République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed El Bachir El Ibrahimi
Bordj Bou Arreridj
Faculté des Mathématiques et de l'Informatique
Département d'Informatique

Mémoire de fin d'étude

En vue de l'obtention du Diplôme de Master en Informatique

Option : Réseaux et multimédia

Thème

Deep Learning For Text Catégorization

Présenté par Mr :
ZEGADI Walid

Encadré par Mr :
BELAZZOUG Mouhoub

Soutenu publiquement le / / devant le jury:

Président	: BENSEFIA Hassina	M.C.B	UMBI.BBA
Examinateur	: BELALTA Ramla	M.C.B	UMBI.BBA
Encadreur	: BELAZZOUG Mouhoub	M.C.B	UMBI.BBA

Année universitaire : 2019 -2020

DEDICACES

Je dédie ce modeste travail tous d'abord
à mes très chers parents, pour
leur sens du devoir et soutien moral
pendent toutes mes années d'études.
J'espère qu'ils trouveront dans ce travail
toute ma connaissance et tout mon amour.
A ma chère épouse.
A mes enfants ISHAK et TAKOUA
Á mes frères et à ma chère sœur.
Á mon grand père et ma grande mère.
Á toute ma grande famille de loin ou de proche.
Á tous mes meilleurs amis
Á tous les gens qui m'ont aidé pour faire ce travail .
Et a tous ceux qui j'ai oublié de mentionner !!!

Z.WALID

REMERCIEMENTS

Je remercie et je loue, d'abord notre ***Dieu*** *de m'avoir aidé à atteindre ce but, et de défier tous les obstacles afin de compléter ce modeste mémoire, sans* ***Dieu****, rien ne se fait, rien ne se crée.*

C'est avec un immense plaisir que j'exprime mes remerciements à mon encadreur ***Dr***. ***BELAZZOUG MOUHOUB*** *qui a toujour suivi ce travail avec intérêt et qui l'a témoigné par des encouragements de toutes sortes.*

Je remercie très vivement le président et les membres de jury qui ont accepté de juger ce travail.

Nous tiens à remercier ***Dr. Mohammed Brahimi*** *de m' avoir aidés à faire ce travail.*

En fin pour tous ceux qui, de prés ou de loin ont rendu ce travail possible.

Résumé

Avec l'accroissement constant du nombre de textes numériques disponibles, de nombreuses recherches sont menées afin d'organiser et de rendre exploitable cette immense base d'informations. Ces données représentent une source d'informations importante pour plusieurs applications telles que les systèmes de recommandation, la détection des communautés, le marketing et la vision par ordinateur. Dans ce contexte, la catégorisation de textes a pour objectif de regrouper les textes « thématiquement proches » au sein d'un même catégorie.

La majorité de ces approches abordent généralement cette problématique comme un tout. Cependant la catégorisation de textes est un problème double : le premier problème correspond à la représentation textuelle ou, en d'autres termes, comment obtenir une représentation mathématique et numérique d'un texte, ainsi la sélection des meilleur caractéristiques (terme pertinent) qui assure une meilleure classification ; la seconde problématique se situe, principalement dans le domaine de l'apprentissage. Il s'agit d'utiliser l'une des technique et principes les plus récente d'apprentissage profond, à partir d'un jeu d'entraînement, permettant de catégoriser tout nouveau texte.

Dans cette thèse, nous appliquons l'un des model performant d'apprentissage profond qui est le réseau de neurone convolutif (CNN) sur un jeu de donné textuel pour résoudre les problèmes de classification des textes, en plus de cela nous utilisons le CNN pour la sélection des caractéristiques. Nous évaluons et comparons les performances du CNN avec défèrent algorithme d'apprentissage automatique tel que (SVM, arbre de décision.), de plus nous comparons les performances de la méthode de sélection des caractéristiques à savoir Rétro Propagation (BP CNN) avec le Gain d'information (IG) le plus couramment utilisé dans la classification des textes.

Mots clés : Text Mining, catégorisation automatique de textes, représentation textuelle, Deep Learning, Réseau de neurone Convolutif, Machine Learning, Rétropropagation, Gain d'information, Sélection des caractéristiques.

Abstract

With the constant growth of the number of available numeric texts numerous researches are led in order to organize and to make exploitable this immense basis of information. This data represents an important source of information for several applications such as recommendation systems, community detection, marketing and computer vision. In this context the categorization of texts has for objective to regroup texts « thematically» near within the same category.

The majority of these approaches generally this problematic like an all. However the categorization of texts is a double problem. The first problem corresponds to the textual representation or, in other words, how to get a mathematical and numerical representation of a text, thus the selection of the best characteristics (relevant term) which ensures a better classification. The problematic second is located, mainly in the domain of the training. This is the use of one of the most recent techniques and principles of deep learning, from a practice game, to categorize all new text.

In this thesis, we apply one of the powerful deep learning models which is the convolutional neural network (CNN) on a textual data set to solve text classification problems, in addition to that we use the CNN as feature selection. We evaluate and compare the performance of CNN with deferred machine learning algorithm such as (SVM, decision tree.), Moreover we compare the performance of the feature selection method namely Back Propagation (BP CNN) with the Gain information (GI) most commonly used in the classification of texts.

Key words: Text Mining, automatic categorization of texts, textual representation, Deep Learning, Convolutional neural network, Machine Learning, BackPropagation, Information Gain, Feature Selection.

.

ملخـــص

مع الزيادة المستمرة في عدد النصوص الرقمية المتاحة ، يتم إجراء الكثير من الأبحاث من أجل تنظيم قاعدة البيانات الهائلة وجعلها قابلة للاستخدام. تمثل هاته البيانات مصدرًا مهمًا للمعلومات للعديد من التطبيقات مثل أنظمة التوصية واكتشاف المجتمع والتسويق ورؤية الكمبيوتر. في هذا السياق ، يهدف تصنيف النصوص إلى تجميع النصوص "المتشابهة موضوعيًا" في مجموعة صنف واحدة .

تعالج غالبية هاته الأساليب بشكل عام هاته المشكلة ككل. ومع ذلك ، فإن تصنيف النصوص يمثل مشكلة مزدوجة: المشكلة الأولى تتعلق بالتمثيل النصي أو ، بعبارة أخرى ، كيفية الحصول على التمثيل الرياضي والرقمي للنص ، وبالتالي اختيار أفضل الخصائص (مصطلح ذات الصلة) مما يضمن تصنيفًا أفضل ؛ المشكلة الثانية هي بشكل رئيسي في مجال التعلم الالي. يتعلق الأمر باستخدام واحدة من أحدث تقنيات ومبادئ التعلم العميق ، حيث انطلاقا من معالجة معطيات ، يمكن تصنيف أي نص جديد.

في هذه الأطروحة ، نطبق أحد نماذج التعلم العميق القوية وهو الشبكة العصبية التلافيفية (CNN) على مجموعة بيانات نصية لحل مشاكل تصنيف النص ، بالإضافة إلى ذلك نستخدم شبكة تلافيفية من اجل اختيار الخصائص. نقوم بتقييم ومقارنة أداء CNN مع بعض خوارزميات التعلم الآلي مثل (SVM ، شجرة القرار.) ، علاوة على ذلك نقوم بمقارنة أداء طريقة اختيار المميزات باستخدام CNN وهي Back Propagation مع (GI) الأكثر استخدامًا في تصنيف النصوص.

الكلمات المفتاحية : الكلمات الرئيسية: التنقيب عن النص ، التصنيف التلقائي للنص ، التمثيل النصي ، التعلم العميق ، الشبكة العصبية التلافيفية ، التعلم الآلي ، النشر العكسي ، اكتساب المعلومات ، اختيار الخصائص.

Table des figures

Liste des tableaux

Acronyme

IA : Intelligence Artificielle

TAL :Traitement Automatique de Langage.

DL : Deep Learning.

LSTM : Long Short Term Memory.

CNN : Convolutional Neural Network..

ML : Machine Learning.

RNN : Recurrent Neural Network..

SVM : Support Vector Machine.

KNN : K Nearest Neighbor.

NB : Naïve Bayes.

ANN : Artificial Neural Network.

IG : Information Gain.

BP : Back Propagation.

GBP : Guided Back Propagation.

TC : Text Classification.

FS : Feature Selection.

Table des matières

Introduction Générale

Dans les dernières décennies, le réseau d'internet a connu une explosion considérable des documents électroniques. Des milliers si ce n'est pas des millions de documents sont publiés et communiqués journalièrement, à savoir les sites web d'entreprises ou d'individus, les blogs, et ainsi des documents autrefois manuscrits sont aujourd'hui disponibles sous format numérique. De ce fait, cette gigantesque base de données documentaire produite nécessite un traitement automatique et particulier. L'organisation automatique de ces documents, offre les avantages suivants : l'accès instantané à l'information, l'organisation et l'exploitation des documents et ainsi que l'extraction des connaissances utiles et qui sont cachés au préalable. L'intelligence artificielle (IA), et plus en particulier l'apprentissage automatique s'avère, potentiellement, capable de répondre à tous ces besoins. Plusieurs disciplines ont vu le jour dans cette dernière décennie, notamment, la recherche d'informations, l'indexation automatique, la catégorisation et le regroupement parmi d'autres.

L'apparition d'une nouvelle technologie qui permet d'extraire des nouvelles connaissances à partir des sources de données massives et variées, le Machine Learning (Apprentissage automatique), permet d'y remédier.

L'apprentissage automatique (Machine Learning en anglais) est une solution alternative à remédier au problème d'extraction automatique de nouvelles connaissances à partir des sources de données massives et variées. En effet, le ML est un sous domaine de l'intelligence artificiel et le sujet le plus brûlant des nouvelles technologies de l'information. C'est la solution d'entrainer un ensemble d'algorithmes sur de grandes quantités de données (Big Data), afin d'en extraire la connaissance et faire des prédictions avec plus d'efficacité que l'intelligence humaine. Elle économise du temps, des moyens et des personnes pour la compréhension de ces informations. '' Le ML C'est une branche de l'intelligence artificielle basé sur l'idée que les systèmes peuvent apprendre à partir de données, identifier des modèles et prendre des décisions avec un minimum d'intervention humaine [1]**.**

Une nouvelle famille des méthodes d'apprentissage automatique à vue le dans ces dernières années, appelée l'apprentissage profond (Deep learning en anglais. Elle se base sur des réseaux de neurones artificiels inspiré du fonctionnement du cerveau humain . L'apprentissage profond est construit pour gérer de larges quantités de données en ajoutant des couches au réseau, cette technique a complètement bouleversé le domaine de l'intelligence artificielle **[2]**. Dernièrement, les approches d'apprentissage profond obtiennent de meilleurs résultats de performances par rapport aux algorithmes d'apprentissage automatique dans plusieurs domaines d'application à savoir, la reconnaissance et la classification d'images, la reconnaissance faciale et aussi le traitement automatique des langues naturelles (TALN ou NLP pour Natural Language Processing), etc.

L'Objectif du mémoire

L'objectif de ce mémoire consiste, à proposer une nouvelle architecture du réseau de neurones qui sera implémenté et intégré dans notre modelé d'apprentissage profond CNN. Cette variante du réseau CNN de Deep Learning aura comme objectif de classer les documents textuels. En plus de la partie classification, nous proposons une deuxième contribution qui sert de son coté à investir dans la sélection des attributs en utilisant le modèle proposé précédemment de CNN. Pour valider notre proposition nous allons comparer l'apprentissage profond appliqué dans ce travail avec les algorithmes d'apprentissage ML. Sur le plan stratégique de notre processus adopté à la classification textuel, ce dernier comprend deux étapes. La première, traite la représentation textuelle et catégorielle qui consiste en un prétraitement de textes (élimination des mots vides, la lemmatisation, stemmatisation…). Le choix d'une représentation numérique (vectoriel) des documents textuels, et une réduction de dimensions. La seconde étape étudie la classification qui comprend l'application de notre modèle de Réseau de neurones artificiel (convolution ou récurent) sur un corpus (data-set) pré-entrainé pour construire un modèle de prédiction, ce dernier servira à la partie de la classification des documents textuels.

Organisation du mémoire

Ce mémoire s'articule autour de cinq chapitres organisés de la façon suivante :

- Chapitre 1, on présente la classification automatique des textes, l'apprentissage automatique d'une façon globale : l'historique et le processus d'apprentissage automatique, les déférents types d'apprentissage automatique sont décrits en

détails dans ce chapitre. À la fin de la première partie du chapitre 01, nous donnons une brève description des déférents types d'algorithmes d'apprentissage automatique.Dans la deuxième partie, le travail sera axé sur les préliminaires de *Deep learning* et ses déférents modèles, les principales architectures neuronales les plus utilisé (convolutif, récurrents) ainsi que ses divers objectifs. Nous allons exposer le traitement automatique du langage naturel (TALN).

- Chapitre deux, s'articule sur le processus de codage de données textuelles et enfin les déférentes opérations de prétraitement et les méthodes de représentation numérique des textes.

- Chapitre trois, on s'intéresse à la réalisation axée sur l'étude comparative entre les performances du modèle d'apprentissage profond et l'apprentissage automatique classique. En deuxième lieu, on va spécifier l'importance de l'étape de sélection des caractéristiques pour le problème de classification en faisons une comparaison entre les défèrent technique d'identification des attributs.

- Chapitre quatre, une conclusion permet de conclure le travail effectué dans ce mémoire tout en proposant dans sa fin des perspectives pour de futurs travaux.

Chapitre I

Classification de textes

Chapitre I

Classification automatique de textes

1.1 Introduction

Le progrès technologique des années 80, a été le moteur des premières avancées dans le domaine de l'augmentation des capacités de stockage numérique et par conséquent, le nombre de documents textuels à traiter. L'évolution actuelle est due à l'accroissement du nombre de textes numériques disponibles. En novembre 2004, Google recensait à lui seul plus de 8 milliards de pages web. Lorsque les volumes de documents en jeux sont tels, et que leurs contenus ne peuvent être analysés manuellement, il est indispensable de créer des outils d'exploitation des informations contenues dans ces données. Le développement du domaine de la classification de texte répond à cette volonté de gestion par contenu des sources volumineuses de textes.

Les techniques de la classification de textes combinent des approches, issues de disciplines comme la linguistique informatique (analyse de textes et extraction d'informations), l'analyse de données (analyse factorielle, classification,), l'intelligence artificielle (techniques d'apprentissage, règles d'inférence). Dans chacune de ces applications, le principe de base est de regrouper les documents similaires selon leur contenu. Deux approches sont possibles : La classification supervisée (catégorisation) ou classement automatique de documents dans des classes préexistantes. La classification non supervisée des documents (clustiring) qui correspond à la découverte de classes de documents sans à priori.

Pour optimiser les processus de classification de textes et répondre au besoin des entreprises, l'intelligence artificielle (IA) devient l'un des piliers de l'innovation technologique au sein de ces derniers. Le terme d'IA est souvent employé pour désigner une technologie qui permet aux ordinateurs de simuler l'intelligence humaine grâce notamment à l'apprentissage automatique (en Anglais : Machine Learning).

On limite le terme d'apprentissage automatique (ML) comme une branche majeure de l'IA, qui représente les outils et programmes ayant pour objet d'analyser des quantités astronomiques de données. En conséquent on aura, des modèles de prédiction qui aident à la décision et capable d'apprendre à partir les données d'entrainement **[3] .**

Dans ce chapitre le travail sera axé sur les préliminaires de la classification de texte. Nous allons donner le processus de catégorisation et les déférentes les techniques d'apprentissage automatique. Ensuite, nous enchainons par introduire quelque mesure d'évaluation et ainsi qu'une partie sera consacré pour l'apprentissage profond.

