour le fichier train et 1000 pour le fichier teste, la taille de dictionnaire sera 7396*5 alors la taille de texte en entrée du réseau sera 7396*5 pour les deux fichiers train et teste, le texte passe d'abord à la première couche de convolution. Cette couche est composée de 100 filtres de taille 3*3 chacune de nos couches de convolution est suivie d'une fonction d'activation ReLU, cette fonction force les neurones à retourner les valeurs positives, après cette convolution 100 features maps de taille 100*100 seront créés, ensuite on applique

Maxpooling sur les 100 feature maps qui sont obtenus pour réduire la taille de texte ainsi la quantité.
À la sortie de cette couche, nous aurons 100 feature maps de taille moitié de la taille d'origine. On répétons ce procédure avec les couches de convolutions deux et trois .
La fonction d'activation ReLU est appliquée toujours sur chaque convolution. Un Global Maxpooling est appliquée après la troisième couche de convolution.
À la sortie de cette couche, nous aurons un vecteur de caractéristiques avec un dimension égale à 100.
Après ces trois couches de convolution, nous utilisons un réseau de neurone composé de deux couches de fully connected, le premier couche est composé de 250 neurones où la fonction d'activation utilisée est le ReLU et la deuxième couche est une softmax qui permet de calculer la distribution de probabilité des 8 classes (nombre de classe dans la base reuters r8).

```
Build model...
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_1 (Conv1D)            (None, 7396, 100)         1600
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 3698, 100)         0
_________________________________________________________________
dropout_1 (Dropout)          (None, 3698, 100)         0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 3698, 100)         40100
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 1849, 100)         0
_________________________________________________________________
dropout_2 (Dropout)          (None, 1849, 100)         0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 1849, 100)         50100
_________________________________________________________________
global_max_pooling1d_1 (Glob (None, 100)               0
_________________________________________________________________
dense_1 (Dense)              (None, 250)               25250
_________________________________________________________________
dropout_3 (Dropout)          (None, 250)               0
_________________________________________________________________
activation_1 (Activation)    (None, 250)               0
_________________________________________________________________
dense_2 (Dense)              (None, 8)                 2008
_________________________________________________________________
activation_2 (Activation)    (None, 8)                 0
=================================================================
Total params: 119,058
Trainable params: 119,058
```

FIGURE 4.2: Les paramètres de CNN utilisent dans notre expérimentale

4.5.0.1 Résultats expérimentaux

Nous avons testé notre modèle sur quatre bases, pour chaque base nous teston la méthode avec 3 itération d'apprentissage (5,10,15) et avec différent taille d'apprentissage et de teste.

1. **Nombre d'itération=5**

TABLE 4.6: Résultat d'apprentissage de CNN avec un itération=5 sur les 4 bases

Taille de corpus	1000/500	1500/1000	3000/2500
R8	word2vec :76,9 % temps=2s Int : 69,4% temps=2s	word2vec :94,5 % temps=2s int= : 87,5% temps=s2	word2vec : 95% temps=4s Int : 87,5% temps=4s
R52	word2vec : 96,8% temps=1s int= : 94,3% temps=1s	word2vec : 96,8% temps=1s int= : 94,3% temps=1s	word2vec :98,7% temps=5s Int : 95,7% temps=5s
20 Newsgroups	word2vec : 92,4% temps=2s int= : 85% temps=2s	word2vec : 92,4% temps=2s int= : 85% temps=2s	avec :taille 2000/800 word2vec :94,9 % temps=2s Int : 92,3% temps=2s
WebKb	word2vec : 75% temps=1s int= : 75% temps=1s	word2vec : 75% temps=1s int= : 75% temps=1s	avec :taille 2800/1390 word2vec :76,9 % temps=2s Int : 69,4% temps=2s

En analysant le tableau , nous trouvons que, pour :

— **la base Reuters-R8 :** Nous obtenons une bonne précision si nous utilisons w2v et aussi si nous augmentons la taille d'apprentissage et la taille de test où nous notons que le taux de reconnaissance à la taille de 1000/500 est égal à 76,9% pour le codage de word2vector et 69,4 pour codage int=5 et pour un taille du d'apprentissage égale = 3000 et 2500 pour le teste nous obtenons un taux 94,5% et 95 % pour le codage word2vector, on remarque que pour le temps d'apprentissage

est augmente si la taille des corpus augmente.

