



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

**Université Mohamed Khider – BISKRA**

Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

**Département d'informatique**

N° d'ordre : SIOD 16/M2/2019

## **Mémoire**

présenté pour obtenir le diplôme de master académique en

# **Informatique**

Parcours : **Systemes d'information, Optimisation et décision**

---

# **Utilisation des techniques de Deep Learning pour l'extraction des concepts à partir des documents textuels**

---

**Par :**

**BERRI Nacira**

Soutenu le 06 juillet 2019, devant le jury composé de :

GUEMEIDA Abdelbasset

MCB

Président

ABDELLI Belkacem

MCB

Rapporteur

MEKLID Abdessalem

MAA

Examineur



# الإهداء

أهدي هذا  
العمل لنفسي.

# الشكر

أولا وأخرا، أحمداً الله على تمكني من الوصول لهذه المرحلة من مشواري  
الدراسي،  
وبعد، أشكر والديّ على التعب، على السهر، وكل عائلتي إخوتي و  
أخواتي،  
أشكر صديقتي الطفولة والدراسة صبرينة و إيمان، على التشجيع  
خاصة في أزمات الفشل،  
ولا أنسى أشكر أساتذتي الدكتور بلقاسم عبدلي الذي كان طوال فترة  
إنجازي لهذا العمل،  
أشكر كل من يحبني من قريب أو بعيد ويتمنى لي الخير والنجاح.

## ملخص / Résumé

---

**Résumé:** La plupart des études se sont concentrées sur l'extraction de données, en particulier du fait de la croissance rapide du nombre de documents, qui contiennent de gros volumes d'informations dans des bases de données textuelles composées de vastes collections de documents provenant de sources telles que des livres, des articles et des documents de recherche. Avec un grand nombre de ces documents, il est difficile mais nécessaire d'organiser automatiquement les documents en catégories afin de faciliter leur extraction et leur analyse ultérieure. Cela a conduit à l'utilisation de l'apprentissage profond dans ces études, cette dernière a dépassé son entrée dans plusieurs domaines. Compte tenu des multiples langues dans lesquelles les documents textuels sont écrits, nous pouvons combiner l'apprentissage profond avec un traitement ce qu'on appelle le traitement du langage naturel.

**Mots-clés:** L'apprentissage profond, Traitement du langage naturel, Les documents textuelles.

**ملخص:** ركزت معظم الدراسات الحديثة على استخراج البيانات ، لا سيما بسبب النمو السريع في عدد الوثائق التي تحتوي على كميات كبيرة من المعلومات في قواعد البيانات النصية التي تتكون من مجموعات كبيرة من الوثائق من مصادر مختلفة. مثل الكتب والمقالات والأوراق البحثية. مع وجود العديد من هذه المستندات ، من الصعب ولكن من الضروري تنظيم المستندات تلقائيًا في فئات لتسهيل استرجاعها وتحليلها. فقد أدى ذلك إلى استخدام التعلم العميق في هذه الدراسات ، والذي تجاوز دخوله إلى العديد من المجالات.

بالنظر إلى اللغات المتعددة التي تُكتب فيها المستندات النصية ، يمكننا الجمع بين التعلم العميق والمعالجة المسماة معالجة اللغة الطبيعية.

**الكلمات المفتاحية:** التعلم العميق ، معالجة اللغة الطبيعية ، الملفات النصية.

<b>Introduction générale.....</b>	<b>10</b>
a. Contexte .....	11
b. Objectifs .....	11
c. Organisation du mémoire.....	12
<b>Chapitre 01 L'apprentissage profond et le traitement de langage naturel .....</b>	<b>13</b>
1.1 Introduction .....	14
1.2 L'apprentissage profond .....	14
1.2.1 Définition .....	14
1.2.2 L'apprentissage profond et l'apprentissage automatique.....	15
1.2.3.1 L'apprentissage automatique .....	15
1.2.3.1.a Apprentissage supervisé.....	15
1.2.3.1.b Apprentissage non supervisé.....	16
1.2.3.2 La différence entre l'apprentissage profond et l'apprentissage automatique .....	16
1.2.3.3 L'importance de l'apprentissage profond.....	17
1.2.3 Objectif de l'apprentissage profond .....	17
1.2.4 Principe de fonctionnement .....	18
1.2.5 Modèles d'apprentissage profond .....	20
1.2.6 Domaines d'application [31] .....	20
1.2.7 Avantages de l'apprentissage profond .....	21
1.2.8 Limites de l'apprentissage profond.....	21
1.3 Le traitement du langage naturel.....	21
1.3.1 Définition .....	21
1.3.2 Classification de traitement de langage naturel .....	22
1.3.3 Tâches de traitement du langage naturel .....	23
1.3.4 Objectif .....	25
1.3.5 Domaines d'application du NLP.....	25
1.4 L'apprentissage profond et le traitement de langage naturel.....	26
1.4.1 Le traitement du langage naturel classique .....	26
1.4.2 Le traitement du langage naturel avec l'apprentissage profond .....	26
1.4.3 Les algorithmes de l'apprentissage profond avec le NLP .....	27