1. 2 La catégorisation de textes

1.2.1 Définition de la catégorisation : Les définitions de la catégorisation sont nombreuses et complémentaires :

- La catégorisation de textes est la recherche d'une relation bijective qui consiste à "*chercher une liaison fonctionnelle entre un ensemble de textes et un ensemble de catégories (étiquettes, classes)*" **[4].**
- En ajoutant la notion de classes cibles prédéfinies, la catégorisation de textes est une tâche de tri **[5].**
- Dans la littérature scientifique, les termes classification et catégorisation sont indifféremment utilisés **[6].**
- La catégorisation est l'action d'affecter des éléments, qui possèdent des caractéristiques communes, à des catégories préétablies, sans relation d'ordre **[7].**

Formellement : soit un ensemble de documents textuels D et un ensemble de catégories C. Le but du processus de catégorisation est d'approximer la fonction de catégorisation $\breve{\theta}$ appartenant à la famille de fonction $\breve{\Theta}$, qui associe à chaque couple (document, catégorie) une valeur binaire ({- 1, 1}) en fonction de l'appartenance ou non du document à une catégorie : $\breve{\theta} : D \times C \longrightarrow \{-1, 1\}$

Se basant sur cette formule, il est possible de modéliser tout processus de catégorisation qu'il soit **[8] [9].** Vu de la subjectivité du problème, il paraît difficile de calculer la fonction de catégorisation exacte $\breve{\theta}$. L'objectif de la phase d'apprentissage est d'approximer au mieux cette fonction de catégorisation. Une fois la fonction de catégorisation « obtenue », il est important de pouvoir évaluer le résultat de cette dernière en termes de catégorisation afin de déterminer l'efficacité du classificateur.

1.2.2 Application de la catégorisation de textes : La classification du texte a été appliquée à une grande variété d'applications pratiques à savoir :

- Comprendre et analyser les sentiments de l'audience à partir des médias sociaux
- Filtrage des emails spams et non-spam
- Marquage automatique des requêtes des clients
- Catégorisation des articles d'actualité topics en sujets prédéfinis catalogage des articles de presse

1.2.3 Le processus général de catégorisation de textes

Le processus de catégorisation se déroule de façon générale selon les trois phases suivantes : apprentissage, validation et test (évaluation), comme est illustré ci-dessus dans la (**Figure. 1.1)**.

1.2.3.1 Phase d'apprentissage

De façon générale la phase d'apprentissage comporte deux étapes à savoir : l'étape de codage numérique de textes et l'étape de catégorisation. L'objectif de la phase d'apprentissage est d'induire une fonction de catégorisation à partir d'un ensemble de données appelé jeu d'entraînement (textes brutes). Ces dernières contiennent des unités de texte, documents. Un document se compose d'un nombre de phrases de sorte que chaque phrase comprend une suite de mots sépares évidement par des espaces. Dans la catégorisation automatique (approche supervisé), On appelle jeu d'entraînement un ensemble de documents dont la catégorie est connue a priori (étiqueté). La première étape de cette phase est la formalisation (numérisation) des textes afin qu'ils soient directement exploitables par l'algorithme d'apprentissage. La représentation textuelle la plus connue est l'utilisation d'une représentation vectorielle **[10]** qui se base sur Le modèle le plus simple appelé Sac de Mot (Bag of Words) (BoW). Cependant, cette étape est enchainée par l'étape de réduction des dimensionnalités pour réduire la taille du vocabulaire et aboutir à des bonnes performances de classifieurs. Une fois les données textuelles formalisées et « étiquetées », un algorithme d'apprentissage est utilisé pour entrainer le jeu de donné d'apprentissage afin d'extraire une fonction et construire un modèle de prédiction, permettant d'associer tout nouveau texte à une catégorie. Dans le but d'atteindre un degré maximal de précision et d'efficacité plusieurs classificateurs ont été mis au point. Parmi ces derniers, on trouve : les algorithmes d'apprentissage automatique (classificateurs des

textes) disponibles sont : les réseaux de neurones (NN), les K plus proches voisins (KNN), les arbres de décisions (TD), les réseaux bayesiens (BN), les Support Vecteurs Machines (SVM), les techniques d'apprentissage profond (en Anglais : Deep learning).

1.2.3.2 La phase de validation

La phase de validation a pour objectif d'ajuster automatiquement les paramètres nécessaires aux algorithmes de la phase précédente. En effet, la majorité des algorithmes d'apprentissage (classificateurs) dépend d'un ensemble de paramètres, et ce qui est nécessaire de les fixer correctement.

1.2.3.3 Le test

La phase de test permet d'évaluer, de façon définitive, le processus de catégorisation grâce au jeu de test. Le jeu de test, comme le jeu de validation, est un ensemble de couples (document, catégorie) dont l'information associée à la catégorie n'est utilisée que lors de l'évaluation du système. Comme le sens d'un document est une notion subjective, l'évaluation d'un processus de catégorisation s'effectue empiriquement sur un jeu de test. Une fois le jeu de test catégorisé, les affectations fournies par le système sont comparées celles qui sont préalablement définies par un expert. En conclusion, l'objectif d'un processus de catégorisation est donc de maximiser une fonction de score.

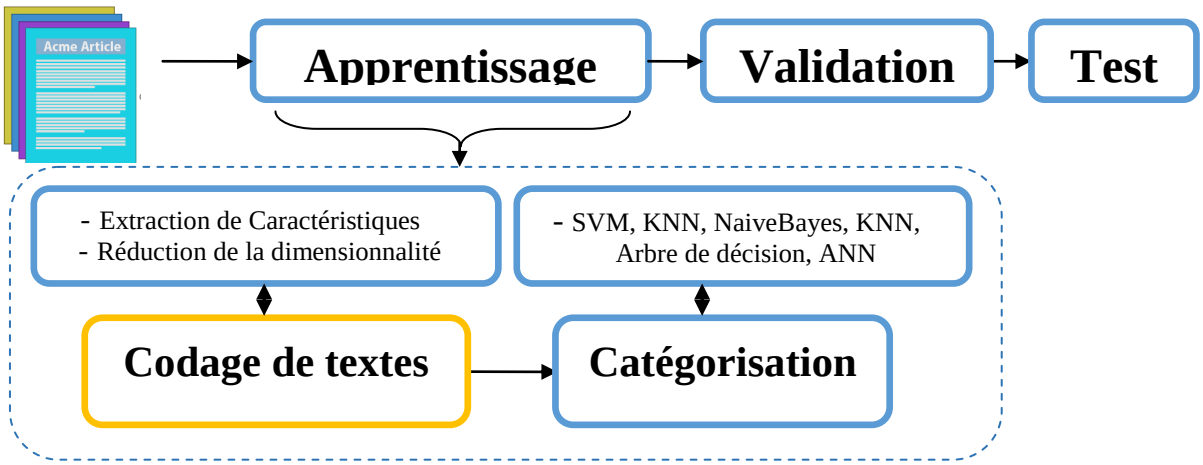

Figure. 1.1 Le processus de Catégorisation de textes

Les mesures d'évaluation

Mesurer la qualité d'un classifieur est une étape nécessaire dans tout processus d'apprentissage. Les principales mesures d'évaluation sont les suivantes **[11]** :

- **Précision** : nombre de documents correctement attribués à une catégorie sur le nombre total de documents attribués à cette catégorie. Voir formule 1

$$P = TP / TP + FN \quad \text{.....(1)}$$

- **Rappel** : nombre de documents correctement attribués à une catégorie sur le nombre total de documents prévus de cette catégorie. Voir formule 2

$$R = TP / TP + FP \quad \text{..... (2)}$$

- **F1 (F-mesure) :** métrique qui combine précision et rappel en utilisant la moyenne harmonique. Voir formule 3

$$F1 = 2PR / P + R \quad \text{.....(3)}$$

Si l'évaluation est effectuée dans des environnements multi-classes ou multi-labels, la méthode devient légèrement plus compliquée car les métriques de qualité doivent être soit affichées par catégorie, soit globalement agrégées. Il existe deux principales approches d'agrégation **[12]** :

- **Macro-average** : pondère également toutes les classes, quel que soit le nombre de documents qui lui appartiennent (toutes les classes ont la même importance).

- **Micro-average** : pondère également tous les documents, favorisant ainsi les performances sur les classes communes. (Tous les documents ont la même importance).

1.2.4 Les problèmes de la catégorisation : Ils se résument à :

- **La grande dimension** :
  - Taille des corpus (majorité des cellules vides)
  - Le coût en temps des algorithmes de classification.
  - Le risque de sur apprentissage.
- **La complexité de l'algorithme** : (la grande taille du vecteur des mots)
  - Réduction de l'efficacité et performance des algorithmes d'apprentissage.
  - Sensibilité au nombre de variables utilisées pour coder le texte.
- **Le sur apprentissage** : Classement correcte des exemples d'apprentissage, et classement erroné des nouveaux exemples de test.
- **Le déséquilibrage** : Les classes peu nombreuses sont mal représentées.
- **L'ambiguïté** : Un même mot peut avoir plusieurs sens et définitions différents.

- **La synonymie** : Les synonymes sont des mots ou des expressions qui indiquent le même sens
- **La subjectivité de la décision** : Les experts ne sont pas d'accord sur la classe d'appartenance du document ou du texte.

1.3 Introduction à L'Apprentissage Automatique (*en anglais : Machine Learning*)

Etant donnée la variété d'apprentissages qu'on peut rencontrer, il est aisé de deviner que les fondements de cette discipline, en occurrence l'apprentissage automatique, proviennent de diverses sciences :

- Des mathématiques pour l'informatique : algèbre linéaire, la probabilité, la logique, l'analyse élémentaire, …
- La théorie statistique de l'estimation,
- L'apprentissage Bayésien,
- L'inférence grammaticale ou l'apprentissage par renforcement, et tant d'autres **[13].**

De manière générale, on peut definir le ML vomme suit : constitue comme une discipline de l'IA qui offre aux ordinateurs la possibilité d'apprendre à partir d'un ensemble d'observations que l'on appelle ensemble d'apprentissage avec un minimum d'intervention humain.il est explicitement lié au Big Data, étant donné que pour apprendre et se développer, les ordinateurs ont besoin de flux de données à analyser, sur lesquelles s'entraîner. L'Apprentissage Automatique (ML) et le Big Data sont donc interdépendants.

L'Apprentissage Automatique (ML) consiste à faire apprendre à la machine un modèle de prédiction à partir d'un ensemble de données appelé ensemble d'apprentissage. Ce modèle de prédiction est obtenu grâce à un algorithme d'apprentissage automatique. Ce modèle servira ensuite à prédire les cas non introduits lors de la phase d'apprentissage **[14].**

1.3.1 Les types d'apprentissage Automatique (ML)

L'apprentissage automatique est axé sur l'analyse prédictive (permet de prévoir ce qui est le plus susceptible de se produire à l'avenir) et prescriptive (recommande le plan d'action le plus logique pour atteindre le résultat souhaité), en fonction de la nature de l'analyse et des

algorithmes utilisés. Cette section donne un aperçu des types le plus courants d'apprentissage automatique.

1.3.1.1 Apprentissage supervisé

Cette approche est divisée en deux phases : phase d'apprentissage et phase de teste. La première, consiste au choix d'une base d'apprentissage représentant l'ensemble des exemples d'entraînement avec leurs étiquettes de classes correspondantes. A partir de ces exemples, on infère les paramètres des fonctions de classement en utilisant les méthodes mathématiques correspondantes. La deuxième, une fois les règles générées, on sélectionne une base de test composée de nouveaux exemples, pour lesquels on cherche à déterminer leurs classes d'appartenance, en les comparants aux classes de références fournies par l'expertise externe (superviseur). Cette phase permet de valider les règles de classification, en fonction du taux de réussite obtenu **[15].**

1.3.1.2 Apprentissage Non supervisé

Il vise à concevoir un modèle structurant l'information. La différence ici est que les comportements (ou catégories ou encore les classes) des données d'apprentissage ne sont pas connus, c'est ce que l'on cherche à trouver **[16].**

1.3.2 Les Algorithmes d'Apprentissage automatique (ML)

Le choix du bon algorithme d'apprentissage automatique dépend de plusieurs facteurs, notamment la taille, la qualité et la diversité des données, ainsi que les réponses que l'on souhaite obtenir à partir de ces données. Il y a bien d'autres aspects à prendre en compte, comme la précision, le temps d'entraînement, les paramètres et les points de données. Choisir le bon algorithme dépend donc à la fois des besoins, des spécifications des essais ainsi que du temps disponible. Voici un résumé sur les algorithmes les plus fréquemment utilisés pour l'apprentissage automatique :

1.3.2.1 L'Arbre de Décision

Un arbre de décision est un arbre permettant de classer des objets en sous-classes par divisions hiérarchiques, dans lequel un nœud représente une sous-classe du nœud parent et un arc représente un prédicat de placement des objets de la classe parente dans la sous-classe. **[17]**. Pour construire un arbre de décision, l'idée de base est de diviser, de façon récursive, l'échantillon d'apprentissage en plusieurs sous-ensemble (dans le cas d'un arbre

n-aire) ou en deux sous-ensemble (dans le cas d'un arbre binaire), de telle sorte que les nœuds descendants soient plus homogènes que les nœuds parents et qu'ils soient les plus différents possibles entre eux.

Dans les réseaux de neurones, on distingue :

- Les *nœuds internes* qui engendrent deux nœuds descendants immédiats, le fils gauche correspond à une réponse positive au test et le fils droit à une réponse négative.
- Les *nœuds terminaux* ou *feuilles* qui ne peuvent plus être divisés.

Les différentes phases de construction de l'arbre sont **[18]**:

1. Établir pour chaque nœud l'ensemble des divisions admissibles.
2. Définir un critère permettant de sélectionner la « meilleure » division d'un nœud parmi toutes celles admissibles pour les différentes variables.
3. Définir une règle permettant de décider qu'un nœud est terminal (feuille).
4. Affecter chaque feuille à l'une des classes (cas de la discrimination) ou à une valeur de la variable à expliquer (cas de régression).
5. Estimer le taux d'erreur associé à l'arbre.
6. Élaguer l'arbre de décision obtenu.

1.3.2.2 Naïve Bayes

La catégorisation de type Naïve Bayes est un catégoriseur probabiliste fondamentalement basé sur le théorème de Bayes. Si l'on considère $\boldsymbol{v}_j = (w_1, \ldots, w_k, \ldots, w_d)$ un vecteur de variables aléatoires représentant un document $\boldsymbol{d}_j$, la probabilité que ce dernier appartienne à la catégorie $\boldsymbol{c}_i$ est définie par **[19]** (Voir formule 1) :

$$P(c_i / v_j) = \frac{P(c_i) P(v_j / c_i)}{P(vi)} \quad \ldots\ldots\ldots\ldots (1)$$

Chaque variable aléatoire du vecteur $\boldsymbol{v}_j$ représente un terme d'indexation et w_k symbolise la valeur de ce dernier au sein du document associé. Afin de minimiser les erreurs de catégorisation, la politique de catégorisation d'un catégoriseur de type Naïve Bayes est d'affecter $\boldsymbol{d}_j$ à la catégorie C_{max} dont la probabilité est la plus élevée, c'est-à-dire (Voir formule 2) ;

$$c_{\max} = \arg\max_{c_i \in C} P(c_i / v_j) \quad \ldots\ldots\ldots\ldots (2)$$

L'approximation de probabilité conditionnelle $P(c_i \mid v_i)$ s'effectue sur le jeu d'entraînement selon : le modèle des variables aléatoires dites de Bernoulli ou le modèle multinomiale. Quant à l'approximation de $P(c_i)$ celle-ci est définie dans les deux modèles de la façon suivante. Voir formule 3 :

$$\widehat{P}(c_i) = \frac{\{\ \}\text{des documents appartenant à } c_i}{\{\ \}\text{des documents}} \quad \ldots\ldots\ldots\ldots\ (3)$$

On peut résume les étapes de l'algorithme comme suit :

Construire un vecteur de type TF ou de type booléen (cela dépend de la méthode utilisée : Bernoulli ou multi nominal) à partir d'un document appartenant à ***D***. La représentation sous une forme vectorielle n'est pas obligatoire et d'autres types de représentations auraient pu être utilisés. Néanmoins la représentation choisie doit permettre de déterminer $P(v_j)$ (avec l'égalité $P(v_j) = P(d_j)$). Évaluer, la probabilité $P(c_i \mid v_j)$. La phase d'apprentissage se résume à déterminer les probabilités d'apparition des termes d'indexation au sein du corpus. Finalement affecter le texte à la catégorisation la plus probable.