— **la base Reuters-R52 :** le taux de reconnaissance est augmenté si la taille de corpus est grand nous obtenons les résultats (96.8%,96.8%,98.7%) pour les tailles(1000,1500,3000 pour l'apprentissage et 500,1000,2500 successivement pour le test) , on remarque que codage int=5 par rapport le codage word2vec est pas bien pour l'apprentissage de CNN.

— **la base 20 Newsgroups :** Nous obtenons 92,4% pour le codage word2vec et 85% pour le codage int=5 avec les taille 1000/500,1500/1000 avec un temps=2s mais avec la taille 2000/800 nous obtenons 94,9% pour word2vec et 92,3% pour int=5.

— **la base Webkb** : nous obtenons un taux de reconnaissance moins si nous utilisons le codage de int=5 par rapport word2vector , pour cette base nous obtenant les taux suivant :74.9,74.9,78.6 pour le codage word2vec avec un temps d'apprentissage égale 1s et 2s , tous ça avec les tailles 1000/500,1500/1000,3000/2500 .

2. **Nombre d'itération=10**

TABLE 4.7: Résultat d'apprentissage de CNN avec un itération=10 sur les 4 bases

Taille de corpus	1000/500	1500/1000	3000/2510
R8	word2vec :92,8% temps=1s Int : 83,4% temps=1s	word2vec :96,3 % temps=2s int= : 87,5% temps=2s	word2vec : 96,5% temps=4s Int : 75,8% temps=4s
R52	word2vec : 96,9% temps=1s int= : 96,2% temps=1s	word2vec : 98,9% temps=2s int= : 98% temps=2s	word2vec :98% temps=5s Int : 98% temps=5s
20 Newsgroups	word2vec : 96,9% temps=2s int= : 92,9% temps=4s	word2vec : 94,9% temps=4s int= : 92,9% temps=4s	avec taille =2000/800 word2vec :94,9 % temps=2s Int : 94,9% temps=2s
WebKb	word2vec : 74,9% temps=2s int= : 74,9% temps=1s	word2vec : 74,9% temps=1s int= : 69,3% temps=1s	avec :taille 2800/1390 word2vec :80,7% temps=2s Int : 74,9% temps=2s

En analysant le tableau , nous trouvons que, pour :

— **la base Reuters-R8 :** Nous obtenons une bon précision si nous utilisons word2vec et si nous augmentons la taille d'apprentissage et la taille de test où nous notons que le taux de reconnaissance à la taille de 1000/500 est égal à 92.8% pour le codage de word2vector et 83.4% pour codage int=5 et pour un taille du d'apprentissage égale = 3000 et 2500 pour le teste nous obtenons un taux égale 96,5% pour le codage word2vector et et 96.3 % pour codage int=5.

— **la base Reuters-R52 :** le taux de reconnaissance est augmenté si la taille de corpus est grand nous obtenons les résultats (96.9%,98.9%,98%) pour les tailles(1000,1500,3000 pour l'apprentissage et 500,1000,2500 successivement pour le test) si nous utilisons codage word2vec ainsi nous obtenons les taux 96.2%,98%,98% pour le codage int=5 .

— **la base 20 Newsgroups :** Nous obtenons 96,9% et 94.9% pour le codage word2vec et 92.9 % pour le codage int=5 avec les taille

1000/500,1500/1000 et avec un temps=4s mais avec la taille 2000/800 nous obtenons 94,9% pour word2vec et 94,9% pour int=5 avec un temps=2s

— **la base Webkb** :nous obtenons un précision égale 74,9% pour codage word2vec pour les taille 1000/500 et 1500/100 et 74.9%,69.3% pour le codage int=5 , le taux de reconnaissance pour la taille 80,7% pour word2vec et 74.9% pour int=5.