1.5	Travaux Connexes .....	27
1.5.1	Classification automatique de textes .....	28
1.5.2	Analyse de sentiments .....	28
1.6	Conclusion .....	28
<b>Chapitre 02 Analyse de données textuelles. ....</b>		<b>29</b>
2.1	Introduction .....	30
2.2	Concepts de base .....	30
2.2.1	Données structurées.....	30
2.2.2	Données non structurées.....	30
2.2.3	Données textuelles .....	31
2.3	Domaines d'application.....	31
2.4	Problème d'analyse de données textuelles .....	31
2.5	Les opérations principales d'analyse de données textuelles.....	32
2.5.1	Collecte des données textuelles .....	32
2.5.2	Le prétraitement du texte .....	32
2.5.2.1	Qu'est-ce que le prétraitement du texte?.....	32
2.5.2.2	Les opérations du prétraitement .....	32
2.5.3	La représentation du texte.....	34
2.5.3.1	Méthodes standard.....	34
2.5.3.2	Word Embedding.....	35
2.6	Niveaux d'analyse de texte .....	35
2.6.1	L'analyse syntaxique .....	35
2.6.2	L'analyse sémantique .....	36
2.6.3	L'analyse pragmatique .....	36
2.7	Conclusion .....	36
<b>Chapitre 03 Conception du système.....</b>		<b>37</b>
3.1	Introduction .....	38
3.2	Conception globale.....	38
3.2.1	Architecture générale .....	38
3.3	Conception détaillée .....	39
3.3.1	Architecture détaillée .....	39

3.3.2Explication .....	40
3.4 Conclusion .....	44
<b>Chapitre 04 Implémentation .....</b>	<b>45</b>
4.1 Introduction .....	46
4.2 Choix du langage de programmation .....	46
4.3 L'environnement de développement .....	47
4.3.2Les étapes de la création du modèle.....	48
4.4 Présentation de la fenêtre d'application .....	50
4.5 Conclusion .....	50
4.6 Annexes .....	51
<b>Conclusion générale .....</b>	<b>53</b>
<b>Références.....</b>	<b>54</b>



<i>Figure 1: Position de notre objectif. [1]</i> .....	11
<i>Figure 1.2.1: La relation entre les trois branches l'intelligence artificielle(IA), (ML) et (DL).</i> 14	
<i>Figure 1.2.3.1.a: Exemple d'apprentissage supervisé. [14]</i> .....	16
<i>Figure 1.2.3.2 La différence entre l'apprentissage profond et l'apprentissage automatique.</i> ....	16
<i>Figure 1.2.3.3 illustration de performance l'apprentissage profond et l'apprentissage automatique.</i> .....	17
<i>Figure 1.2.4.a Topologie de réseau de neurones avec une seule sortie.</i> .....	18
<i>Figure 1.2.4.b Topologie de réseau de neurones profond.</i> .....	19
<i>Figures1.2.4.c les fonctions d'activation d'un réseaux de neurones. [31]</i> .....	19
<i>Figure 1.3.1 Schéma aux 3 domaines NLP, IA et DL. [36]</i> .....	22
<i>Figure 1.3.2 Classification générale du traitement du langage naturel. [12]</i> .....	22
<i>Figure 1.3.3.a Exemple de Part-Of-Speech Tagging.</i> .....	23
<i>Figure 1.3.3.b Tags et leurs descriptions.</i> .....	24
<i>Figure 1.3.3.c Exemple de la tache NER.</i> .....	24
<i>Figure 1.3.3.d Exemple de la tache Parsing.</i> .....	25
<i>Figure 1.4.1 illustration une processus classique du NLP. [37]</i> .....	26
<i>Figure 1.4.2 Exemple d'application du NLP avec le Deep learning. [37]</i> .....	27
<i>Figure 1.4.3 Algorithmes d'apprentissage et leur utilisation avec le NLP. [36]</i> .....	27
<i>Figure 1.5.2 Exemple de la classification automatique de textes.</i> .....	28
<i>Figure 2.2.1 : Exemple sur les données structurées.</i> .....	30
<i>Figure 2.2.2 : Exemple de données non structurées.</i> .....	30
<i>Figure 2.5.2.1 Exemple du prétraitement d'un texte.</i> .....	32
<i>Figure 2.5.2.2.a Exemple de l'opération Stemming.</i> .....	33
<i>Figure 2.5.2.2.b Exemple de l'opération Lemmatisation.</i> .....	33
<i>Figure 2.5.2.2.c Exemple de normalisation d'un Token.</i> .....	34
<i>Figure 2.6.1 Exemple d'un analyse de phrase syntaxiquement</i> .....	35
<i>Figure 3.2.1 : Représentation de l'architecture générale du système.</i> .....	38
<i>Figure 3.3.1 : Représentation de l'architecture détaillée de système.</i> .....	39
<i>Figure 3.3.2.a : Exemple d'un document normalisé (sans Stopword).</i> .....	40
<i>Figure 3.3.2.b : Illustration de la liste de termes extraits.</i> .....	41
<i>Figure 3.3.2.c : Exemple matrice Docs-termes avec la pondération tf-idf.</i> .....	42
<i>Figure 3.3.2.d illustration de transformation de données.</i> .....	43
<i>Figure 3.3.2.e représentation de texte.</i> .....	43
<i>Figure 3.3.2.f Exemple d'Embedding.</i> .....	44
<i>Figure 3.3.2.g illustration détaillé du modèle d'apprentissage crée.</i> .....	44
<i>Figure 4.2 Logo en langage python.</i> .....	46
<i>Figure 4.3 l'environnement du PyCharm.</i> .....	47
<i>Figure 4.3.1 processus de l'exécution du prétraitement d'un document (.Txt)</i> .....	48
<i>Figure 4.3.2.a processus de la création du modèle d'apprentissage.</i> .....	48
<i>Figure 4.3.2.b processus d'entraînement un modèle.</i> .....	49
<i>Figure 4.3.2.c illustration des quelques instructions de la création du modèle.</i> .....	50
<i>Figure 4.4 fenêtre d'application.</i> .....	50
<i>Figure 4.6 dépendances de la bibliothèque NLTK.</i> .....	51