1.3.2.3 Rocchio

Rocchio est un des plus vieux algorithmes de classification et l'un des plus simples. Dans cette approche, les documents sont représentés de manières classiques par un vecteur (habituellement de type TF-IDF). Les catégories sont représentées par un vecteur appartenant au même espace vectoriel que les documents. Le vecteur catégoriel se défini comme étant une moyenne pondérée de deux vecteurs qui sont :

1. Le vecteur représentant la moyenne des vecteurs des textes qui appartiennent à la catégorie.
2. Le vecteur représentant la moyenne des vecteurs des textes que la catégorie ne contient pas.

Un profil prototypique [c] est calculé pour chaque classe selon : $c_w = \frac{t}{N_c}\sum_{d \in c} d_w - \frac{1-t}{N_{\bar{c}}}\sum_{d \notin c} d_w$

Où N_c est le nombre de documents dans c, $N_{\bar{c}}$ est le nombre de documents n'appartenant pas à c, et ***t*** est un paramètre du modèle compris entre 0 et 1. Dans les situations où un document peut être attribué à une seule classe, ***t*** est souvent positionné à 1. Ces profils correspondent au barycentre des exemples (avec un coefficient positif pour les exemples de

la classe et négatif pour les autres). Ces vecteurs sont également normalisés de la même façon que les documents. Le classement de nouveaux documents s'opère en calculant la distance euclidienne (équivalente au produit scalaire et à la similarité en cosinus puisque tous les vecteurs sont de norme 1) entre la représentation vectorielle du document et celle de chacune des classes. Le document est assigné à la classe la plus proche.

1.3.2.4 K plus proches voisins (k-Nearest Neighbor)

K-NN (K-Nearest Neighbor) est une méthode très connue dans le domaine de la catégorisation automatique. Ses performances la situent parmi les meilleures méthodes de catégorisation **[16]**. La représentation textuelle utilisée dans k-NN est une représentation classique de type TF-IDF. Chaque texte est représenté par un vecteur dont chaque axe est défini en fonction de la fréquence d'apparition du segment textuel associé (à l'axe). Les segments textuels utilisés la plupart du temps, comme terme d'indexation, dans une approche de type k-NN, sont des radicaux ou des lemmes. Une des particularités de cette approche est qu'il n'y a pas de représentation catégorielle à proprement parler. Aucune phase d'apprentissage n'est donc nécessaire car chaque catégorie est définie par l'ensemble des vecteurs des textes qu'elle contient. Chaque nouveau texte ***d*** est comparé par le classificateur à l'ensemble des textes du jeu d'apprentissage ($\boldsymbol{C_{app}}$) en calculant la similarité du document, avec l'ensemble des autres exemples du corpus. Et il est affecté à la catégorie majoritaire déduite de ses *k* plus proches textes. La catégorie de ***d*** attribuée par le vote de ces k documents, donnant lieu aux formules suivantes :

$$- \qquad f(d) = \arg\max_{c_j \in C} \sum_{i=1}^{k} (d_i, c_j)$$

$$f(d) = \arg\max_{c_i \in C} \sum_{i=1}^{k} sim(d, d_i) \times C(d_i, c_j)$$

1.3.2.5 les machines à Support de vecteurs SVM

Les premières ébauches sur le sujet datent des années 60 avec les travaux de Chervonenskis **[20]**de réelles applications ont vu le jour depuis l'article de Vapnik **[21]**.

L'approche par SVM permet de définir, par apprentissage, une surface de séparation entre des exemples positifs et négatifs maximisant la marge entre deux catégories, de type c et $\bar{c}$, afin de minimiser par la suite le risque d'erreurs lors de la catégorisation. Cette catégorisation est donc de type binaire. Dans le cas de catégorisation à plusieurs catégories,

une surface de Séparation est définie pour chacune d'elles. Ensuite, un document est affecté à une catégorie si ce dernier est du « bon » côté de l'hyperplan qui la représente. SVMs non seulement tendent vers trouver une séparation mais aussi elles essayent de générer un séparateur optimal dans un sens de tracer un hyperplan, une droite, qui maximise la distance (la marge) entre les bornes ou les frontières des nuages de points positifs et négatifs à la fois.

1.3.2.6 Les Réseaux de neurones

Sont des réseaux inspirés de système neuronal biologique, qui contient un ensemble d'éléments ou unités extrêmement simples (neurones) se comportant comme des fonctions de seuil, suivant une certaine architecture. Chaque neurone prend en entrée une combinaison des signaux de sortie de plusieurs autres neurones, affectés de coefficients (les poids). Le célèbre modèle Perceptron, est le premier modèle développé par Frank Rosenblatt dans les années 1958. Le réseau Perceptron est une structure, réseau mono couche, dont on a plusieurs entrées Xi et une seule sortie Y. Ces entrées sont pondérées par des poids Wi appelé poids synaptiques. La sortie Y, la valeur de classe, est calculée récursivement sur la base d'une fonction prédéfinie et en fonction des entrées Xi, le vecteur de caractéristiques, et leurs poids synaptiques qui sont aléatoirement initialisés. L'apprentissage dans cette architecture, le Perceptron, se fait au niveau de ses poids en les modifiant. Autrement dit, si la valeur calculer de Y est différente de celle de la théorie, cas de classification supervisé, alors une mise à jour des poids synaptiques est faite en fonction de l'écart produit entre la sortie calculer et la valeur théorique de Y, cet apprentissage est connu sous le nom : la loi de HEBB. Ce processus est répété jusqu'à on atteint un écart de zéro ou un nombre maximal d'itérations.

Les réseaux de neurones simples tel que perceptron mono et multicouche, sont aptes à gérer des données structurées. Un type de données non structuré comme le texte brut et libre écrit en langage naturel ne peut pas être traité directement par ces algorithmes. La prochain partie, traite les détails d'une nouvelle technologie qui est une branche de l'apprentissage automatique appelé l'apprentissage profond (en anglais :Deep Learning) qui se base sur les réseaux de neurones profond, ainsi montrer les déférents modèles et les outils utilisés pour résoudre les problème d'apprentissage automatique en précisant leur domaine d'application .

1.4 L'Apprentissage Profond (*en anglais : Deep Learning*)

1.4.1 Définition : d'une façon générale, L'apprentissage profond (Deep Learning, Deep Structured Learning , Hierarchical Learning) est une famille des méthodes dérivé du machine Learning , qui a permis des avancées importantes en intelligence artificielle dans les dernières années **[22].**

Le Deep Learning est un ensemble des techniques qui permettent de donner la capacité a une machine d'apprendre et de s'améliore de façon autonome (par elle-même) indépendamment de l'intervention d'un expert, basé sur l'idée des « **réseau de neurone artificiel** » inspiré à l'origine par une étude du fonctionnement physiologique du cerveau humain et il est taillé pour gérer de larges quantités de données en ajoutant des couches au réseau. Ce modèle de Deep Learning a la capacité d'extraire des caractéristiques à partir des données brutes (non structuré) grâce aux dizaines et centaines des « **couches** » composé de millier d'unité de « **neurones** », chacune recevant et interprétant des informations de la couche précédente afin d'extrait des nouvelles caractéristiques plus complexe servent d'entrées pour la couche suivante et ainsi de suite **[23].**

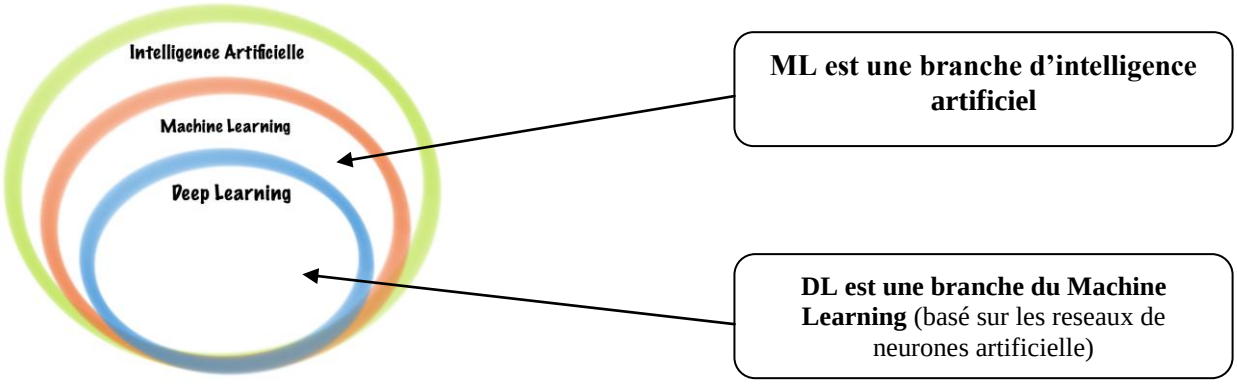

Figure. 1.2 Relation entre les concepts d'intelligence artificiel (AI),(ML),(DL) [24]

1) Selon les fondateurs Yann LeCun, Yoshua Bengio Geoffrey Hinton dans **[25]** :

"L'apprentissage profond permet aux modèles informatiques composés de plusieurs couches de traitement d'apprendre des représentations de données avec plusieurs niveaux d'abstraction."

2) Autre définition par les auteurs dans **[26]** :

"L'apprentissage profond est une classe de techniques d'apprentissage machine, où L'information est traitée en couches hiérarchiques pour comprendre les représentations et les caractéristiques des données dans des niveaux de complexité croissante."

1.4.2 Pourquoi avons-nous besoin du DL ?

- Dispositifs super intelligents.
- Des besoins croissants dans le domaine de l'IA .
- Meilleure solution pour la reconnaissance d'image et vocale, traitement du langage naturel (NLP).
- Utiliser de grandes quantités de données d'entraînement (BigData) **[27].**
- L'augmentation de la puissance des ordinateurs assure plus de performance des modèles d'apprentissage et également une meilleure prédiction.
- L'apprentissage Profond pouvant être appris pour représenter des informations mondiales, visuelles et linguistiques.
- La capacité d'apprendre et de s'améliore de façon autonome qui nécessite un peu d'intervention de la part du programmeur.
- L'extraction automatique des caractéristiques permet d'atteindre un taux de précision élevé pour les taches de vision par ordinateur **[28].**

1.4.3 Déférence entre ML et DL :

Il est important que les organisations comprennent clairement la différence entre la machine Learning et le Deep Learning, tel que :

- l'apprentissage automatique est un concept basé sur l'application des algorithmes sur les données traité et analysé (donné structuré) statistique afin de créer un modèle de prédiction qui aide a la décision , ces algorithme doivent connaitre comment faire des prédictions précise , en lui fournissant plus d'informations et détaille avec l'intervention du programmeur , tandis que dans le cas de l'apprentissage profond, l'algorithme est capable d'apprendre cela de façon autonome grâce à son propre traitement de données .
- Les algorithmes d'apprentissage automatique classique possèdent une borne supérieur appelé "**plateau de performance** » a la quantité de donnes qu'ils peuvent recevoir et interprété, tandis que les algorithmes d'apprentissage profond n'ont pas de telle limitation c'est qu'il s'adapte bien avec le « BigData » et plus que la quantité de donné fournis est grande, plus que les performances sont meilleures. Ils sont même allés jusqu'à dépasser la performance humaine dans des domaines comme l'image processing **[29].**
- Les modèles d'apprentissage automatique ne disposent pas du mécanisme pour identifier les erreurs, dans de tels cas, le programmeur doit intervenir pour régler le

modèle pour des décisions plus précises, tandis que les modèles d'apprentissage profond peuvent identifier la décision inexacte et corriger le modèle de lui-même sans intervention humaine.

- Dans Les algorithmes d'apprentissage automatique traditionnelle, l'étape d'extraction des caractéristiques « **features** »se fait manuellement par des spécialistes, prend du temp, couteuse et difficile a manipulé, alors que dans le (DL) cette étape est faite automatiquement durant le processus d'apprentissage par l'algorithme elle-même. **[30].** Comme illustré dans la figure suivant (**Figure 1. 3**) :

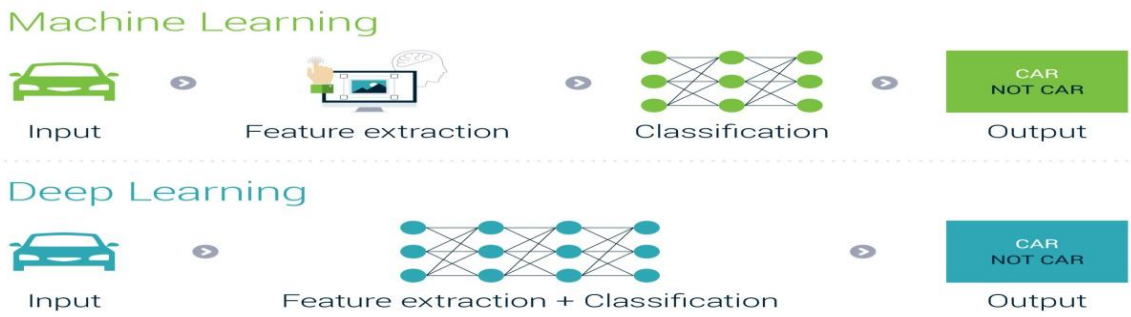

Figure. 1.3 Déférence entre DL et ML (classification d'une image) [31]

1.4.4 Principes de Fonctionnement

Comme expliquer précédemment, le réseau de neurone profond tire son nom en raison d'un nombre élevé de couches, tel que chaque réseau profond comporte 3 composant principale : **couche d'entrée**, **couches cachées** et **couche de sortie** (**Figure. 1.4**).

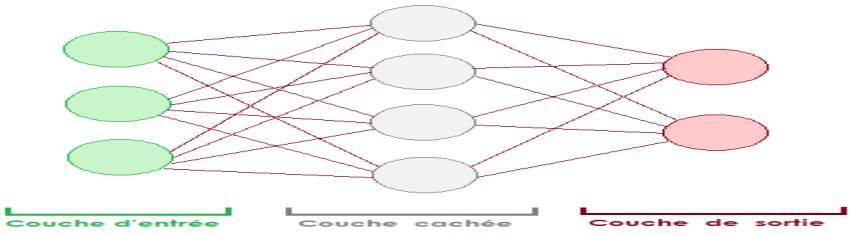

Figure. 1.4 structures de réseau de neurone

1.4.4.1 Type des couches dans les réseaux profonds :

Les réseaux de neurones simples ne comportent que 2 a 3 couche caché contrairement au réseau de neurone profond qui peuvent compter jusqu'a 150 couche avec un seuil de 300 neurone par couche (notion du terme « profond ») **[32].** Tel que Chacun de ces couches comporte des centaines des nœuds appelé les « **neurones** » interconnecté les unes des autres via des liaisons dirigées « **flèches** », tel que chaque neurone (nœud) étant une

unité de traitement qui exécute une fonction de nœud statique sur son signal entrant pour générer une sortie de nœud unique **.** Chacun des neurones de réseau est connecté avec tous les neurones de la couche suivante et chacun de ces connexion (flèches) porte « **une poids** ».

1.4.4.2 Les poids :

Sont utilisé pour attacher une certaine valeur d'importance (préférence) a une certaine fonctionnalité par rapport à d'autre pour obtenir les sorties souhaitées.Tous les poids des flèches sont utilisés pour **calculer la somme pondérée** pour chaque neurone d'une couche caché, et chacune de ces dernières exécute une **fonction d'activation** qui lui est associée.

1.4.4.3 La fonction d'activation :

Joue un rôle très important dans les réseaux neuronal profond, et sans ça le réseau profond agirait comme un simple model de régression linéaire, car cette fonction d'activation capable de prend la décision si un neurone doit être activé ou non, en fonction de leur somme pondérée. Elle est utilisée pour minimiser le taux de calcule en évitant l'activation simultané de tous les nœuds et faire des calcule pour des taches plus complexes ainsi pour introduire la non linéarité en utilisant des fonctions comme **sigmoid et tanh (Fig. 2.6) .**

■ Fonctionnement

- Les valeurs entrantes dans un neurones **(x_1, x_2, x_3, ..., x_n)** sont multipliées avec leur poids (référence au poids synaptique) qui leur sont associés **(w_1, w_2, w_3, ..., w_n)** .
- On fait ensuite la somme de ces multiplications et on ajoute enfin le biais .

$$Z(x_1,x_2) = w_1x_1 + w_2x_2 + b$$

$$Z(x) = \sum(w_i x_i) + b$$

- La fonction **Z(x)** correspond à la pré-activation, c'est à dire l'étape qui précède l'activation. Ensuite, la fonction d'activation intervient.
- Le résultat **Z** de la fonction de pré-activation **Z(x)** est interprété par une fonction d'activation **A(z)** produisant en sortie un résultat **Y**.