3. **Nombre d'itération=15**

TABLE 4.8: Résultat d'apprentissage de CNN avec un itération=15 sur les 4 bases

Taille de corpus	1000/500	1500/1000	3000/2500
R8	word2vec :89,6% temps=1s Int : 89,6% temps=1s	word2vec :96,1 % temps=2s int= : 87,5% temps=2s	word2vec : 95,6% temps=4s Int : 85,5% temps=4s
R52	word2vec : 98,8% temps=1s int= : 96,2% temps=1s	word2vec : 99% temps=2s int= : 98% temps=2s	word2vec :98,7% temps=5s Int : 92,3% temps=5s
20 Newsgroups	word2vec : 98% temps=2s int= : 90% temps=2s	word2vec : 95% temps=4s int= : 93,8% temps=4s	avec :taille 2000/800 word2vec :89.9 % temps=2s Int :90,1% temps=2s
WebKb	word2vec : 74,9% temps=1s int= : 74,9% temps=1s	word2vec : 74,9% temps=1s int= : 69,3% temps=1s	avec :taille 2800/1390 word2vec :78,6% temps=2s Int : 74,9% temps=2s

En termine l'analyse de tableau , nous trouvons que, pour :

— **la base Reuters-R8 :** Nous obtenons les taux de reconnaissance suivant : 89.6 % ,96.1% ,95.6 % pour le codage word2vec pour les différent taille des corpus 1000/500, 1500/1000 , 3000/2500 et 89.6%,87.5%,85.5% pour le codage int=5.
— **la base Reuters-R52 :**nous obtenons les résultats (98.8%,99%,98.7%)

pour les tailles(1000,1500,3000) pour l'apprentissage et 500,1000,2500 successivement pour le test) si nous utilisons codage word2vec ainsi nous obtenons les taux 96.2%,98%,92.3% pour le codage int=5 .

— **la base 20 Newsgroups :** Nous obtenons 98% et 95%,89.9% pour le codage word2vec et 90% , 93.8%,90.1% pour le codage int=5 avec les taille 1000/500,1500/1000,2000/800

— **la base Webkb** :nous obtenons un taux égale 74,9% pour codage word2vec pour les taille 1000/500 et 1500/100 et 74.9%,69.3% pour le codage int=5 , le taux de reconnaissance pour la taille 3000/2500 est égale 78.6% pour word2vec et 74.9% pour int=5.

D'après ces dernier tableau , nous pouvons conclure que codage de word2vector est mieux que les autres codage ainsi ma taille des corpus a un rôle pour obtenir un bon taux de reconnaissance tel-que si la taille est plus grande, plus la précision est obtenu est grande , et finalement on constate que le nombre des itération pas une mesure pour obtenir le taux de reconnaissance le plus élevé.

4.5.0.2 Comparaison entre la méthode CNN et les autres méthodes d'apprentissage automatique

Nous proposons une série de comparaisons entre la méthode CNN et les autres méthodes d'apprentissage automatique (CNN,MLP,LSTM,SVM,KNN) : Ce tableau reflète les résultats obtenus pour une classification des Reuters-21578 avec 46 classes.

Les méthodes	CNN	MLP	LSTM	KNN
Taux de reconnaissance	97,8	76,6	97,8	79
Temps	223 S	1 S	435 S	139 S

TABLE 4.9: Comparaison entre résultat de CNN et les autres méthodes

le taux de reconnaissance obtenus pour la méthode CNN est 97,8 % et le temps nécessaire pour l'exécution de cette approche est 223S.
Dans le cas de Mlp : le taux de reconnaissance est 76.6 tandis que le temps nécessaire pour l'exécution est 1S .
Avec LSTM : le taux de reconnaissance obtenus est égale le taux obtenus par la méthode CNN mais le temps de nécessaire pour l'exécution est différent nous obtenions 435S pour le LSTM.

Avec KNN : le taux de reconnaissance est égale 79 % et le temps nécessaire pour l'exécution est égale à 139S. Permis ces résultat nous constatons que la méthode CNN est mieux que les autres méthodes soit pour le taux de reconnaissance soit pour le temps d'exécution.

4.5.0.3 Discussion des résultats

Les résultats obtenus dans les différentes expérimentations montrent l'efficacité de notre proposition et la possibilité de sont utilisation dans ce domaine. En effet, dans le cadre d'analyse de texte, notre proposition permet de reconnaitre correctement plus de 92 % des documents ce qui permet d'accélérer considérablement le processus d'analyse des texte et d'éviter les erreurs de l'analyse par d'autre méthode.