## *Introduction générale*

---

### **a. Contexte**

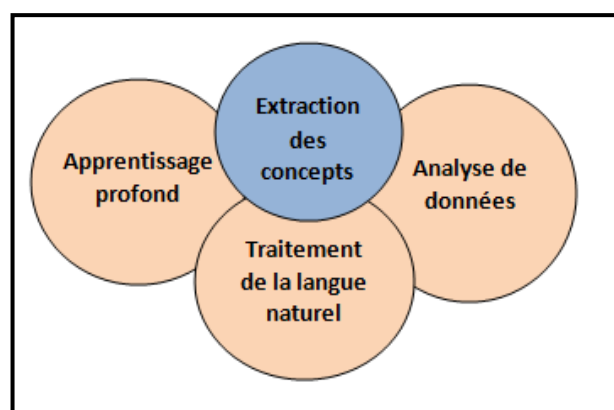
L'émergence de données volumineuses a créé de nouveaux défis pour les équipes d'analyse de données. Il est lié à une partie de l'analyse des données qui traite des données non structurées sous forme de texte. De nombreuses applications impliquent de plus en plus une grande quantité de ce type des données. Des exemples de telles données comprennent les courriers électroniques, les textes, les pages Web, les enregistrements de texte, les documents de recherche etc.

Depuis quelques années, le monde a connu un grand renouveau avec les techniques de l'apprentissage profond, inspirées des réseaux de neurones du cerveau. Plusieurs communautés travaillent depuis de nombreuses années sur l'information textuelle, avec des outils et des objectifs souvent très différents. En informatique, le principal objet d'étude est l'utilisation du langage naturel en vue de la réalisation d'interfaces homme-machine. Avec l'intégration de l'apprentissage profond dans le domaine du traitement du langage naturel, ce dernier a assisté à une évolution de la méthode classique à la méthode moderne.

### **b. Objectifs**

Notre but est de présenter les liens entre apprentissage profond et les documents textuels, afin que l'extraction des informations textuelles telles que les termes les plus importants.

La problématique de la différence dans le contenu des textes et leurs différents domaines, technologie, médecine, sport, politique, etc. nous conduit à l'intersection de nombreuses disciplines différentes (la *figure 1*):



*Figure 1: Position de notre objectif. [1]*

### **c. Organisation du mémoire**

Après, la présentation de notre objectif et l'introduction générale, ce mémoire sera organisé en quatre chapitres de la façon suivante:

- Dans **le premier chapitre**, nous présentons un aperçu sur l'apprentissage profond, ainsi que ses divers objectifs, puis dans la deuxième partie du chapitre, nous exposerons le traitement du langage naturel et de ses divers domaines, ainsi que la manière dont l'apprentissage profond peut être intégré au traitement du langage naturel.
- Dans **le deuxième chapitre**, tout d'abord, nous allons aborder quelques concepts de base, ensuite nous allons exposer les différentes opérations de prétraitement nécessaires pour l'analyse de données textuelles. Enfin, nous allons découvrir les différentes manières de représenter le contenu du texte, ce qui est une étape importante pour extraire des informations du texte.
- Dans **le troisième chapitre**, une vue conceptuelle de notre système est décrite dans ce chapitre, on va détailler les deux architectures (générale et détaillée) avec l'identification de la modélisation appropriée à atteindre avec un modèle d'apprentissage profond.
- Dans **le quatrième chapitre**, ce chapitre présente les outils utilisés dans la phase de l'implémentation. Nous terminons notre mémoire par une conclusion générale.

## *Chapitre 01*

# *L'apprentissage profond et le traitement du langage naturel*

---

## 1.1 Introduction

À l'heure actuelle, la technique d'apprentissage profond représente la plus grande part de la recherche effectuée par des spécialistes, en particulier dans la mesure où elle intervient dans plusieurs domaines tels que le traitement du langage naturel.

Ce chapitre est divisé en 3 parties : une première, nous allons définir l'apprentissage profond ; certaines de ses différences avec l'apprentissage automatique ; et principe à la fin de cette partie. Une seconde partie nous allons explorer le traitement du langage naturel ; les tâches les plus importantes du traitement du langage naturel et les domaines d'application. La dernière partie nous consacrerons la discussion au traitement du langage naturel classique, enfin nous allons discuter sur le NLP avec l'apprentissage profond.

## 1.2 L'apprentissage profond

### 1.2.1 Définition

L'apprentissage profond (ou Deep Learning en anglais) est une branche de l'apprentissage automatique (ou Machine Learning en anglais), ce dernier est une branche de l'intelligence artificielle[24]. Nous illustrons cette imbrication entre les trois branches par la *figure 1.2.1*.