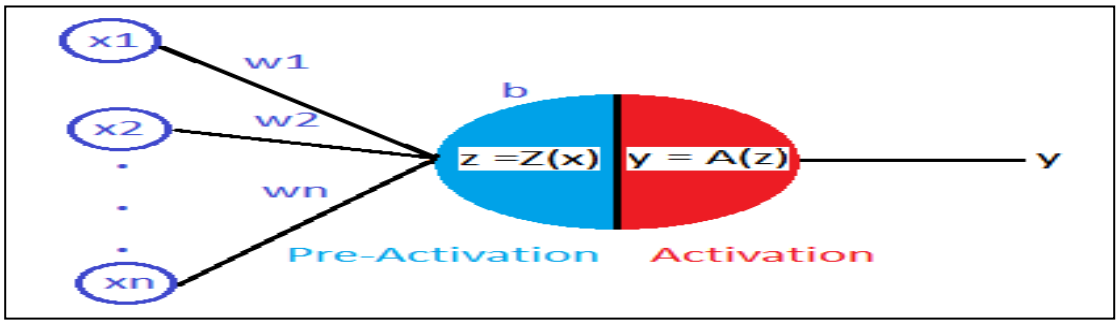

Figure. 1.5 principes de la fonction d'activation

1.4.4.4 Forward propagation :

C'est un processus de propagation ver l'avant a travers les couches du réseau, qui commence par les traitements effectuer dans le réseau profond (calcule des somme pondéré), ainsi l'intervention des fonctions d'activation correspondant à chaque couche cachée qui agit sur les sommes pondérées pour donner un rendement final **[33].**

1.4.4.5 Back propagation (BP) :

C'est un processus indispensable dans les réseaux de neurones profond. Faire une propagation en arrière a travers le réseau pour déterminer les erreurs et les pertes en comparant les sorties prévues avec les sorties réelle. Ce processus utilisé pour régler les poids et les biais de telle sorte que nous obtenions les valeurs optimales. Chaque exemple d'entrées faire une propagation et back propagation afin de mettre à jour les poids utilisés par les neurones plus proches de l'entrée.

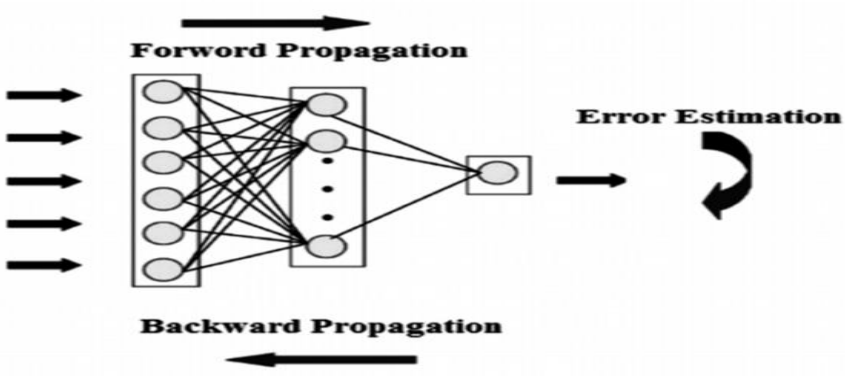

Figure. 1.6 le concept de BackPropagation [34]

1.4.5 Les différents types de modèles Deep Learning :

Deep Learning est un domaine à croissance rapide, et de nouvelles architectures, variantes ou algorithmes apparaissent toutes les semaines. C'est pour ça il existe beaucoup de variantes d'architectures et model des réseaux profonds qui sont dérivé d'une

architecture originale. Les premiers réseaux de neurones qui sont utilisé dans les années 80 ne peux comporter qu'une seule couche cachée en raison du cout de calcule, limitation des performances des machines et la disponibilité limitée des données. De nos jour, l'évolution technologique dans les domaines « data science », « intelligence artificiel » et « BigData », nous permettre d'avoir des réseaux avec des architecture neuronal profond qui peuvent comporte des dizaines et des centaines des couche caché, d'où le surnom d'apprentissage profond.

Les différents types de réseaux de neurones disponibles à l'utilisation ont des modèles très utilisé et très célèbre dans le domaine Deep Learning, tels que les réseaux de neurones Convolutifs (**CNN**), les réseaux de neurones récurrents (**RNN**) et (**LSTM** « Long Short Term Memory »).

1.4.5.1 Le réseau de neurone Convolutif (CNN)

Un réseau de neurone a convolution est un type de réseau de neurones artificiels acycliques (**feed-forward**), apparu en 2012 lors de la compétition annuelle de vision par ordinateur. Il se base sur une méthodologie similaire à celle des méthodes traditionnelles d'apprentissage supervisé tel que : ils reçoivent des données en entrée, détectent les ***features*** de chacune d'entre elles, puis entraînent un classifieur dessus.il est l'un des modèles les plus répondu s'appuis sur des dizaine et des centaines des couche cachées a convolution de 2D .

L'architecture spécifique des CNN permet d'extraire automatiquement des caractéristique (***features***) de différentes complexités, des plus simples au plus sophistiquées. Ils constituent une évolution des réseaux de neurones artificiels traditionnels dont le motif de connexion entre les neurones est inspiré du ***cortex visuel des animaux***.

Aujourd'hui, le réseaux de neurone CNN , aussi appelé «**ConvNet** » pour Convolutional Neural Network, sont toujours les modèles les plus performants pour la classification d'images et ont donné d'excellents résultats pour le traitement automatique du language naturel (NLP) dans l' analyse syntaxique, sémantique en modélisation de phrases **[35][36].**

- Principe et fonctionnement

  Les reseaux de neurones CNN est basé sur la réduction du nombre des éléments structurelle (nombre de neurones artificiels) tel que :

- Ils reçoivent des images en entrée.
- Emploient des opérations de convolution (filtre) au moins une dans ses couches.
- Détectent les caractéristiques de chacune d'entre elles.
- Entrainent un classifieur sur ces données.

L'architecture des réseaux de neurone convolutive, se compose de 4 types de couche :

1- Couche convolutive :

C'est une couche critique qui consiste a appliques des défèrent **filtre de convolution** sur les images pour identifier et extraire leur caractéristiques , tel que chaque image passe par une succession des filtres (***noyaux de convolution***) permet de créer de nouvelles images appelé (**cartes de convolutions**), ainsi certain filtre peuvent appliqué une **opérations de maximum local** afin de réduisent la résolution de l' image .en fin de cette couche les **carte de convolution** sont miss a plat sous forme d'une vecteur des caractéristiques.

2- Couche de correction ReLU :

C'est la couche qui consiste à appliquer une fonction d'activation non linéaire (**Rectified Linear Unit**) sur **carte de convolution** résultantes dans la couche précédentes , en remplaçant les nombres négatifs des images filtrées par des zéros .

3- Couche de Pooling

Consiste à réduire le nombre des paramètres et des calcule dans le réseau en réduisant progressivement la taille spatial de la représentation (image, texte..) tel que en ne gardant que les informations les plus importantes et cela va permettre de contrôler le sur-apprentissage.

Plusieurs fonctions peuvent être appliquer dans cette couche selon la qualité des donné et les besoin cible , nous citons la plus populaire qui prend la valeur maximal des nombre « **Max Pooling** » ou la moyenne « **AVG Pooling » .**

4- Couche Fully Connected (entierement connectée):

C'est une couche attachée a la fin du réseau sous forme d'un **Perceptron** où **MLP** (Multi Layer Perceptron). Elle reçoit un vecteur en entré contenant les caractéristiques filtré, corrigé et réduit par la couche **pooling .**

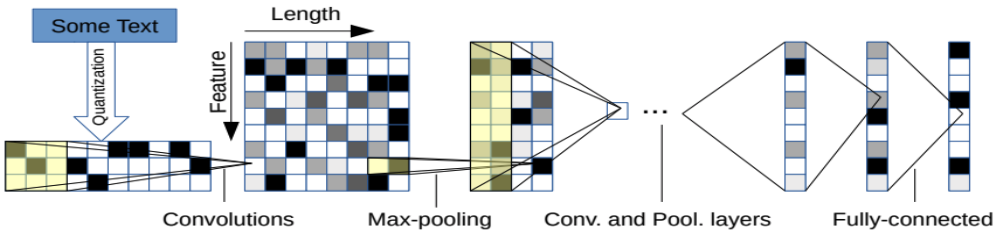

Figure. 1.7 Réseau de neurone Convolutif Multicouche[37]

1.4.5.2 Réseau de neurones Récurrents (RNN)

C'est un réseau de neurone plus proche du vrai fonctionnement du système nerveux, tel que u compris des couche profondes aux premières couches dans lesquelles l'information peut propager dans les deux sens .il s'appelle **récurrent** car il exécute la même tâche pour chaque élément d'une séquence, la sortie étant dépendante des calculs précédents.

Les RNN possède des connexions récurrentes au sens où elles conservent des informations en « **mémoire a court terme** », cette dernière peuvent prendre en compte à un instant (T) un certain nombre d'états passés. Pour cette raison, les RNNs sont particulièrement adaptés au traitement des séquences temporelles comme l'apprentissage et la génération de signaux, c'est à dire quand les données forment une suite et ne sont pas indépendantes les unes des autres (information séquentielle) **[38].**

- Principe et fonctionnement

Pour apprendre le fonctionnement des réseaux RNN , nous allons prendre l'exemple présenté dans la figure ci-dessus (**Figure. 2.9**). Si la séquence qui nous intéresse est une phrase de 3 mots, le réseau serait déroulé en un réseau de neurones de 3 couches, une couche pour chaque mot **[39] .**

Les formules qui régissent les calculs dans un RNN sont les suivantes :

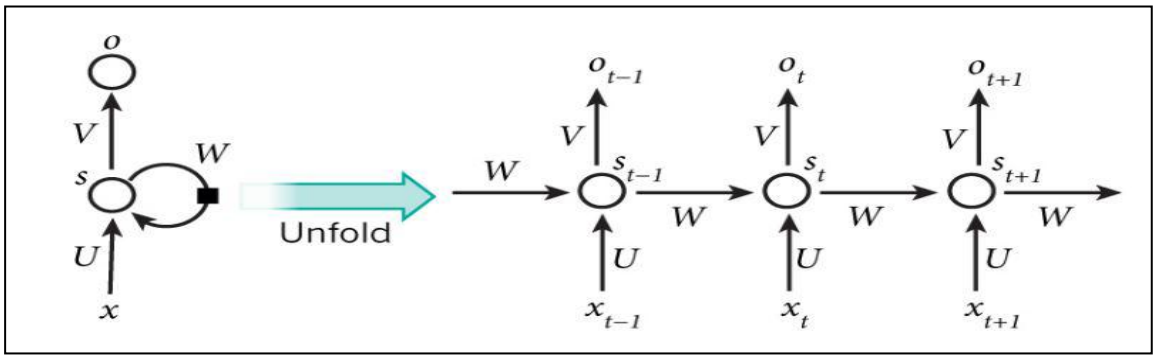

Figure. 1.8 Réseau de neurone Récurent dérouler [39]

- **X_t** : est l'entrée a un instant **t**.
- **U**, V, W : sont les paramètres que le réseau va apprendre des données de l'apprentissage.
- **S_t** : est l'état caché au moment **t**. correspond au « **mémoire** » du réseau.
- **S_t** : est calculé en fonction de l'état caché précédent et de l'entrée à l'étape actuelle
- ***f*** : est une fonction non linéaire telle que : **ReLu** ou **tanh**. $S_t = f(Ux_t + Ws_{t-1})$
- **O_t** : est la sortie au moment **t**.

Par exemple, si on veut prédire le prochain mot dans une phrase, ce serait un vecteur de probabilités dans un vocabulaire. $O_t = softmax(Vs_t)$

Problème : « Les mémoire à court terme » dans les réseaux de neurones récurent « classique » n'est pas suffisante pour résoudre les problèmes de la reconnaissance vocal et la génération des textes. Ils ne sont capables de mémoriser que le passé dit proche, et commencent à « oublier » au bout d'une cinquantaine d'itérations environ. Ce transfert d'information à double sens rend leur entrainement beaucoup plus compliqué, et ce n'est que récemment que des méthodes efficaces ont été mises au point comme les LSTM (Long Short Term Memory).

1.4.6 Domaines d'application

- **Conduite automatisée** : la reconnaissance des chaussées et des obstacles.
- **Électronique :** utilisé pour la reconnaissance audio et vocale,
- **La reconnaissance faciale** : reconnaissance des yeux, le nez, la bouche ...etc.
- **Reconnaissance d'image :** reconnaître les personnes dans les images.
- **Analyse des sentiments du texte :** évaluer la polarité sentimentale.
- **Assurance :** estimation de la gravité des sinistres à partir de photographies.
- **Le traitement automatique de langage naturel .**
- **Traduction automatique :** Traduction automatique de texte.

1.5 Conclusion

La classification automatique de textes est un domaine de recherche rendu très actif par la multiplication du nombre de documents numériques actuellement disponibles.

Dans ce chapitre, on a représenté dans la première partie la classification de texte et l'Apprentissage automatique (ML) d'une façon brève et précise. On s'est étalé sur le Processus de classification de textes, problèmes liés, domaine d'application et des Algorithmes de classification les plus fréquemment utilisés. Ensuite dans la deuxième partie, on a présenté les notions les plus importantes autour de l'apprentissage profond (DL), où on a présenté une introduction de ce dernier, les déférents type et modèles d'apprentissage profond les plus fréquemment utilisés pour la classification. En plus, nous avons décri une présentation des déférents domaines d'application.

Dans le prochain chapitre, nous allons détailler les déférents méthodes de transformation de donné textuel (codage de textes) utilisé dans le processus de classification de textes.

Chapitre II

Codage des textes

Chapitre II

Codage des textes

2.1 Introduction

Les dix dernières années ont produit une masse énorme de données, et engendré une surabondance d'informations textuelles, notamment publiées et facilement accessibles sur le réseau mondial. (90% de l'information accessible : bibliothèques électroniques, pages HTML, forums de discussion, etc.). L'analyse et le traitement d'informations textuelles sont devenus un enjeu majeur. Cependant, les tâches d'exploration et de récupération de l'information dans ces réservoirs de connaissances deviennent extrêmement difficiles. Ainsi, avec les bases de données multimédia, les dépêches d'agences de presse, les Publications scientifiques, etc... Qui sont consultés habituellement sur le réseau, on dispose de plus en plus de grandes masses de documents non ou faiblement structurées, représenté et stocker sous déférents formats de codage comme (HTML, XML, DOC, PDF). Face au problème de non structuration au sein des collections des données volumineuse, l'accès à l'information deviennent difficiles, d'où la nécessité de trouvé des méthodes de structuration automatique des corpus pour les rendre exploitables et utilisable par les différentes techniques et notamment par les algorithmes d'apprentissage automatique. Cette structuration que l'on nomme aussi la représentation numérique ou le codage des textes, vise a facilité la transformation de documents dans un autre espace et de préparer ces dernières à l'informatisation.

Dans ce chapitre le travail sera axé sur la présentation de déférentes approches de codage des documents textuels, les méthodes de représentation des documents, et on donne par la suite quelques méthodes de réduction de la dimensionnalité.

2.2 Linguistique descriptive

Une phase de prétraitement (transformation) doit être effectuée sur les textes exploitable par le processus de catégorisation, afin d'extraire le sens de chacun des textes. La majorité des approches repose sur l'idée que l'information du sens d'un texte est portée par un ensemble d'unités linguistiques. L'unité linguistique peut être issue d'une **Tokeninsation**, **lemmatisation** ou d'une **analyse** plus ou moins complexe.