4.6 Conclusion

Dans ce chapitre, Nous avons représenté l'implémentation de notre système : L'environnement et les outils de développement, ensuite nous avons présenté quelques expérimentations appliquées sur des différentes bases puis on exposé les résultats obtenus en terme de taux de reconnaissance et de temps nécessaire pour l'exécution .

CONCLUSION GÉNÉRALE

Conclusion générale

L'analyse de textes s'est avérée au cours des dernières années comme un domaine majeur de recherche pour les entreprises comme pour les particuliers. Catégorisation une méthode d'analyse peut améliorer la précision du texte dans l'analyse de texte, mais on recherche d'améliorer la vitesse de l'analyse. Les applications d'apprentissage en profondeur nous aident non seulement à augmenter la vitesse, mais aussi à améliorer la qualité de l'analyse avec le développement de la technologie de l'information et du réseau.
L'objectif de ce travail était de faire une catégorisation des textes non structurer, nous avons essayé de représenté ces texte dans un format structuré, puis transformer ces textes comme des entrées pour notre réseau de convolution. Nous avons vu en premier temps les réseaux de neurone et les réseaux de neurone profond.
Ensuite, nous avons présenté le domaine d'analyse des textes et ses différent spécification(processus, techniques,domaine d'application,...etc). Pour présenter ce système, nous avons besoin d'une méthode de classification, la méthode la plus efficace et la plus utilisée est la méthode CNN que nous avons déjà expliqué dans le premier chapitre dans la section des réseaux profond.
Dans la phase de conception, nous avons proposé une architecture pour implémenter ce système, et nous essayons de donné le fonctionnement du système et son conception général et détaillée. Le dernier chapitre donne une vue de fond de notre système, à travers les outils de programmation.
Nous avons rencontré quelques problèmes dans la phase d'implémentation, l'utilisation d'un CPU a fait que le temps d'exécution était trop couteux. Afin de régler ce problème on doit utiliser des réseaux de neurones à convolution plus profonds déployé sur un GPU au lieu d'un CPU sur des bases plus importantes

BIBLIOGRAPHIE

[1] Adeline Abbé. *Analyse de données textuelles d'un forum médical pour évaluer le ressenti exprimé par les internautes au sujet des antidépresseurs et des anxyolitiques*. PhD thesis, Paris Saclay, 2016.

[2] Lynda Amimer. *Modélisation et Commande des Systèmes Non Linéaires Fractionnaires par des Réseaux de Neurones Fractionnaires*. PhD thesis, Université Mouloud Mammeri, 2015.

[3] Mohamed Yessin Ammar. *Mise en œuvre de réseaux de neurones pour la modélisation de cinétiques réactionnelles en vue de la transposition batch/continu*. PhD thesis, 2007.

[4] Fatiha Barigou. *Contribution à la catégorisation textes et à de l'extraction d'information*. PhD thesis, Université Oran1 ahmed ben bella, 2013.

[5] Youcef Bendaoud. Prédiction des résistances mécaniques des bétons à base des ciments composés en utilisant les réseaux neurones artificiels. 2014.

[6] Ameni Bouaziz. *Catègorisation automatique de news à l'aide de techniques d'apprentissage supervisè*. PhD thesis, Universitè nice sophia antipolis.

[7] Arnaud Buhot. *Etude de propriétés d'apprentissage supervisé et non supervisé par des méthodes de Physique Statistique*. PhD thesis, Université Joseph-Fourier-Grenoble I, 1999.

[8] Maria Fernanda Caropreso, Maria Fernandacnandad, Stan Matwin, and Fabrizio Sebastiani. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization, 2001.

[9] Hicham Chaoui. *Conception et comparaison de lois de commande adaptative à base de réseaux de neurones pour une articulation flexible avec non-linéarité dure.* PhD thesis, Université du Québec à Trois-Rivières, 2002.

[10] Hicham Chaoui. *Conception et comparaison de lois de commande adaptative à base de réseaux de neurones pour une articulation flexible avec non-linéarité dure.* PhD thesis, Université du Québec à Trois-Rivières, 2002.

[11] Thierry Charnois. *Accs à l'information : vers une hybridation fouille de données et traitement automatique des langues.* Habilitation à diriger des recherches, Université de Caen, December 2011.