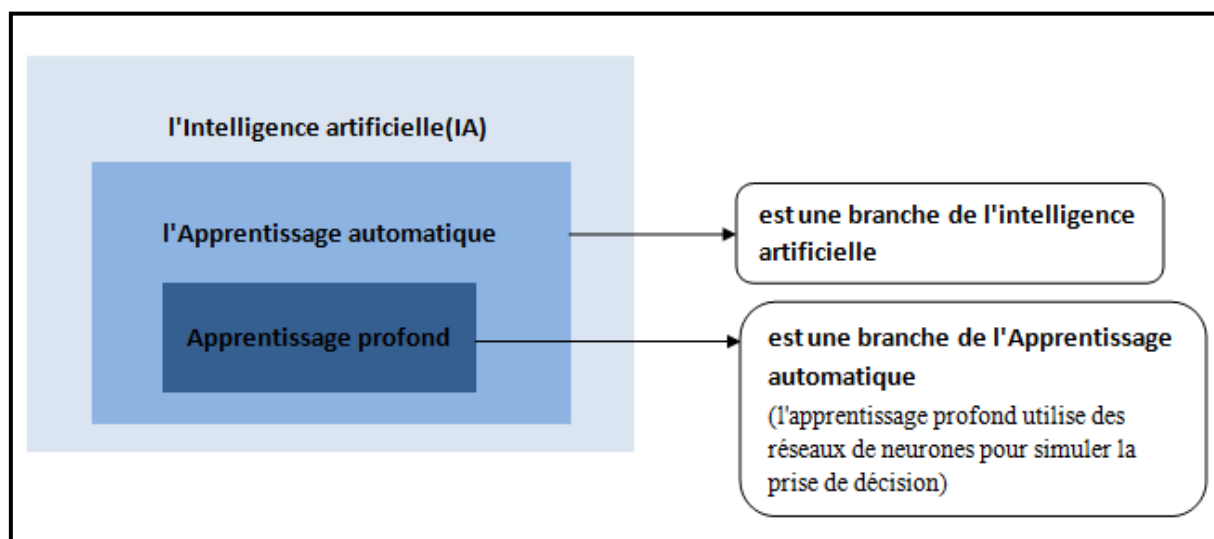


Figure 1.2.1: La relation entre les trois branches l'intelligence artificielle(IA), (ML) et (DL).

L'apprentissage profond a été introduit dans l'étude des réseaux de neurones, inspirée à l'origine par une étude du fonctionnement physiologique du cerveau, L'apprentissage profond capable d'apprendre et de traiter des données complexes, et tente également de résoudre des tâches complexes ([6] ;[28] ).

“ *Le Deep Learning permet à des modèles composés de plusieurs couches de traitement d'apprendre des représentations des données avec de multiples niveaux d'abstraction.* ”

Deep Learning – Yann LeCun, Yoshua Bengio and Geoffrey Hinton, Nature, 2015

L'apprentissage profond est basé sur :

- Réseaux de neurones profonds.
- Algorithme d'entraînement spécifique.

L'apprentissage profond utilise des algorithmes de réseaux de neurones avec un nombre important de couches, les premières couches traversées par les données faisaient ce travail de préparation des données pour les couches plus profondes. L'aspect problématique de ce principe est qu'elle donne l'impression dont l'apprentissage profond est comme s'il existait depuis les années 1980 ([23] ;[24] ;[40]).

## 1.2.2 L'apprentissage profond et l'apprentissage automatique

### 1.2.3.1 L'apprentissage automatique

L'apprentissage automatique (ou apprentissage machine, Machine Learning en anglais) est un sous-domaine de l'intelligence artificielle (IA), qui a évolué à partir de l'étude de la reconnaissance des modèles et de la théorie de l'apprentissage computationnel en intelligence artificielle. Son but est d'entraîner un ensemble d'algorithmes sûr de grandes quantités de données afin de pouvoir classifier des données futures (la précision dépendant de la quantité et la qualité des données). Cette technique s'appuie sur le développement de programmes informatiques capables d'acquérir de nouvelles connaissances afin de s'améliorer et d'évoluer d'eux mêmes des qu'ils sont à exposer de nouvelles données. Ils fonctionnent en construisant un modèle à partir d'exemples d'entrées afin de faire des prédictions ou des choix basés sur les données plutôt que de suivre des instructions de programme statiques[10].

L'apprentissage automatique est généralement divisé en :

- L'apprentissage automatique supervisé ;
- L'apprentissage automatique non supervisé

#### 1.2.3.1.a Apprentissage supervisé

##### Définition

La forme la plus commune d'apprentissage automatique est l'apprentissage supervisé. L'apprentissage supervisé est une méthode permettant de transformer un jeu de données en un autre, le programme est formé sur un ensemble prédéfini d'exemples de formation, ce qui facilite ensuite sa capacité à parvenir à une conclusion précise lorsque de nouvelles données sont fournies ([10] ;[14];[26]).

##### Exemple

Par exemple, si vous avez un jeu de données appelé Monday Stock Prices (la *figure 1.2.3.1.a*), qui enregistre le prix de chaque stock sur chaque Lundi depuis 10 ans et un deuxième jeu de données appelé Tuesday Stock Prices cours boursiers enregistré au cours de la même période, un algorithme d'apprentissage supervisé pourrait essayer d'utiliser l'un pour prédire l'autre [14].