Avant de détailler ces prétraitements, il faut étudier les bases de la linguistique descriptive pour mieux comprendre les différents prétraitements linguistiques. La linguistique correspond à une description de la langue qui peut donner une représentation sous forme de grammaire.

Dans notre cas de représentation textuelle, nous ne traitant que la linguistique descriptive, celle qui cherche à isoler les caractéristiques structurelles de la langue afin d'en déterminer son fonctionnement par l'intermédiaire de règles. Elle se décompose en trois tendances :

- La morphologie ou l'étude des variations (flexion, dérivation, composition),
- La syntaxe ou l'étude de la relation des mots dans la phrase,
- La sémantique et la pragmatique, ou l'étude de la signification des mots et du sens de la phrase.

L'étude de la langue repose sur le concept d'unité linguistique. Les unités linguistiques les plus évidentes sont les mots. Mais ces derniers ne sont pas suffisamment représentatifs. Un mot peut être variable et/ou ambigu. De plus, l'importance sémantique de chaque unité linguistique est différente en fonction de son rôle syntaxique dans la phrase. A partir de ce

Constat, un certain nombre de prétraitement linguistique peuvent être mis en œuvre pour extraire les unités linguistiques les plus pertinentes **[40] .**

2.3 Prétraitement

Dans la catégorisation, l'objectif de la comparaison est de mettre en évidence une notion de proximité entre deux textes, ou entre un texte et une catégorie donnée. Tandis

que, son principe est une comparaison sémantique. Il est nécessaire de représenter chacun des textes dans un formalisme permettant de modéliser cette « sémantique ». Il est souvent admis que le sens d'un texte peut être porté par un ensemble d'unités linguistiques particulières **[41].** Cependant, même si l'association entre sens et unité linguistique semble aléatoire, elle reste quand même une base solide, même si elle est incomplète quant à la représentation du sens. Les expérimentations sur le corpus Reuters 21578 **[42] [9] [40],** la preuve.

2.3.1 Normalisation en Minuscule

C'est une étape qui consiste à unifier les écritures des documents en lettre majuscule ou lettre minuscule. Nous pouvons appliquer cette étape avant ou après les opérations de nettoyage indiqué ci-dessous. Cette technique possède des inconvénients et notamment le risque de perte de sens ou d'ambigüité.

2.3.2 Tokenisation (Segmentation)

C'est une étape qui permet de découper et représenter un document par un ensemble de mots, termes (**Token**) à partir de l'ensemble d'entités linguistiques qui sont apparus dans ce document. Elle permet de supprimer les mots inutiles et isoler les ponctuations en utilisant les techniques de reconnaissance des débuts et fin des phrase, c.-à-d. découper les séquences de caractères en fonction de la présence ou l'absence de caractères de séparation (espace, tabulation ou retour à la ligne). Puis regrouper les chiffres pour former les nombres (reconnaissance éventuelle des dates) et aussi de reconnaitre les mots composés.

Pour notre travaille, la librairie **Keras Tokeniser** offre une bibliothèque des caractères spéciaux, nombre et signe de ponctuation.

```
Tokenizer (filters=' !"#$%&() *+0123456789,-/:;<=>?@[\\]^_`{|}~\t\n')
```

2.3.3 Élimination des mots vides (en anglais : Stop Words)

Avant d'effectuer un des prétraitements, il est préférable d'utiliser une «stop-list». L'objectif de la «stop-list» est d'éliminer tous les mots vides (stop words) ne participant pas activement à la sémantique du texte. La stop-list est une liste de tous les mots vides (pronoms, articles, préposition etc.), et les mots trop fréquents qui ne sont pas discriminants. L'inconvénient de la «stop-list» est qu'il est difficile d'en connaître à priori,

la taille optimale. De plus, la «stop-list» reste dépendante d'une langue donnée. Cette phase joue un rôle important dans la réduction de la taille des vecteurs et également elle diminue le taux d'erreur du classificateur.

Adjectif	Difficile, coûteux, précieux	**Prépositions**	à, de, pour, sur, dans, avec,
Verbes fonctionnels	Levez-vous, allez, voulez, faites, sortez, ayez, montrez,	**Articles & pronoms relatifs**	le, la, les, l', un, une, des, je, tu, il, elle, on,qui,que,quoi,
Noms payes	France,Algerie,Canada…..	**Noms personne**	Jack,nilson,jackson
Noms abrégés	FBI, EU , USA, FIFA….	**Dates**	22/05/2007
Chiffres isolés	25415, 9, 5, 2,	**Auteurs**	André Gide,
Références	N°= 15D 584U225	**Pourcentages**	100%, 24%, 2%
Sociétés	Shlumberger,Google,	**Journaux**	Reuters,newyork times
Quantités	Milliard, Kilograme,….	**adverbes**	ailleurs ,autour,avant,dedans,

Figure 2.1 L'aspect de la « stop liste »

2.3.4 Dé suffixation (Stemming)

Cette phase consiste à regrouper tous les termes qui peuvent avoir la même racine lexicale pour résoudre les difficultés de sélection des termes pertinents. On remplace tous les termes portant la même racine par leur forme originale, en éliminant tous les préfixes et les suffixe des mots. Par exemple les mot « likes» « liking » « liking » peuvent regrouper sous un même racine en supprimant leur suffixe au même terme « like » , mais dans certain cas cette méthode présente une précession et qualité inferieur , du fait qu'elle ne gère que les règles principales et ne peut pas prendre en compte les nombreuses exceptions des règles de dérivations . Comme exemple en langue français les termes « *fraise* » et « *frais* » sont des mots du peuvent transformer au même terme « *frais* » alors qu'il n'existe pas une relation entre eux, et également sa pose un problème de perte de sens. Par ailleurs, cette opération nécessite une adaptation pour chaque langue utilisée. La désuffixation est une méthode de traitement linguistique moins compliqué que la lemmatisation, tel que, il utilise des algorithmes simple et plus rapide que la lemmatisation et ne faisant référence aux dictionnaires et règles de dérivation). Parmi les algorithmes de stemming qui sont développé pour déterminer les racines lexicales, nous citons l'algorithme le plus couramment utilisé dans la langue anglaise est celle de PORTER définit dans **[43]**.

2.3.5 La lemmatisation

L'approche « sac de mots » est de moins en moins utilisée car elle pose un certain nombre de problèmes. Vu l'existence d'un nombre important de mots, l'association d'un

sens à chacun des mots est inexacte. En effet, beaucoup de mots ont des racines communes. Les premières unités linguistiques à s'être imposées comme représentatives du sens sont les radical ou les lemmes, où la reconnaissance de ces unités linguistiques nécessite d'effectuer un pré- traitement linguistique pour la recherche des lemmes. La technique de lemmatisation cherche à résoudre la difficulté de la dé suffixation (ou stemming), qui consiste à rechercher les racines lexicales. Elle consiste à remplacer les verbes par leur forme infinitive, et les noms par leur forme au singulier en regroupant sous le même terme tous les mots de la même famille. L'implémentation la plus connue et efficace est celle de l'algorithme nommé TreeTagger (Schmid, 1994) a été développé pour les langues anglaise, française, allemande et italienne. Cet algorithme utilise des arbres de décision pour effectuer l'analyse grammaticale, puis des fichiers de paramètres spécifiques à chaque langue.

Verbe	***Infinitif***:
Accepted, cleared	Accept ,Clear
Nom :	***singulier*** :
Books Buses Cities	Book Bus City

Figure 2.2 Exemple de lemmatisation

2.4 La représentation des textes

Le formalisme le plus utilisé dans la représentation des textes est le vectoriel. Le modèle standard est issu de **[42]** dont l'implémentation la plus connue est SMART. Dans ce formalisme, chaque dimension de l'espace vectoriel correspond à un segment textuel que l'on nomme terme d'indexation, préalablement extrait du jeu d'apprentissage. Un segment textuel représentera donc indépendamment un mot, un paragraphe, une lettre ou d'autres assemblages de lettres.

Le rôle de la représentation textuelle est de projeter les textes au sein d'une représentation mathématique, tout en conservant au maximum la sémantique de ceux-ci, en utilisant un espace vectoriel comme espace de représentation cible.

La caractéristique principale de la représentation vectorielle est que chaque segment textuel est associé à une dimension propre au sein de l'espace vectoriel.

2.4.1 Techniques de représentation

2.4.1.1 La représentation en « sacs de mots » « BOW » « Bag Of Word » :

Dans cette représentation, l'unité linguistique choisie comme étant un élément représentatif du sens est le mot. Chaque mot est en général extrait sur la base d'un ensemble de séparateurs tel que l'espace, la virgule, le point, etc. La représentation par « **sac de mots** » est difficilement utilisable telle quelle. En effet, le nombre d'unités linguistiques générées peut être volumineux. Un « filtrage » est utilisé, afin de ne conserver que les unités linguistiques pertinentes.

Doc	Il déteste les tartes aux pommes						Kamel déteste les œufs			
Termes	Il	déteste	les	tartes	aux	pommes	Kamel	déteste	les	Œufs

Vocab	Il	déteste	les	tartes	aux	pommes	Kamel	Œufs
Doc1	1	1	1	1	1	1	0	0
Doc2	0	1	1	0	0	0	1	1

Figure 2.3 Exemple de l'approche « Sac de mot »

2.4.1.2 La représentation par phrases : Cette approche utilise des groupes de mots (des phrases), comme élément représentatif du sens. L'intérêt d'utiliser une phrase c'est que :

- La phrase possède une unité de sens plus complète qu'un simple mot **[44].**
- La phrase, même dans le désordre, offre d'avantage d'informations sur le champ sémantique dans lequel on se trouve.
- La phrase définit une certaine relation d'ordre entre les mots.
- La phrase offre une certaine gestion des collocations.

2.4.1.3 La représentation â base de n-grammes : Les n-grammes ont été introduit par [Sha48] en 1948. En général, le n-gramme se définit comme étant une séquence de n caractères consécutifs. Le mot « exemple » est composé des n-grammes suivants :

- Bi-grammes « _e », «ex», «xe», «em», «mp», «pl», «le», «e_».
- Tri-grammes « _ _e », «_ex», «exe», «xem»,«emp»,«mpl»,«ple»,«le_»,«e_ _ ».
- 4-grammes « _ _ _e »,«_ _ex»,«_exe»,«exem»,«xemp»,«empl»,«mple»,«ple_», «le_ _», «e_ _ _».
- De manière générale pour une chaîne de k caractères entourée de blancs, on génère k + 1 n-grammes .

2.4.1.4 Représentation distribuée : « *Word Embedding* »

- Les Word Embeddings actuellement les plus populaires dans la littérature sont Fournis par la boite à outils ***word2vec,***
- **Word2vec** est une technique de plongement de mots (en anglais : word embedding) introduite par Mikolov et al. (2013) ; Cette technique focalise sur l'apprentissage d'une représentation de mots. Ou chaque mot est représenté par un vecteur.
- La méthode **Word2vec** constitue une projection des mots du vocabulaire dans un espace de faible dimension de manière à préserver les similarités sémantiques et syntaxiques. Ainsi, si les vecteurs de mots sont proches les uns des autres en termes de distance, les mots doivent être sémantiquement ou syntaxiquement proches.
- Chaque dimension représente une caractéristique latente du mot, qui peut capter des propriétés syntaxiques et sémantiques.
- **Word2vec** a proposé d'abord deux architectures (**CBOW et Skip- gram**) pour l'apprentissage des *word embeddings* qui sont moins coûteuses en termes de temps de calcul **[45].**

2.5 La Pondération des termes

. Dans **[46],** les auteurs évaluent l'importance d'un terme d'indexation au sein d'un texte selon trois critères la pondération locale, la pondération globale et la longueur du texte.

2.5.1 Pondération locale : La pondération locale prend en compte l'information liée à la fréquence d'occurrence d'un terme d'indexation, localement à un texte. Les facteurs les plus utilisés sont :

a) **Le facteur binaire** : Ce facteur vaut « **1** » si le terme d'indexation est présent et « **0 »**, s'il ne l'est pas. La représentation binaire présence/ absence est utilisée avec des mesures de type ensembliste comme mesure de similarité.

b) **Le facteur TF** : Il correspond à la fréquence d'un terme d'indexation au sein d'un texte.

c) **Le facteur logarithmique :** Il correspond à une variante du facteur TF et vaut (***1+log (TF (w_K, dn)),*** avec w_k un terme d'indexation et d_n un texte. Ce facteur proposé par **[17]** part du constat qu'un texte qui contient un

grand nombre de fois un terme de la requête, n'est pas forcément plus pertinent qu'un texte qui ne contient qu'un petit nombre de fois plusieurs termes de la requête.

d) Le facteur TF augmenté : Ce facteur, vise à réduire l'influence que peut causer un grand nombre d'occurrences d'un même terme sur une mesure de pertinence **[2].** Ce facteur vaut (Voir formule 1) :

$$0.5 + 0.5 * \frac{TF(w_k, d_n)}{\max_i TF(t_i, d_n)} \quad \text{…….. (1)}$$

2.5.2 Pondération globale : La Pondération globale prend en compte l'information liée à la fréquence d'occurrence d'un terme d'indexation. Les facteurs les plus utilisés sont :

a) Le facteur IDF : Il correspond à l'inverse de la fréquence en textes d'un terme d'indexation et vaut : $\log(\frac{|D|}{DF(w_k)})$, avec $|D|$ le nombre de textes du jeu d'entraînement et DF (w_k) le nombre de textes où le terme d'indexation w_k est présent.

b) Le facteur IDF probabiliste : ce facteur est une variante du facteur IDF et se calcule de la manière suivante : $\log(\frac{|D| - DF(w_k)}{DF(w_k)})$ On remarquera qu'il possède les mêmes caractéristiques que le facteur IDF.

2.5.3 Pondération TF-IDF

Le codage TF-IDF a été introduit dans le cadre de représentation vectorielle en prendre en compte la loi de Zipfs .son principe de base est que le vecteur qui représente le document se calcule en donnons aux termes les deux importance decrites ci-dessous , l'une correspond a l'importance local exprimé par l'importance du terme dans le document (Term Frequency : TF) et l'autre correspond a l'importance globale exprimé par l'importance du terme dans le corpus (Inverse Document Frequency : IDF) .
La multiplication de ces deux notion permet d'attribuer un poids d'autant plus fort que le terme apparaît souvent dans le document et rarement dans le corpus complet.(Formule 2)

*Term Frequency * Inverse Document Frequency* …….. (2)

Le codage TF-IDF peut représenter par la formule suivante (Formule 3) :

*Poid (d,wi)=TF*IDF =TF(d,wi)*log(Nbr_Doc / DF(wi))* …….. (3)

- *TF(d,wi) :*représente le nombre des termes dans le document.
- *Nbr_Doc :* nombre complet des textes dans le corpus.
- *DF(wi) :* nombre des documents qui contient le mot wi .

2.6 La réduction de dimensionnalité

La taille de l'espace vectoriel dépend directement du nombre de segments textuels différents extraits lors de l'étape de prétraitement. Pour cela on a introduit la notion de la réduction qui a un double objectif :

- Elle permet d'éviter le phénomène d'apprentissage par cœur **[9].**
- Elle améliorer l'efficacité des algorithmes d'apprentissage ayant des difficultés à gérer un nombre important de termes d'indexation.

La réduction de la dimension peut se voir ainsi comme :

2.6.1 Extraction de termes : consiste à générer les termes d'indexation à partir d'une combinaison ou d'une transformation des segments textuels.

2.6.2 Sélection de termes : consiste à choisir comme terme d'indexation (terme utile) un sous-ensemble de segments textuels extraits lors de l'étape de prétraitement, selon un critère fixé préalablement tandis que les autres sont rejetés.