[12] Yahui Chen. Convolutional neural network for sentence classification. Master's thesis, University of Waterloo, 2015.

[13] Hacène Cherfi. *Etude et réalisation d'un système d'extraction de connaissances à partir de textes.* PhD thesis, Université Henri Poincaré-Nancy I, 2004.

[14] Shilpa Dang and Peerzada Hamid Ahmad. Text mining : Techniques and its application. *IJETI International Journal of Engineering & Technology Innovations*, 1(4) :2348–0866, 2014.

[15] Li Deng, Dong Yu, et al. Deep learning : methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4) :197–387, 2014.

[16] Hércules Antonio Do Prado. *Emerging Technologies of Text Mining : Techniques and Applications : Techniques and Applications.* 2007.

[17] Grzegorz Dziczkowski. *Sentiment analysis : an autonomous system exploring opinions expressed in cinema reviews.* Theses, École Nationale Supérieure des Mines de Paris, December 2008.

[18] Ronen Feldman and James Sanger. *The text mining handbook : advanced approaches in analyzing unstructured data.* Cambridge university press, 2007.

[19] George Forman. Feature selection for text classification. *Computational methods of feature selection*, 1944355797, 2007.

[20] Quentin Fresnel and Geoffroy Peeters. *Apprentissage de descripteurs audio par Deep learning, application pour la classification en genre musical.* PhD thesis, Univserité Paris 6, 2015.

[21] Sébatien Frizzi, Rabeb Kaabi, Moez Bouchouicha, Jean-Marc Ginoux, Farhat Fnaiech, and Eric Moreau. Détection de la fumée et du feu par réseau de neurones convolutifs. In *Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle*, Caen, France, July 2017.

[22] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[23] Philippe Hamel. Apprentissage de représentations musicales à l'aide d'architectures profondes et multiéchelles. 2012.

[24] Jean-Claude Heudin. *Comprendre le DEEP LEARNING*.

[25] Geoffrey Hinton and Ruslan Salakhutdinov. Discovering binary codes for documents by learning deep generative models. *Topics in Cognitive Science*, 3(1) :74–91, 2011.

[26] Tellier Isabelle. Introduction à la fouille de textes. 10 nov 2017.

[27] Radwan Jalam. Apprentissage automatique et catégorisation de textes multilingues. *PhD Tesis, Université Lumiere Lyon*, 2, 2003.

[28] Killian Janod, Mohamed Morchid, Richard Dufour, and Georges Linares. Apport de l'information temporelle des contextes pour la représentation vectorielle continue des mots. *22ème Traitement Automatique des Langues Naturelles*, 2015.

[29] Shaidah Jusoh and Hejab M Alfawareh. Techniques, applications and challenging issue in text mining ues, applications and challenging issue in text mining ues, applications and challenging issue in text mining. *IJCSI International Journal of Computer Science Issues*, 9(6) :431–436, 2012.

[30] Mohammed KOUDRI. Modèle de mélange gaussien. application sur image cytologique. 2011.

[31] Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S Gerber, and Laura E Barnes. Hdltex : Hierarchical deep learning for text classification. *arXiv preprint arXiv :1709.08267*, 2017.

[32] Yann LeCun. Les enjeux de la recherche en intelligence artificielle. *Interstices*, 2016.

[33] Yann LeCun, Sumit Chopra, Marc'Aurelio Ranzato, and Fu-Jie Huang. Energy-based models in document recognition and computer vision. In *Proc. International Conference on Document Analysis and Recognition (ICDAR)*, 2007. (keynote address).

[34] Adam Luotonen and Erik Jalsborn. Text analysis-exploring latent semantic models for information retrieval, topic modeling and sentiment detection. 2011.

[35] NASSIM MAMI MOHAMMED. *Extraction des connaissances dans un environnement distribué : synthèse.* PhD thesis, 2014.

[36] S Marcos, P Roussel-Ragot, L Personnaz, O Nerrand, G Dreyfus, and C Vignat. Réseaux de neurones pour le filtrage non linéaire adaptatif. *Traitement du Signal*, 8 :409–422, 1993.

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

[38] Juffi McCallum. The 4 universities data set. http ://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/. Accessed : 2018-06-17.

[39] Danny Miller and Peter H. Friesen. Porter's (1980) generic strategies and performance : An empirical examination with american data : Part ii : Performance implications. *Organization Studies*, 7(3) :255–261, 1986.