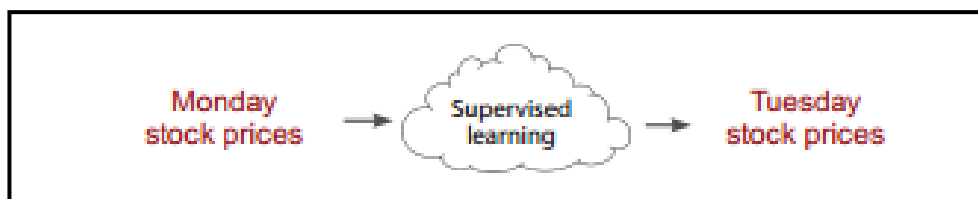


Figure 1.2.3.1.a: Exemple d'apprentissage supervisé. [14]

### 1.2.3.1.b Apprentissage non supervisé

#### Définition

L'apprentissage non supervisé, encore appelé apprentissage à partir d'observations, partage une propriété commune avec l'apprentissage supervisé: il transforme un jeu de données en un autre. Mais l'ensemble de données dans lequel il se transforme n'est pas connu ou compris auparavant. Contrairement à l'apprentissage supervisé sera quant à lui alimenté uniquement par des exemples, et créera lui-même les classes qui lui semblent les plus judicieuses (clustering) ou des règles d'associations (algorithmes Apriori). L'algorithme K-moyen (K-means) permet de comprendre facilement le concept de classification non supervisée ([14];[16]).

### 1.2.3.2 La différence entre l'apprentissage profond et l'apprentissage automatique

Un processus de l'apprentissage automatique (Machine Learning) commence par l'extraction manuelle de caractéristiques pertinentes à partir de données en s'appuyant sur ces caractéristiques, un modèle est créé, mais l'apprentissage profond ignore ces étapes manuelles. Par exemple pour classer des images avec la Machine Learning (la figure 1.2.3.2), les choix de caractéristiques et de classificateur doivent être effectués manuellement. Avec l'apprentissage profond, l'extraction de caractéristiques et le processus de modélisation sont automatiques [31].

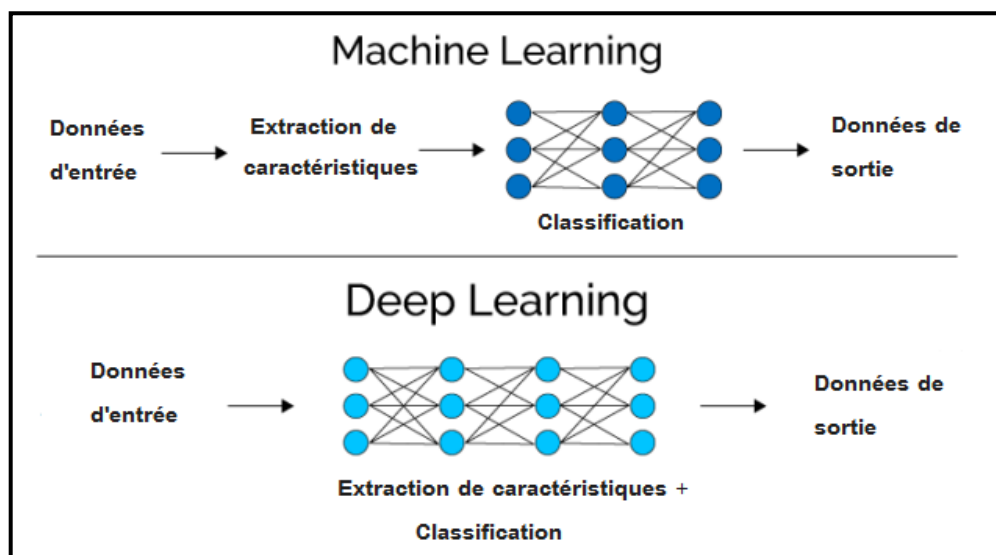


Figure 1.2.3.2 La différence entre l'apprentissage profond et l'apprentissage automatique.

L'entraînement des modèles s'effectue à l'aide de vastes ensembles de données labellisées et d'architectures de réseaux de neurones qui apprennent des caractéristiques directement depuis les données, sans avoir à effectuer une extraction manuelle. Une autre différence majeure est le fait que les algorithmes de Deep Learning évoluent avec les données [31].



Pour réussir une application de Deep Learning, vous avez besoin d'un volume de données très important (des milliers d'images par exemple, souvent quelques millions et des entrées de très grande dimension) pour entraîner le modèle, en plus d'un ou de plusieurs GPU (processeur graphique) pour traiter les données rapidement. Si vous n'avez pas besoin de ces éléments, il est préférable d'utiliser l'apprentissage automatique plutôt que l'apprentissage profond[31].

### 1.2.3.3 L'importance de l'apprentissage profond

L'apprentissage automatique n'est pas utile lorsque vous travaillez avec des données de grandes dimensions, c'est-à-dire que nous avons un grand nombre d'entrées et de sorties. Ne peut pas résoudre des problèmes cruciaux d'intelligence artificielle comme le NLP, la reconnaissance d'image etc.

Extraction de caractéristiques est un des grands défis des modèles d'apprentissage machine traditionnels.

Cette extraction automatisée des caractéristiques permet aux modèles de Deep Learning d'atteindre un taux de précision particulièrement élevé pour les tâches de vision par ordinateur (la figure 1.2.3.3) [31].

Les modèles d'apprentissage profond sont capables de se concentrer sur les fonctionnalités appropriées par eux-mêmes, nécessitant peu de conseils de la part du programmeur.