Pour identifier les meilleurs descripteurs, il est nécessaire d'utiliser une méthode de Sélection de termes. En effet, il existe plusieurs méthodes de sélection de termes :

2.6.2.1 ***Fréquence en document***. Elle ne correspond pas à la notion de fréquence au sens mathématique, mais à la fréquence d'apparition d'un segment textuel. La fréquence en document d'un segment textuel correspond au nombre de documents où le segment textuel est présent au moins une fois. Dans **[42]** les auteurs montrent qu'une réduction de l'espace par un facteur de 10 n'entraîne aucune perte de performance. C'est seulement à partir d'un facteur de 100 que l'on constate une perte sensible de performance.

2.6.2.2 ***La loi de Zipf.*** La fréquence des mots ainsi que leur distribution au sein d'un corpus n'est pas uniforme. Certains mots sont très fréquents alors que d'autres n'apparaissent que très rarement. Les mots vides ont tendance à être très fréquents quel que soit le corpus utilisé.

3.6.2.3 ***Term Clustering*** **:** La méthode par regroupement de termes consiste à regrouper les termes d'indexation ayant une « proximité sémantique » élevée. L'idée sous-

jacente est que seul le groupe représentatif d'un ensemble de termes d'indexation est associé à une dimension de l'espace vectoriel.

2.6.2.4 ***Mesures provenant de la théorie de l'information*** : Des mesures issues de la théorie de l'information sont aussi utilisées pour éliminer les segments textuels trop peu discriminants. Tel que :

- ✓ L'information mutuelle (MI).
- ✓ Le gain d'information (IG) **[44].**
- ✓ Chi-deux (x2) **[40].**
- ✓ L'«odds ratio» **[7] [41] [43].**
- ✓ Le facteur d'association (DIA) **[7].**

Nous allons présenter dans la section suivante la méthode Gain d'Information, utilisé pour notre expérience comparative des déférentes techniques de sélection des caractéristiques.

- **Le Gain d'Information** (Information Gain : IG) consiste à compter et mesurer la quantité et le nombre des bits d'information obtenus pour la prédiction d'une classe (c) en connaissant la présence ou l'absence d'un terme (t) dans un document. En d'autres termes le GI est une mesure de la fréquence des termes dans une classe particulière par rapport à sa fréquence dans toutes les autres classes **[47][48]**. Le gain d'information du terme t peut se calculer comme suite (

$$GI_t = e(p,n)[P_w\, e(t_p, f_p) + P_w^- e(f_n, t_n)] \quad (4)$$

.....

Où
- p, n : représente le nombre positive et négative des instance (doc)
- $e(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$
- $P_w = \frac{(tp+fp)}{N}$; $P_w^- = 1 - P_{term}$

- **L'entropie** : selon [shannone1949] : exprime la quantité d'information, c'est-à-dire le nombre des bites nécessaire pour représenter une information. L'entropie est la mesure indiquant le désordre et la distribution uniforme d'un ensemble d'exemple **S** par rapport à une classification **[49]** .

L'entropie d'un ensemble d'exemple **S** représenté par la formule suivante (Voir formule 5) :

$$\text{Entropie}(S) = H(S) = -\sum_{i=1..k}^{c} Pi * \log_2 Pi \quad \text{.....(5)}$$

Ou *Pi* est la proportion d'exemple de catégorie **i** dans **S**

2.7 Conclusion

La catégorisation automatique de textes est un domaine de recherche rendu très actif par la multiplication du nombre de documents numériques actuellement disponibles. Dans ce chapitre, on a abordé l'étape de représentation textuelle qui consiste à transformer les documents initiaux de façon à ce qu'ils puissent être exploitables par un algorithme d'apprentissage. Nous avons vu les déférentes méthodes de codage, de représentation des documents et ainsi que les techniques de réduction de la dimensionnalité. Dans le chapitre suivant, nous allons présenter notre travail réalisé tout en donnant une vue générale sur l'architecture de notre système. Par la suite on montre les détails concernant l'utilisation des algorithmes de Deep Learning pour la classification des données textuelles.

Chapitre III

Réalisation

Chapitre III

Réalisation

3.1 Introduction

Tout processus de catégorisation automatique de documents textuels peut se diviser en trois étapes principales qui sont : *l'étape de Codage des textes*, *la classification automatique et finalement l'étape d'évaluation*. Les approches modernes utilisent généralement plusieurs tâches pour le codage des documents textuels. Cette dernière comporte deux phases : la première qui est le prétraitement ou le nettoyage de texte. Elle permet aussi à la représentation précise des documents (ex : Tokenisation, filtrage des mots vides (stop words), lemmatisation, stemming, etc.) **[50]**. Quant à la deuxième phase, il s'agit de la sélection/réduction de l'espace des caractéristiques avec un impact négatif et également minimal sur la précision de la classification **[51].** Les techniques d'apprentissage profond (*Deep Learning*) sont aujourd'hui très populaires dans le domaine d'apprentissage automatique où il a été prouvé leur performance dans les domaines de traitement d'image et particulièrement dans le traitement du langage naturel (NLP), notamment dans la traduction automatique, la segmentation, la reconnaissance du parole, l'analyse des sentiments et la classification des textes.

Après avoir étudié dans le chapitre 01 un état de l'art sur les différentes techniques d'apprentissage profond (Deep Learning). Nous allons présenter dans cette partie un nouveau modèle de CNN qui est à la fois proposé et adapté pour la classification textuelle. Une comparaison est introduite tout en présentant les résultats de notre modèle **CNN Kim Modifié** par rapport aux méthodes d'apprentissage automatique classique à savoir **SVM, Naive Bayes, Random Forrest et les Arbres de décision**. Ensuite, une autre comparaison entre les déférentes techniques de sélection des caractéristiques notamment **Back Propagation** et **Gain d'Information est présentée en détail dans ce travail. Cette dernière sert** à montrer l'utilité de l'algorithme de **BackPropagation** et notamment son impact que ce soit sur les performances de classification ou sur la sélection des caractéristiques. Cette partie fournit aussi une description sur les Benchmarcks utilisées dans cette expérimentation, les outils appliqués et bien évidemment elle décrit les résultats des tests réalisés et leurs discutions.

3.2 DataSets

Afin de pouvoir évaluer les performances des déférentes méthodes de classification, notre travail sera fondé sur un corpus basé sur les articles de presse (**Reuters 21578-Top10**) en utilisant La répartition **« ModApte »** vers des ensembles d'apprentissage **(75%)** et du teste **(25%)** compilé par **David Lewis**. Toutes les classes qui ont un seul document sont supprimées avec les documents correspondants. L'ensemble d'entrainement résultant contient **7769 documents** et l'ensemble de tests **3019 documents**. Après avoir traité les documents et supprimé les symboles, la ponctuation, Stop Word et les mots qui n'apparaissent que dans un seul document, l'ensemble des termes (**tokens**) résultant constitue un vocabulaire spécialisé de taille **15715 mots** lié à plusieurs domaines. Pour implémenter une classification multiple, nous avons choisir d'évaluer seulement les **10 catégories** pour lesquelles il existe un nombre suffisant d'exemples de documents d'entrainements et de tests **[52].**

Dans cette étude nous étudions une classification multiple qui consiste à appliquer plusieurs classificateurs binaires distincts (classificateur binaire par classe), nous avons divisé les documents en 2 groupes pour chaque catégorie **: un** groupe étiqueté « Oui », il s'agit d'un document qui appartient à la catégorie et par « Non » dans le cas contraire. La répartition des documents (**train, test**) pour chaque catégorie est présentée dans le tableau. **(Tableau 4.2).**

Enfin, pour chaque classe, seuls **256 mots**, descripteurs (pertinents) qui peuvent représenter les documents (mesurés par le gain d'informations **GI**) sont sélectionnés comme des caractéristiques de l'ensemble de données textuel.

Tableau 3.1- la répartition de la collection Reuters (10 classes) selon ModeApte (train-test)

Category	**Train**		**Test**	
	Yes	**No**	**Yes**	**No**
Acq	1615	6154	719	2300
Corn	175	7594	56	2963
Crude	383	7386	189	2830
Earn	2817	4952	1087	1932
Grain	422	7347	149	2870
Interest	343	7426	131	2888
Money-fx	518	7251	179	2840
Ship	187	7582	89	2930
Trade	356	7413	117	2902
Wheat	206	7563	71	2948
Total	**7769**		**3019**	

Les documents du corpus **Reuters** sont répartis par étiquète comme illustré dans la figure (**Figure. 3.1**).

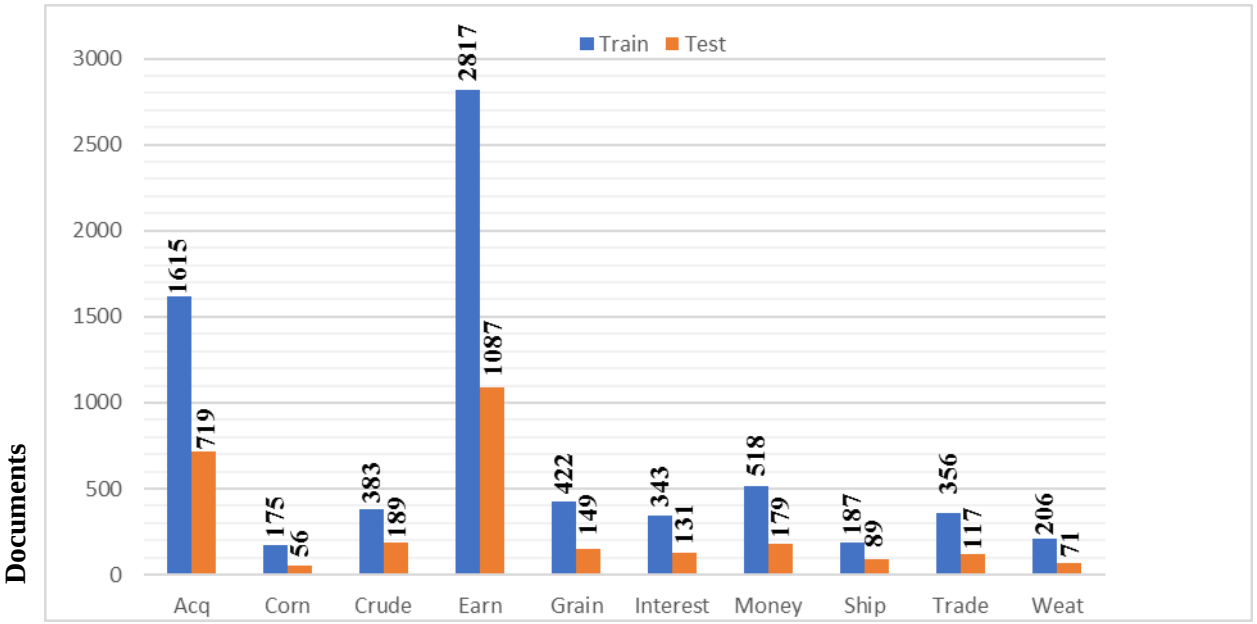

Figure. 3.1. Distribution des documents par étiquette de la collection Reuters.

3.3 Outils :

Pour réaliser ce travail, nous avons basé sur un ensemble des outils, logiciels et également matériels. Dans les sous sections suivantes nous donnons une petite description de chacun d'eux.

3.3.1 Logiciel utilisé

En bas on montre une liste des logiciels, API et Bibliothèques qui sont utilisé ç la réalisation de notre travail.

- **Weka [53]** : est un package Open Source très populaire. Il implante différents algorithmes de data-mining et d'apprentissage (Naïve Bayes, Arbre de décision, SVM, réseau de neurones, entre autres). Il offre une interface GUI conviviale pour manipuler et inspecter les données et visualiser les résultats. Ce package peut fonctionner sur les plateformes Linux, Windows et Mac.
- **TensorFlow [54] :** est une framework Open Source de programmation développée pour le calcul numérique par l'équipe Google Brain en Novembre 2015 qui l'utilisait initialement en interne. Depuis son release, TensorFlow n'a cessé de gagner en popularité, pour devenir très rapidement l'un des frameworks les plus utilisés pour le Deep Learning et donc basé sur le principe des réseaux de neurones profond. Son nom

est notamment inspiré du fait que les opérations courantes sur des réseaux de neurones sont principalement faites via des tables de données multi-dimensionnelles, appelées Tenseurs (Tensor). Un Tensor à deux dimensions est l'équivalent d'une matrice. Aujourd'hui, les principaux produits de Google sont basés sur TensorFlow: Gmail, Google Photos, Reconnaissance de voix.

- **Keras [55] :** est une API de réseaux de neurones de haut niveau, écrite en Python et capable de fonctionner sur TensorFlow ou Theano. Il a été développé en mettant l'accent sur l'expérimentation rapide. Il a été développé dans le cadre de l'effort de recherche du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), et son principal auteur et mainteneur est François Chollet, un ingénieur Google. En 2017, l'équipe TensorFlow de Google a décidé de soutenir Keras dans la bibliothèque principale de TensorFlow. Chollet a expliqué que Keras a été conçue comme une interface plutôt que comme un cadre d'apprentissage de bout en bout. Il présente un ensemble d'abstractions de niveau supérieur et plus intuitif qui facilitent la configuration des réseaux neuronaux indépendamment de la bibliothèque informatique de backend. Microsoft travaille également à ajouter un backend CNTK à Keras aussi.
- **Python [56] :** est un langage de programmation de haut niveau, interprété (il n'y a pas d'étape de compilation) car, avant de pouvoir les exécuter, un logiciel spécialisé se charge de transformer le code du programme en langage machine, orienté objet avec une sémantique dynamique, multiparadigme et multiplateformes. Le langage python est placé sous une licence libre, et fonctionne sur la plupart des plates-formes informatiques, des smartphones aux ordinateurs centraux. Il est très sollicité par une large communauté de développeurs et de programmeurs et Python peut être utilisé pour gérer des données volumineuses et effectuer des calculs complexes. Il existe ce qu'on appelle des bibliothèques (packages) qui aident le développeur à travailler sur des projets particuliers. Plusieurs bibliothèques peuvent ainsi être installées pour, par exemple, développer des interfaces graphiques en Python.
- **Scikit-learn [57] :** est une bibliothèque libre Python destiné à l'apprentissage automatique. Elle est développée par de nombreux contributeurs notamment dans le monde académique par des instituts français d'enseignement supérieur et de recherche comme Inria et Télécom ParisTech. Elle comprend notamment des fonctions pour estimer des forêts aléatoires, des régressions logistiques, des algorithmes de

classification, et les machines à vecteurs de support. Elle est conçue pour s'harmoniser avec des autres bibliothèques libre Python, notamment NumPy et SciPy.

3.3.2 Matériels utilisés

Dans ce qui suit, on présente les caractéristiques techniques de la machine utilisée dans les expérimentations.

- **Colaboratory [58]** : est un outil de recherche open source pour l'enseignement et la recherche en apprentissage machine. Il s'agit d'un environnement pour ordinateur portable **Jupyter** qui ne nécessite aucune installation ou bien une configuration pour être utilisé, il s'exécute entièrement dans le cloud. Cet environnement supporte **Python 2.0** et ses versions supérieures. Il nous permet d'écrire et d'exécuter du code, de sauvegarder et partager, et d'accéder à de puissantes ressources informatiques depuis notre navigateur. Nos expériences ont été réalisées sur une machine virtuelle dans l'environnement Google Colab qui offre une bonne performance dont les caractéristiques sont les suivantes (**Tableau 3.2**).

Tableau 3.2 – illustre la configuration du colab.

CPU	Intel Xeon CPU 2.30GHz
GPU	NVIDIA Tesla T4
RAM	12 GB

3.4 Une nouvelle Architecture CNN

Dans cette section, nous décrivons la topologie de réseaux CNN utilisées dans notre expérience. À noter que, nous avons utilisé le modèle CNN simple de ***Kim* 2014 dans [3]**.

3.4.1 le Modèle de Kim 2014 : c'est un modèle de Réseau de CNN d'apprentissage profond qui se composé de :

- Couche d'intégration des mots (Word Embeding) initialisé par un model pré-entrainé Word2vec.
- Couche convolutive (Conv 1D) se base sur des noyaux bidimensionnels (kernel) de taille *K x E* avec fonction d'activation ReLu.
- Couche de regroupement maximale (MaxPoll).

- Couche entièrement connectée (Fully Connected) ajouter avant la couche de sortie et une fonction d'activation Sigmoid (Classification binaire).