[40] Mohammed Zakaria Mokri. *Classification des images avec les réseaux de neurones convolutionnels.* PhD thesis, 09-01-2018.

[41] Antonio Moreno and Teófilo Redondo. Text analytics : The convergence of big data and artificial intelligence. *IJIMAI*, 3(6) :57–64, 2016.

[42] MEADI MOHAMED NADJIB. Technique basée hits/svm pour la réduction et la pondération des caractéristiques des pages web. *THÉSE de Docteur en Sciences.*

[43] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1) :1, 2015.

[44] Josh Patterson and Adam Gibson. *Deep Learning : A Practitioner's Approach.* " O'Reilly Media, Inc.", 2017.

[45] Filipo Studzinski Perotto. *Un Mécanisme Constructiviste d'Apprentissage Automatique d'Anticipations pour des Agents Artificiels Situés.* PhD thesis, Institut National Polytechnique de Toulouse-INPT, 2010.

[46] Isabelle Rivals. *Modélisation et commande de processus par réseaux de neurones ; application au pilotage d'un véhicule autonome.* PhD thesis, Université Pierre et Marie Curie-Paris VI, 1995.

[47] Hans Schmid. Multi-ferroic magnetoelectrics. *Ferroelectrics*, 162(1) :317–338, 1994.

[48] Chirag Shah, Bhoopesh Choudhary, and Pushpak Bhattacharyya. Constructing better document vectors using universal networking language (unl). In *apresentada em International Conference on Knowledge-Based Computer Systems (KBCS)*, pages 1–10, 2002.

[49] Ramzan Talib, Muhammad Kashif Hanif, Shaeela Ayesha, and Fakeeha Fatima. Text mining : Techniques, applications and issues. *International Journal of Advanced Computer Science & Applications*, 1(7) :414–418, 2016.

[50] Yannick Toussaint. *Text Mining : Symbolic methods to build ontologies and to semantically annotate texts*. Habilitation à diriger des recherches, Université Henri Poincaré - Nancy I, November 2011.

[51] Claude Touzet. *LES RESEAUX DE NEURONES ARTIFICIELS, INTRODUCTION AU CONNEXIONNISME*. Collection de l'EERIE. EC2, 1992.

[52] BARRIERE Valentin. *APPROCHES « DEEP LEARNING » APPLIQUEES AUX SIGNAUX AUDIO : PAROLE ET MUSIQUE*. PhD thesis, Institut de Recherche en Informatique de Toulouse (IRIT), 2015.

[53] Suzanne van den Bosch. Automatic feature generation and selection in predictive analytics solutions.

[54] S Vijayarani and R Janani. Text mining : open source tokenization tools–an analysis. *Advanced Computational Intelligence*, 3(1) :37–47, 2016.

[55] Pascal Vincent. Modèles à noyaux à structure locale. 2004.

[56] Henning Wachsmuth. Text analysis pipelines. In *Text Analysis Pipelines*, pages 19–53. Springer, 2015.

[57] Haohan Wang, Bhiksha Raj, and Eric P. Xing. On the origin of deep learning. *CoRR*, abs/1702.07800, 2017.

[58] Sholom M Weiss, Nitin Indurkhya, Tong Zhang, and Fred Damerau. *Text mining : predictive methods for analyzing unstructured information*. Springer Science & Business Media, 2010.

Dédicace

À cœur vaillant rien impossible

Je dédie cette thèse à . . .

Mes parents

À mes sœurs

À tous mes frères

À celles qui me sont chers : Mohamed Amine et Riad.

À Tous Mes enseignants tout au long de mes études.

À tous ma famille

À tous ceux qui ont participé de près ou de loin à la réalisation de ce travail.

Remerciement

Aucune œuvre humaine ne peut se réaliser sans l'aide de Dieu. Je le remercie en premier lieu de m'avoir donné la santé, le courage ainsi qu'une grande volonté pour aboutir à ce travail.

Entreprendre une mémoire de fin d'études en informatique après 18 ans de rupture avec l'univers d'études et de recherche, c'était plus qu'un défi pour moi, ce mémoire est le fruit de longues heures de lecture, de recherches, de réflexion et le résultat d'un effort constant, cet effort n'aurait pu aboutir sans la contribution d'un nombre de personnes que je tiens à remercier.