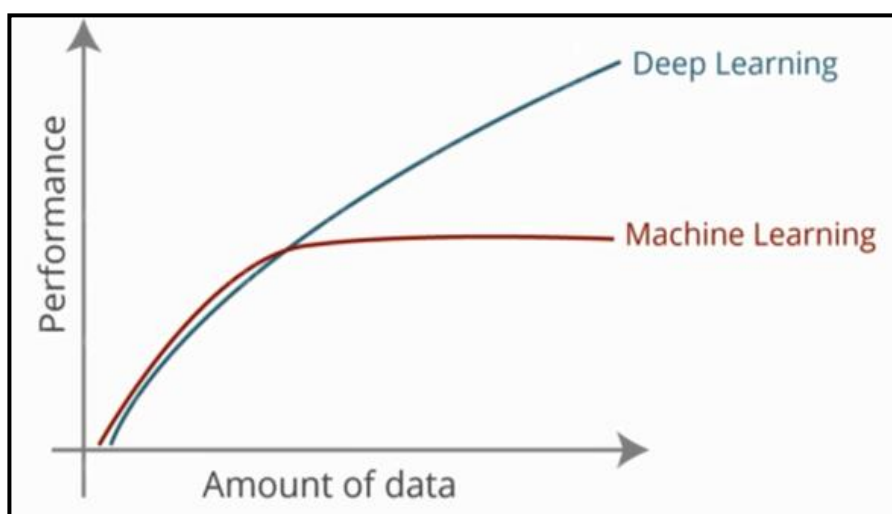


Figure 1.2.3.3 illustration de performance l'apprentissage profond et l'apprentissage automatique.

### 1.2.3 Objectif de l'apprentissage profond

L'apprentissage profond a pour objectif d'explorer comment les ordinateurs peuvent exploiter les données pour développer des fonctionnalités, utiliser les données disponibles pour apprendre à bien décider, à acquérir de bonnes connaissances pour d'être capable de donner de bonnes réponses sûr de nouveaux exemples, inconnus du système, en généralisant les exemples connus, pour apprendre mieux ([22] ;[29]).

Il est nécessaire de donner des données prétraitées, par exemple, on fait décomposer les images en formes simples, qui seront ensuite données au réseau de neurones artificiels afin qu'il tente de les reconnaître. L'extraction automatique de caractéristiques est un autre des grands avantages de l'apprentissage en profondeur ([24] ;[40]).

#### 1.2.4 Principe de fonctionnement

Le réseau comporte 3 composants: couche d'entrée couche cachée ou couches et couche de sortie. Le terme « profond » se rapporte généralement au nombre de couches cachées du réseau de neurones. Les réseaux de neurones classiques ne comportent que 2 à 3 couches cachées, tandis que les réseaux profonds peuvent en compter jusqu'à 150 [32].

L'idée est d'utiliser la structure de couche de réseau neuronal en empilant plusieurs couches les unes sur les autres, de manière à faciliter le mécanisme de décomposition. Par conséquent, chaque couche d'un réseau de neurones profonds (Deep Neural Networks DNN) fonctionne comme une seule transformation pour extraire davantage les données[8].

Le réseau de neurones le plus connu et le plus simple à comprendre est le réseau de neurones multicouches à anticipation. Il contient un calque d'entrée, un ou plusieurs calques masqués et un seul calque de sortie (la figure 1.2.5.a). Chaque couche peut avoir un nombre différent de neurones et chaque couche est entièrement connectée à la couche adjacente [24].

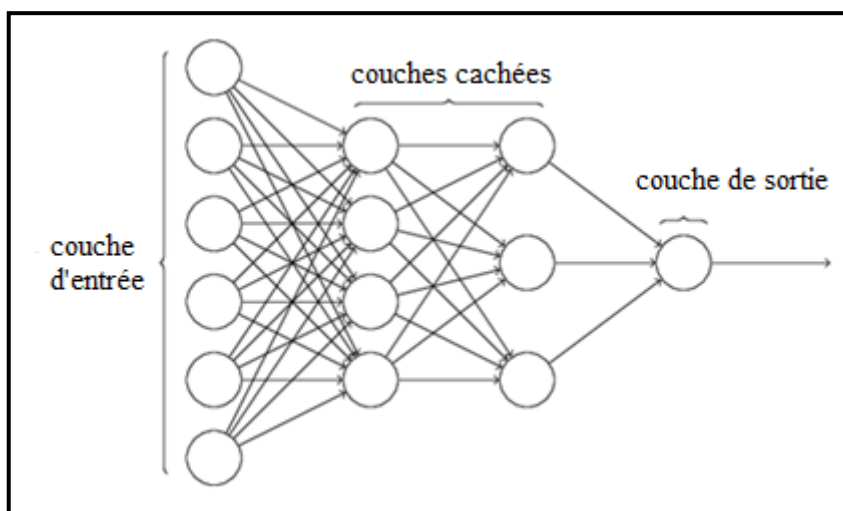


Figure 1.2.4.a Topologie de réseau de neurones avec une seule sortie.

Un réseau de neurones est défini comme un ensemble de nœuds (appelés neurones) connectés via des liaisons dirigées (flèche), chaque flèche représente une connexion entre la sortie d'un neurone et l'entrée d'un autre (les flèches entrantes étant les entrées du neurone et les flèches sortantes étant les sorties du neurone). Chaque flèche porte un poids, reflétant son importance, chaque nœud étant une unité de traitement qui exécute une fonction de nœud statique sur son signal entrant pour générer une sortie de nœud unique[28]. Les valeurs d'entrée, ou en d'autres termes, nos données sous-jacentes, sont transmises via ce «réseau» de couches masquées jusqu'à ce qu'elles convergent vers la couche de sortie. La couche en sortie correspond à notre prédiction: il peut s'agir d'un nœud si le modèle ne génère qu'un nombre ou de quelques nœuds s'il s'agit d'un problème de classification multi-classe. La forme à l'intérieur des neurones dans les couches centrales représente

une fonction d'activation (typiquement un  $\sigma(x) = \frac{1}{1 + e^{-x}}$ ) qui est appliquée à la valeur du neurone avant de le transmettre à la sortie[4].