L'architecture du CNN appliquée dans cette expérience est une version modifiée de l'originale architecture du réseau de **Kim** et qui est sous forme d'un **modèle séquentiel** (pille des couches).

3.4.2 Un nouveau Modèle de CNN (Kim Modifié) : notre modèle proposé est conçu à partir du modèle original de CNN-Kim. Il se caractérise par les points suivants :

- Une couche convolutive (Conv 1D) se base sur des noyaux unidimensionnelle (kernel) de taille *K=2* avec fonction d'activation ReLu.
- une couche de regroupement maximale (MaxPoll).
- Une couche entièrement connectée (Fully Connected) ajouter avant la couche de sortie et une fonction d'activation SOFTMAX (Classification Multiple).

La figure 3.2 ci-dessus, affiche la partie de notre **Réseaux de Neurone Convolutive de Kim Modifié dans le processus global et** La figure 3.3 affiche les différentes couches selon leur ordre du modèle modifié.

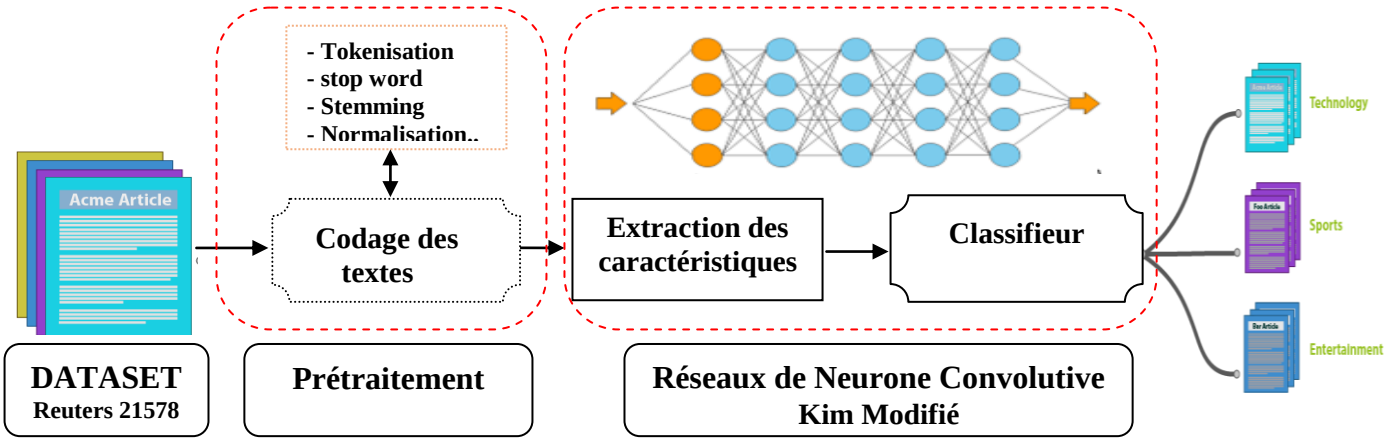

Figure. 3.2 Architecture du Classification des Textes avec CNN Proposé

Le tableau 3.3 au-dessus affiche les paramètres utilisés dans les expérimentations de notre Modèle de Réseau CNN (**Modèle Kim modifié**).

Tableau 3.3 – les paramètres du Model Kim modifié

Couches	Itération (époche)	Paramètre
Input	**20**	Quantité des termes pertinent = 256
Conv1D (ReLu)		Taille Filtre (*KxE*) =2 X 1 Nombre Filtre = 64
MaxPool Size		Taille MaxPool =Max du 2
Fully Connected		Taille Couche =64 neurones
Output (SoftMax)		Nombre Classe =10 classes
Dropout , Gradient (BP)		**Désactiver**

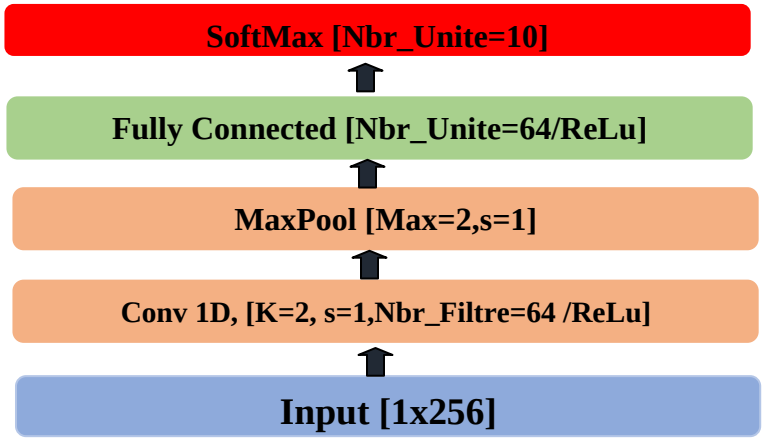

Figure. 3.3 Illustration du modèle de base CNN Kim Version Modifié (256 termes)

3.5. Expérimentation

Dans cette section, nous décrivons les différentes expériences effectuées dans notre travail avec une discussion sur les résultats obtenus. En plus, les performances de notre modèle Kim proposé sont comparées par rapport à ceux obtenus par les algorithmes d'apprentissage automatique notamment **SVM, Naive Bayes, Random Forrest et les Arbres de décision. Tout le présent travail a été réalisé dans l'environnement** de développement **Weka**. Par ailleurs, on a choisi la base de test **Reuters 21578 pour appliquer afin d'évaluer les performances des algorithmes de classification.**

3.5.1 Notre Stratégie de classification

✓ **Partie de Prétraitement des textes :**

L'entrée de la première couche de notre réseau est le résultat d'un processus de prétraitement des documents textuels. Ils existent plusieurs outils pour y faire, Dans notre cas, nous avons choisi l'outil **keras Tokenizer** et ses librairies afin de produire ces attributs finaux.

L'outil **keras Tokenizer** comporte les phases : Normalisation, filtrage et enfin la sélection de descripteur qui seront utilisées dans la couche d'Entrée de notre model **CNN Kim Modifié**. Les étapes effectuées dans cette partie sont :

- Conversion des majuscules en minuscule, élimination du ponctuation, chiffres et signes inutiles.
- Utilisé les librairies (**NLTK, SpaCy**) pour construire un vocabulaire général de tous les termes du corpus d'apprentissage (***790148 jetons).***
- Suppression des termes les plus rare, plus fréquent (stop words), stemming et lemmatisation (***15715 jetons).***
- Sélectionner **256 mots** comme des caractéristiques pour chaque catégorie.
- Construire une matrice binaire (Chaque document = un vecteur binaire).

✓ **Apprentissage et teste :**

Concernant la tache de classification. Cette dernière est réalisée dans l'ordre suivant :

- Entrainer notre classifieur proposé (CNN Kim Modifié) sur l'ensemble des données d'apprentissage.
- Tester la performance de notre classifieur en utilisant les donné du teste.
- Evaluer et comparer les résultats de notre classifieur par les mesures de performances calculées précédemment.

3.5.2 Résultat et discussion :

Nous avons traité chacune des 10 catégories (**ModApte**) comme une tâche de classification binaire et nous avons évalué les classificateurs séparément pour chaque catégorie. L'évaluation de notre modèle se base sur les 3 métriques qui sont (**Précision, Rappel, Mesure F1**). Les mesures de performance résultantes de notre modèle **CNN Kim Modifié** pour chaque catégorie sont résumées dans le **Tableau 3.4.**

Tableau 3.4 – illustre mesure F1 Moyen pour CNN Kim Modifié

Catégories	Nbr Features	F-measure
Corn	**256**	0,886956522
Crude		0,830687831
Earn		0,966139955
Grain		0,897959184
Interest		0,699186992
Money		0,656804734

Ship		0,863636364
Trade		0,684684685
Wheat		0,896103896
Acq		0,906444906
Average		**0,828860507**

Afin de valider notre model **CNN Kim Modifié** et évaluer sa performance de classification, Nous l'avons testé et comparé avec un ensemble d'algorithmes très reconnus en apprentissage automatique, particulièrement (NaiveBayes, J48, BayesNet, Random Forrest et SMO) **[59].**

3.5.3 Comparaison

3.5.3.1 CNN Kim Modifié vs Algo ML:

Dans cette expertise, Nous avons utilisé l'outille Weka pour tester, évalué et mettre en œuvre les algorithmes d'apprentissage automatique (Machine Learning). Le tableau 3.5 présente les résultats de performance (exprimer en F-mesure-Moyen) obtenu par les algorithmes d'apprentissage automatique (NaiveBayes, J48, BayesNet, Random Forrest et SMO) et ainsi avec ceux du model proposé CNN Kim Modifié.

Tableau 3.5 – résultat de performance F-mesure Moyen (F1) (CNN Kim Modifié vs Algo ML)

Classes	**BayesNet**	**J48**	**NaiveBayes**	**Random Forrest**	**SMO**	**CNN KIM Modifié**
Corn	0,422310757	0,9	0,254742547	0,744680851	0,778947	0,886956522
Crude	0,734607219	0,746356	0,507131537	0,837606838	0,824513	0,830687831
Earn	0,963910461	0,956444	0,798794273	0,982584785	0,977148	0,966139955
Grain	0,693586698	0,875	0,550308008	0,884210526	0,835821	0,897959184
interest	0,586894587	0,586667	0,452380952	0,660287081	0,549223	0,699186992
Money	0,568507157	0,637931	0,494505495	0,710843373	0,606667	0,656804734
Ship	0,787037037	0,7125	0,443243243	0,595419847	0,746667	0,863636364
Trade	0,468292683	0,677551	0,372943327	0,693467337	0,688372	0,684684685
Wheat	0,592274678	0,899329	0,343007916	0,827067669	0,784	0,896103896
Acq	0,901388889	0,860088	0,898672257	0,922746781	0,923952	0,906444906
Average	0,671881017	0,785187	0,511572956	0,785891509	0,771531	0,828860507

D'après les résultats du tableau (tableau 3.5**),** Nous avons constaté que tous les classificateurs classiques d'apprentissage automatique (ML) à savoir (SMO, J48, Random forrest, bayesNet) et ainsi que notre modèle CNN (Kim Modifié) sont les meilleurs et qui ont obtenu des bonnes résultats en matière de la mesure **F-mesure. Cela est juste** dans certaines

classes majoritaires comme (Earn, acq) et également dans certain cas dont les classes minoritaires comme c'est passé dans la catégorie (grain). Mais, en matière de **F-mesure moyen**, les résultats obtenus indiquent clairement que les performances de notre Modèle CNN Kim (Modifié) proposé dépasse considérablement tous les autres algorithmes d'apprentissage automatique classique quel que soit la taille des classes utilisées à partir du jeu de donnés. La **Figure 3.4** montre en bars les résultats. En line les classes. En colonne la moyenne de F-Mesure. .

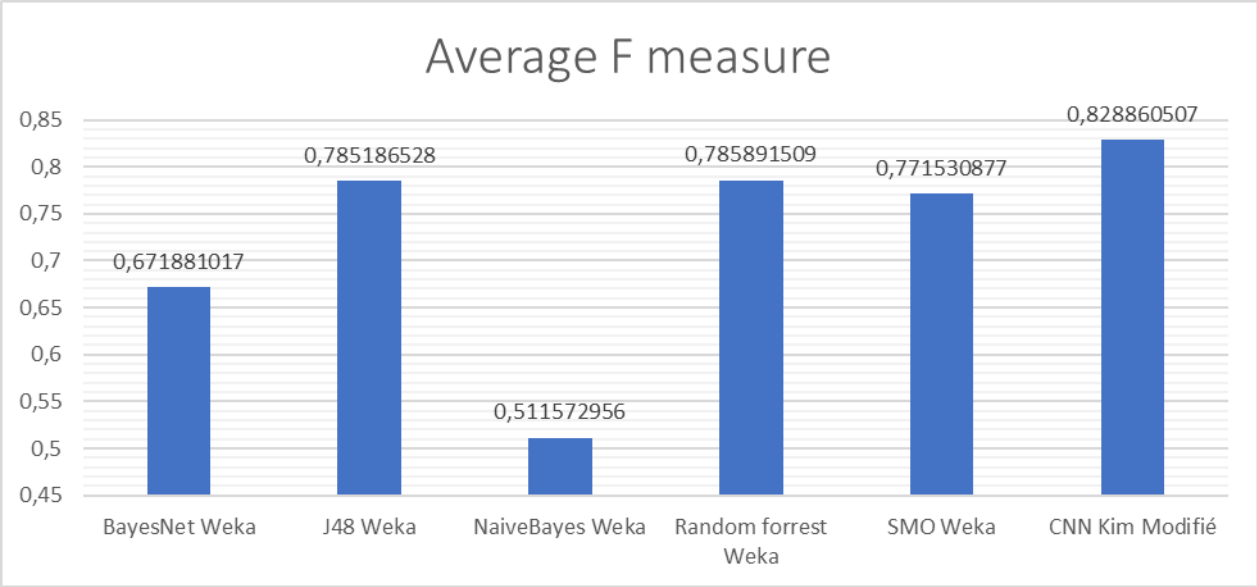

Figure. 3.4. Graphique Average F-measure /Classifieur

3.5.3.2 BP (CNN Kim Modifié) vs GI (Algo ML) :

Dans un premier temps et nous avons réalisé l'expérimentation d'un ensemble d'algorithmes ML ainsi avec notre CNN modifié de KIM. Les résultats montrent l'avance de notre model CNN Kim Modifié par rapport aux autres méthodes en matière des mesures précision, rappel et F-mesure. Dans un second temps, nous allons tester le rôle et l'impact de sélection des caractéristiques (termes pertinents) sur la précision et la performance des classificateurs. Pour cela, nous avons réalisé une deuxième expérience qui est complémentaire à la première partie.

Dans cette partie, Nous allons comparer les performances de notre modèle CNN Kim Modifié avec ceux des Algorithme d'apprentissage automatique classique en appliquant les deux méthodes de sélection des caractéristiques (**BackPropagation, Gain d'information**). En plus, Nous visons d'utiliser l'apprentissage par transfert (FineTuning) pour évaluer la performance de la méthode Backpropagation (**Figure 3.5**). de ce fait, les résultats obtenus par la technique BackPropagation seront aussi comparé avec ceux obtenus par l'autre technique

qui est le Gain D'information. Ce dernier (filtre IG) a était combiné avec les classificateurs classiques en s'appuyant sur l'outil Weka.