Tout d'abord, j'exprime ma gratitude à mon encadreur «Mr. Meadi Mohamed Nadjib » pour ses conseils et orientations en dépit d'un emploi du temps chargé, pour tous les efforts qu'il fournis durant cette année .

Mes vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont portés à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions

Je remercie chaleureusement ma famille ,tous mes amis et collègues qui m'ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire, à tous ceux qui m'ont était une source d'aide ou de motivation importante, en particulier je cite parmi eux, Boukezzoula Mohamed Amine, Riad Belkhiri, Sana agti , Ben Amar Nour Elhouda, Sid Zineb , Aicha Regig, Selmi Aymen takie eddine, Labed Ayoub, Djoudi Asma, . . . etc.

Enfin, ce travail de recherche n'aurait jamais été terminé sans le soutien de plusieurs personnes qui n'ont pas hésité à me donner le courage et le dynamisme pour l'accomplir. Qu'ils trouvent ici l'expression de mes sincères remerciements. Merci à tous et à toutes

RÉSUMÉ

La révolution d'information est arrivé par le développement à grande échelle des accès réseaux Internet qui fait exploser la quantité d'informations textuelles importante dans l'activité quotidienne. Des chercheurs et des entreprises ainsi que les utilisateurs ont besoin d'accès intelligents aux immenses bases de données textuelles et leurs manipulations qui augmente très largement.

les chercheurs en intelligence artificielle cherche à développer des programmes capables de reconnaître des différentes tâches concernant l'analyse des textes, ils posent un module d'apprentissage qui permet d'adapter leur comportement à des situations jamais rencontrées, l'un des algorithmes d'apprentissage automatique les plus utilisées est l'apprentissage profond.

Ce travail vise à étudier le domaine d'analyse de texte et plus particulièrement la classification des textes dans le cas où nous utilisons un approche de l'apprentissage automatique qu'était différent des autres approches, nous visons d'analyser des textes avec les réseaux de neurones à convolution.

Mots clés : Analyse des textes, classifications des textes, Apprentissage automatique, Apprentissage profond, les réseaux de neurone à convolution.

ملخص

لقد أحدثت ثورة المعلومات تطورا واسعا في كمية المعلومات النصية. والتي أصبحت ذات أهمية متزايدة في الأنشطة اليومية.
يحتاج الباحثون والشركات وكذلك مستخدمي الانترنت الى وصول ذكي لقواعد البيانات النصية الضخمة التى نمت على نطاق واسع لغاية العمل بها.

يعمل الباحثون في مجال الذكاء الاصطناعي على تطوير برامج قادرة في التعرف على المهام المختلفة المتعلقة بتحليل النصوص ، ويطلب منهم نموذج تعلم يسمح بتكييف سلوكهم مع حالات لم يتم مواجهتها ، أحد الخوارزميات الأكثر استخداما في التعلم الآلي : التعلم العميق.

يهدف هذا البحث إلى دراسة مجال تحليل النصوص وتحديداً تصنيف النصوص في الحالة التي نستخدم فيها نموذجا مختلفاًا عن النماذج الاخرى ، هدفنا هو تحليل النصوص باستخدام التعلم العميق وبصورة أدق شبكات الخلايا العصبية التلافيفية.
الكلمات المفتاحية: تحليل النص ، تصنيفات النص ، التعلم الآلي ، التعلم العميق ، شبكات الخلايا العصبية التلافيفية.

ABSTRACT

he information revolution has come through the large-scale development of Internet network access that is exploding the amount of textual information important in everyday business. Researchers and companies as well as users need intelligent access to huge textual databases and their manipulations, which is increasing dramatically.

researchers in artificial intelligence seeks to develop programs able to recognize different tasks concerning the analysis of texts, they pose a learning module that allows to adapt their behavior to situations never encountered, one of the algorithms of most used machine learning is deep learning.

This work aims to study the field of text analysis and more particularly the classification of texts in the case where we use a machine learning approach that was different from the other approaches, we aim to analyze texts with the networks of convolutional neurons.

Keywords: Text analysis, text classifications, Machine learning, Deep learning, convolutional neuron networks.