La topologie d'un réseau de neurones illustrée dans la figure 1.2.5.b.

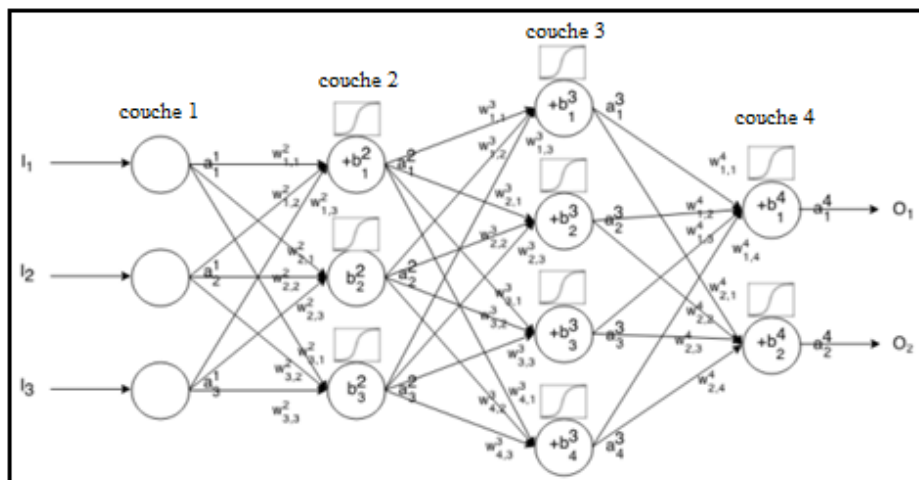


Figure 1.2.4.b Topologie de réseau de neurones profond.

Les couches cachées d'un réseau de neurones apportent des modifications aux données pour éventuellement déterminer quelle est sa relation avec la variable cible. Chaque nœud a un poids et multiplie sa valeur d'entrée par ce poids. Pour déterminer ce que devraient être ces petits poids, nous utilisons généralement un algorithme appelé Back propagation.

### Back propagation

Une partie importante des réseaux de neurones, y compris des architectures profondes modernes, est la propagation en arrière des erreurs à travers un réseau. Chaque exemple d'entrées fait une propagation et back propagation afin de mettre à jour les poids utilisés par les neurones plus proches de l'entrée.

### Fonctions d'activation

Ci-dessous sont les types de fonction d'activation (la figure 1.2.4.c).

Nom f(x)	Graphe	Equation	Description
<b>Sigmoid ou Logit</b>	<p>Sigmoid  <math>y = \frac{1}{1 + e^{-x}}</math></p>	$y = \sigma(x) = \frac{1}{1 + e^{-x}}$	En gros, elle réduit la sortie entre 0 et 1 en appliquant une fonction de journalisation, ce qui rend le problèmes de classification plus faciles
<b>Tanh</b>		$y = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	la plage de la fonction tanh est comprise entre (-1 et 1), le reste restant identique à celui de sigmoïde.
<b>ReLU</b>		$y = \max(0, x)$	convertit tout ce qui est inférieur à zéro en zéro. Ainsi, la plage devient 0 à l'infini.

Figures 1.2.4.c les fonctions d'activation d'un réseaux de neurones. [31]

### Autres fonctions d'activation

- **Fonction Softmax:** Softmax est presque similaire à Sigmoid, mais il calcule les probabilités de l'événement sur 'n' différentes classes, ce qui sera utile pour déterminer la cible dans les problèmes de classification multi-class.
- **Fonction d'activation linéaire:** un neurone linéaire prend une combinaison linéaire des entrées pondérées; et la sortie peut prendre n'importe quelle valeur entre -infinité et l'infini.
- **Fonction d'activation non linéaire:** ce sont les plus utilisées et elles rendent la sortie restreinte entre certaines plages.

### 1.2.5 Modèles d'apprentissage profond

Dans cette section on va présenter quelques types de modèle ou bien des algorithmes d'apprentissage profond.

Dans les années 1980, la plupart des réseaux de neurones ne formaient qu'une seule couche en raison du coût de calcul et de la disponibilité des données. De nos jours, nous pouvons nous permettre d'avoir plus de couches cachées dans nos réseaux de neurones, d'où le surnom d'apprentissage profond. Les différents types de réseaux de neurones disponibles à l'utilisation ont également proliféré, des modèles tels que les réseaux de neurones Convolutionnels (CNN), les réseaux de neurones récurrents(RNN) et (LSTM).

#### Les réseaux de neurones Convolutionnels (CNN)

Un réseau de neurones Convolutionnelles ou (Convolutional neural networks en anglais) un des types de réseaux de neurones profonds le plus répandu utilise des couches à convolution 2D. Cette architecture sont donc parfaitement adaptées au traitement des données 2D telles que les images. Les réseaux de neurones à convolution s'appuient sur plusieurs dizaines, voire plusieurs centaines de couches cachées pour apprendre à identifier les caractéristiques d'une image. La complexité des caractéristiques apprises augmente avec le nombre de couches cachées du réseau, les modèles CNN ont été inventés à l'origine pour voir l'ordinateur, sont ceux par la suite avérés efficaces pour le traitement de langage naturel et ont donné d'excellents résultats en analyse syntaxique sémantique en modélisation de phrases([2] ;[31] ).