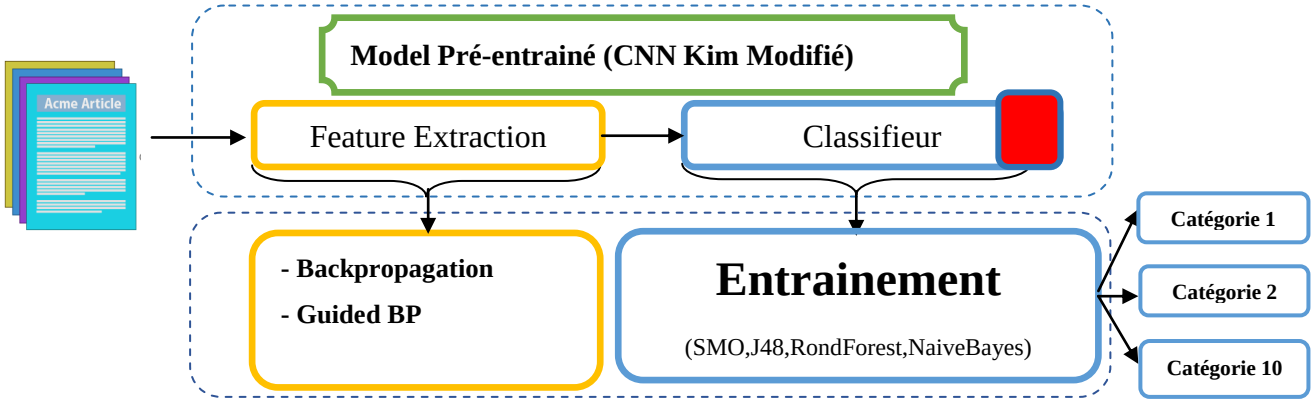

Figure. 3.5. Architecture BP (CNN Kim Modifié) – Algos ML (Transfert Learning)

Les résultats de performance (Précision, Rappelle, F mesure) obtenues pour chacun des algorithmes d'apprentissages automatique avec les technique BP (CNN Kim Modifié) et GI sous l'environnement (Weka) sont présenté dans les tableaux ci-dessous **[Tableau 3.6] [Tableau 3.7] [Tableau 3.8]**

Tableau 3.6 résultats de Performance Algo ML – GI (Weka)

Classifieur	Précision	Rappelle	F- Mesure
SMO	0,886	0,683	0,771
Naive Bayes	0,392	0,735	0,511
Random Forrest	0,899	0,696	0,785
J48	0,809	0,769	0,785
Average	0,747	0,721	0,713

Tableau 3.7 résultats de Performance Algo ML – BP (CNN Kim Modifié)

Classifieur	Précision	Rappelle	F- Mesure
SMO	0,795	0,825	0,809
Naive Bayes	0,847	0,836	0,841
Random Forrest	0,859	0,867	0,862
J48	0,815	0,83	0,822
Average	0,829	0,839	0,834

Tableau 3.8 évaluation des performances de GI (Weka) contre BP (CNN Kim Modifié)

Average	Precision	Recall	F-Measure
Gain	0,747	0,721	0,713
BP	0,829	0,839	0,834
BEST	**0,829**	**0,839**	**0,834**

D'après les résultats présentés dans les tableaux précédents. Nous avons constaté clairement que la méthode de BackPropagation dérivé du notre model CNN Kim Modifié a obtenu les meilleures performances. De cela elle surpasse largement la technique du Gain d'information et qui était utilisé avec les algorithmes classiques d'apprentissage automatique utilisés dans cette partie. En, résumé, toutes ces expériences exprimées en-dessous montrent la haute performance de notre model convolutive CNN Kim Modifié par rapport aux algorithmes ML que ce soit en matière du résultat obtenu par la classification ou par la sélection des caractéristiques.

4.6 Conclusion

Dans ce chapitre, nous avons nous avons représenté un aperçu sur les outils d'apprentissage profond (Deep learning) et également les jeux de données (DataSet) utilisé dans cette étude. Dans cette dernière on a étudié un cas de problème de classification multiple des documents textuel. Plus particulièrement, nous avons proposé une nouvelle architecture du modèle d'apprentissage profond CNN de KIM. Dans les expérimentations on a vu que notre modèle proposé surpasse tous les autres types d'apprentissage automatiques, notamment les arbres de décisions et les SVMs. En outre, une étude comparative est accompagnée afin de montrer l'écart entre les performances des *classificateurs d'apprentissage automatique appliqués avec notre modèle d'apprentissage profond*. Une autre partie a était consacré pour étudier les techniques de sélection des termes. Dans cette optique, nous avons utilisé la technique de Back propagation qui est enveloppée dans notre CNN de KIM. Et une deuxième mesure qui est le gain d'information. Le but de cette deuxième expérience est de montrer l'impact de ces techniques de réduction de dimensionnalité sur la bonne performance du modèle de prédiction. Comme conséquence, Les résultats montrent la réussite des techniques de Back-prpg de CNN.

Conclusion

Conclusion et perspectives

Dans le cadre de ce mémoire, on a travaillé dans un domaine de recherche récent et très intéressant qui est le Deep Learning. Le Deep Learning est le domaine a la mode en intelligence artificiel qui permet de constituer une réponse de mise en œuvre des techniques intelligentes d'apprentissage profond de recherches et d'extraction des connaissances à partir de vastes ensembles de données pour construire des modèles de prédiction qui aide à prendre des décisions. Ces connaissances seront utilisées pour des applications se rattachant à plusieurs domaines. Parmi celles-ci, le traitement automatique du langage naturel, plus précisément la catégorisation automatique des documents textuelle qui est un domaine de recherche rendu très actif par la multiplication du nombre de documents numériques actuellement disponibles. Vu la croissance du flux et de la masse d'information disponible, il est nécessaire de livrer aux lecteurs des mécanismes de classification de l'information qui lui permettront de prendre et d'aide à la décision de lire ou non un document pour répondre à ses besoins.

Dans ce travail, on s'intéresse principalement aux applications des technique d'apprentissage profond pour résoudre les problèmes de classification automatique supervisé des textes, ainsi spécifié l'impact et l'importance des caractéristiques sur la précision du model de prédiction. La première étape est la représentation textuelle catégorielle qui consiste à transformer les textes initiaux en formes exploitables (format texte vers format numérique). La représentation la plus utilisée est la représentation vectorielle (Matrice). En plus une approche de classification supervisée d'apprentissage profond basé sur les réseaux convolutive « CNN » pour l'entrainement d'un modèle de prédiction et la détermination de la catégorie de textes.

L'un des principaux buts de notre travail est d'acquérir un ensemble de connaissances dans le domaine de Deep Learning et leur application pour ressourdre les problèmes de traitement automatique du langage naturel et la classification des textes. De plus des savoirs théorique et pratique sur les techniques de sélection des caractéristiques (FS). Il aurait pu être intéressant de dépasser le cadre d'étude choisie, mais cela conduit à multiplier les thèmes de recherche à aborder.

A travers ce travail on a pu réunir plusieurs compétences :

- Des connaissances dans le domaine du Deep Learning (processus, taches et techniques).
- Les différents traitements de la représentation et codage textuelle (la préparation des textes, la pondération et la réduction de dimensions).
- Les différentes techniques de sélection et identification des caractéristiques (des termes pertinents) pour la classification (IG, BP).
- Les concepts de bases des différentes techniques d'apprentissage profond pour la classification (CNN, LSTM) en comparaison avec les techniques d'apprentissage automatique classique (Arbre de décision, K plus proche voisin, SVM, Naïve Bayes,).
- La démarche à suivre pour effectuer une recherche technique (à travers ce modeste travail).
- La programmation modulaire sous Windows en utilisant le Python qui est un langage le plus populaire dans le domaine du big data, du machine learning et deep learning grâce à ses déférents package et framwork (tanserflow ,Keras).

Tout ou long de ce travail, on a fait le point sur l'ensemble des techniques d'apprentissage profond qui se rapportent directement à la classification textuelle et à l'analyse de données textuelles. Ainsi on a spécifié l'influence et l'impact des caractérisés (terme pertinente) sur la performance du model de prédiction.

Par conséquent le souhait pour ce travail est de le compléter dans les axes suivants :

- L'application de la représentation vectorielle des documents textuels par la méthode d'intégration et plongement des mots (Word2Vec).
- L'application de la classification des textes par la méthode du réseau de neuronne récursif (RNN), plus particulièrement la technique LSTM (Long Short Term Memoiry).
- Développé la complexité de notre model CNN avec plus des couche convolutive en intégrant les techniques d'optimisation (Dropout) pour traité les problèmes de surapprentissage.
- Tester notre modèle sur d'autres jeux de donné plus complexe, volumineuse, nombreuse en termes des catégories.
- L'application de la sélection des caractéristiques par d'autre méthodes qui se base sur la rétropropagation guidé et CAM .

- L’application de notre approche CNN pour classer les document textuel dans d’autre langue défèrent de l’anglais (comme les texte arabe).

- Développement d’une application qui permet le traitement et l’analyse des texte (Tokenisation,stopword,stemming,...) puis les classer par CNN, LSTM, avec prendre en charge les défèrent type de représentation numérique (TF-IDF,W2V,OneHote) .

- Amélioration du modèle en analysant les autres contenus multimédias par exemple : le prétraitement, le filtrage et la catégorisation des images.

- L’application de la catégorisation en générant des modelés non supervisé (Clustering) en se basant sur les caractéristiques de chaque groupe.

Bibliographie

[1] F. Dufour and others, *Les réalités de la'réalité'-deuxième partie: Vers une compréhension des facteurs responsables des progrès de la science moderne (Volume 2)*. Fritz Dufour, 2019.

[2] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[3] M. Batty, P. Lemberger, M. Morel, and J.-L. Raffaëlli, "Big data et machine learning." Dunod, 2016.

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

[5] F. Sebastiani, "Text categorization text mining and its applications," *Text Min. its Appl. A*, pp. 109–129, 2005.

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

[7] D. Nakache and E. Metais, "Evaluation: nouvelle approche avec juges.," in *INFORSID*, 2005, vol. 5, pp. 555–570.

[8] T. M. Mitchell, "Machine Learning, McGraw-Hill Higher Education," *New York*, 1997.

[9] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002.

[10] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[11] "Classifying Reuters-21578 collection with Python – The Practical Academic." https://miguelmalvarez.com/2016/11/07/classifying-reuters-21578-collection-with-python/ (accessed Sep. 10, 2020).

[12] M. Belazzoug, M. Touahria, F. Nouioua, and M. Brahimi, "An improved sine cosine algorithm to select features for text categorization," *J. King Saud Univ. Inf. Sci.*, vol. 32, no. 4, pp. 454–464, 2020.

[13] M. TAFFAR, "INITIATION AL'APPRENTISSAGE."

[14] N. Malki and T. Guerram, "Classification automatique des textes par les réseaux de neurones à convolution," 2019.

[15] E. F. Ilhame, "Clustering des News," UNIVERSITE DE NICE SOPHIA ANTIPOLIS, 2013.

[16] H. B. Barlow, “Unsupervised learning,” *Neural Comput.*, vol. 1, no. 3, pp. 295–311, 1989.

[17] G. Gardarin, “Internet, Intranet et bases de données,” 1999.

[18] L. Lebart, A. Morineau, and M. Piron, *Statistique exploratoire multidimensionnelle*, vol. 3. Dunod Paris, 1995.

[19] T. Bayes, “LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S,” *Philos. Trans. R. Soc. London*, no. 53, pp. 370–418, 1763.

[20] V. Vapnik and A. Chervonenkis, “A note on one class of perceptions,” *Autom. Remote Control*, vol. 25, pp. 821–837, 1964.

[21] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[22] J. Patterson and A. Gibson, *Deep learning: A practitioner’s approach*. “ O’Reilly Media, Inc.,” 2017.

[23] “Définition | Deep Learning - Apprentissage profond | Futura Tech.” https://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/ (accessed Sep. 10, 2020).

[24] A. Gulli and S. Pal, “Deep Learning with Keras. Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK,” 2017.

[25] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[26] I. Vasilev, D. Slater, G. Spacagna, P. Roelants, and V. Zocca, *Python Deep Learning: Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow*. Packt Publishing Ltd, 2019.

[27] J. Zhou and O. G. Troyanskaya, “Predicting effects of noncoding variants with deep learning--based sequence model,” *Nat. Methods*, vol. 12, no. 10, pp. 931–934, 2015.

[28] “Le Deep Learning Partie 2 : définition - Saagie.” https://www.saagie.com/fr/blog/qu-est-ce-que-le-deep-learning/ (accessed Sep. 10, 2020).

[29] D. Y. Moualek, “Deep Learning pour la classification des images.,” 07-03-2017, 2017.

[30] AKSHAY BADKAR, “Difference Between Machine Learning And Deep Learning,” Nov. 21, 2017. http://www.iamwire.com/2017/11/difference-between-machine-learning-and-deep-learning/169100 (accessed Sep. 10, 2020).

[31] “Let’s Understand the Difference between Machine Learning Vs. Deep Learning - Inteliment Technologies,” Sep. 18, 2019. https://www.inteliment.com/blog/our-thinking/lets-understand-the-difference-between-machine-learning-vs-deep-learning/ (accessed Sep. 10, 2020).

[32] “Deep Learning : 3 choses à savoir - MATLAB & Simulink.” https://fr.mathworks.com/discovery/deep-learning.html (accessed Sep. 10, 2020).

[33] “Introduction to Deep Learning and Neural Network.” https://blog.quantinsti.com/introduction-deep-learning-neural-network/ (accessed Sep. 10, 2020).

[34] HOURRANE Oumaima, “DataScienceToday - Réseaux neuronaux récurrents et LSTM,” Jun. 26, 2018. https://www.datasciencetoday.net/index.php/fr/machine-learning/148-reseaux-neuronaux-recurrents-et-lstm (accessed Sep. 10, 2020).

[35] nacira berri, “Utilisation des techniques de Deep Learning pour l’extraction des concepts à partir des documents textuels,” Mohamed Khider – BISKRA, 2019.

[36] Y. Kim, “Convolutional neural networks for sentence classification,” *arXiv Prepr. arXiv1408.5882*, 2014.

[37] Jason Brownlee, “Best Practices for Text Classification with Deep Learning,” *Deep Learning for Natural Language Processing*, Oct. 23, 2017. https://machinelearningmastery.com/best-practices-document-classification-deep-learning/ (accessed Sep. 10, 2020).

[38] H. T. Siegelmann, “Réseaux de neurones récurrents - Data Analytics Post,” *Neural Networks and Analog Computation*, 1997. https://dataanalyticspost.com/Lexique/reseaux-de-neurones-recurrents (accessed Sep. 10, 2020).

[39] B. Ramsundar and R. B. Zadeh, *TensorFlow for deep learning: from linear regression to reinforcement learning*. “ O’Reilly Media, Inc.,” 2018.

[40] Y. Yang and X. Liu, “A re-examination of text categorization methods,” in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pp. 42–49.

[41] R. Besançon, A. Rozenknop, J.-C. Chappelier, and M. Rajman, “Intégration probabiliste de sens dans la représentation de textes,” in *Actes de la 8eme conférence sur le Traitement Automatique des Langues Naturelles (TALN’2001)*, 2001, vol. 1, no. CONF, pp. 83–91.

[42] G. Salton and M. J. McGill, “Introduction to Modern Information Retrieval McGraw Hill Book Company,” *New York*, 1983.

[43] M. F. Porter and others, “An algorithm for suffix stripping.,” *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[44] D. D. Lewis, “An evaluation of phrasal and clustered representations on a text categorization task,” in *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, 1992, pp. 37–50.

[45] L. Màrquez, C. Callison-Burch, and J. Su, “Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,” 2015.

[46] S. Goto, *Algorithmic Learning Theory:... Workshop, ALT...: Proceedings*. Springer, 1990.

[47] M. N. Asim, M. Wasim, M. S. Ali, and A. Rehman, "Comparison of feature selection methods in text classification on highly skewed datasets," in *2017 First International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT)*, 2017, pp. 1–8.

[48] A. M. Oudshoff, I. E. Bosloper, T. B. Klos, and L. Spaanenburg, "Knowledge discovery in virtual community texts: Clustering virtual communities," *J. Intell. Fuzzy Syst.*, vol. 14, no. 1, pp. 13–24, 2003.

[49] "Apprentissage automatique et traitement du langage (chapitre 18 AIMA, pp Tom Mitchell Machine Learning) - ppt video online télécharger." https://slideplayer.fr/slide/1576321/ (accessed Sep. 10, 2020).

[50] C. S. Lim, K. J. Lee, and G. C. Kim, "Multiple sets of features for automatic genre classification of web documents," *Inf. Process. Manag.*, vol. 41, no. 5, pp. 1263–1276, 2005.

[51] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Icml*, 1997, vol. 97, no. 412–420, p. 35.

[52] S. Tufféry, *Data mining et statistique décisionnelle: l'intelligence dans les bases de données*. Editions Technip, 2005.

[53] "Weka 3 - Data Mining with Open Source Machine Learning Software in Java." https://www.cs.waikato.ac.nz/ml/weka/ (accessed Sep. 10, 2020).

[54] "TensorFlow." https://www.tensorflow.org/?hl=fr (accessed Sep. 10, 2020).

[55] "Keras — Wikipédia." https://fr.wikipedia.org/wiki/Keras (accessed Sep. 10, 2020).

[56] "Welcome to Python.org." https://www.python.org/ (accessed Sep. 10, 2020).

[57] "scikit-learn: machine learning in Python — scikit-learn 0.23.2 documentation." https://scikit-learn.org/stable/ (accessed Sep. 10, 2020).

[58] "Bienvenue dans Colaboratory - Colaboratory." https://colab.research.google.com/notebooks/intro.ipynb (accessed Sep. 10, 2020).

[59] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.