#### Les réseaux de neurones récurrents (RNN)

Est une classe de réseaux de neurones artificiels dans lesquels les connexions entre les nœuds forment un graphe dirigé le long d'une séquence temporelle. Les réseaux de neurones récurrents, ont été conçus pour traiter les problèmes de prédiction de séquence. Cela inclut les séquences de texte. Ils sont également utilisés comme modèles génératifs nécessitant une sortie de séquence, non seulement avec du texte, mais également dans des applications telles que la génération d'écritures manuscrite, données texte, et la reconnaissance de la parole.

### 1.2.6 Domaines d'application [31]

L'apprentissage profond a de nombreuses applications en informatique, on cite quelques domaines:

## **Conduite automatisée**

Les chercheurs du secteur automobile ont recours au Deep Learning pour détecter automatiquement des objets tels que les panneaux stop et les feux de circulation. Le Deep Learning est également utilisé pour détecter les piétons, évitant ainsi nombre d'accidents.

## **Recherche médicale**

À l'aide du Deep Learning, les chercheurs en oncologie peuvent dépister automatiquement les cellules cancéreuses. Des équipes de l'Université de Californie à Los Angeles (UCLA) ont conçu un microscope qui génère un ensemble de données de grandes dimensions afin d'entraîner une application de Deep Learning à identifier avec précision des cellules cancéreuses.

## **Électronique**

Le Deep Learning est utilisé pour la reconnaissance audio et vocale. Par exemple, les appareils d'assistance à domicile qui répondent à votre voix et connaissent vos préférences fonctionnent grâce à des applications de Deep Learning.

### **1.2.7 Avantages de l'apprentissage profond**

Contrairement aux algorithmes classiques, qui apprennent des solutions ne fonctionnant que pour un seul problème, les solutions trouvées par les réseaux de neurones profonds peuvent souvent être appliquées à d'autres tâches similaires (moyennant quelques adaptations). On appelle cela le Transfert Learning(en anglais). Lorsque l'on fait du Transfert Learning, on coupe en quelque sorte la partie du réseau qui est spécifique à une tâche particulière et on garde la partie généraliste. Un des avantages majeurs des réseaux de Deep Learning réside dans leur capacité à continuer à s'améliorer en même temps que le volume de vos données augmente[31].

### **1.2.8 Limites de l'apprentissage profond**

Cela quelques limites de l'apprentissage profond :

- Pour résoudre des problèmes de plus en plus complexes, il n'est pas suffisant d'ajouter toujours plus de couches. Les deux grosses problématiques de Deep learning;
- la difficulté d'apprentissage;
- la complexité calculatoire croissante avec le nombre de couches ;
- le Deep Learning exige une puissance de calcul considérable ;
- entraînement coûteux en ressources (calcul, mémoire, ...).
- L'établissement de réseaux de neurones profonds pose un défi: déterminer le nombre de couches cachées ainsi que le nombre de neurones par couches.

## **1.3 Le traitement du langage naturel**

### **1.3.1 Définition**

Le traitement du langage naturel (NLP) est un domaine de recherche et d'application qui explore comment analyser et représenter naturellement les textes à un ou plusieurs niveaux d'analyse linguistique afin de comprendre et de manipuler du texte ou des mots en langage naturel afin de

réaliser des tâches utiles en utilisant l'ordinateur et d'essayer d'atteindre le niveau humain de compréhension et d'analyse son langage naturel ([8];[11]).

Le traitement du langage naturel est un domaine interdisciplinaire de l'informatique; ses applications englobent un certain nombre de domaines d'étude, le NLP dépend de la linguistique, de l'intelligence artificielle, de l'apprentissage automatique, de mathématiques et de robotique, etc. ([5];[11];[19]).

Le NLP est un sous-domaine de la science informatique qui vise à permettre aux ordinateurs de comprendre le langage de manière «naturelle», comme le font les humains. En règle générale, cela fait référence à des tâches telles que la compréhension du sentiment du texte, la reconnaissance de la parole et la génération de réponses aux questions. Le NLP remonte au développement de la science informatique dans les années 1940 [21].

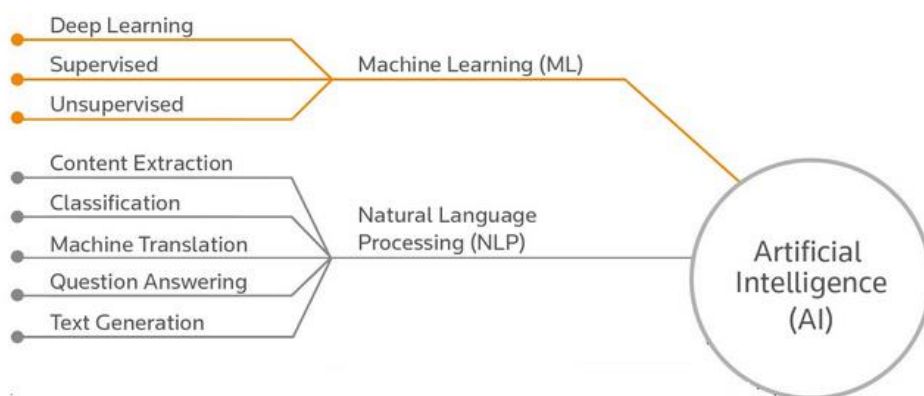


Figure 1.3.1 Schéma aux 3 domaines NLP, IA et DL. [36]

### 1.3.2 Classification de traitement de langage naturel

Le traitement du langage naturel. Il regroupe deux domaines spécifiques (la figure 1.3.2):

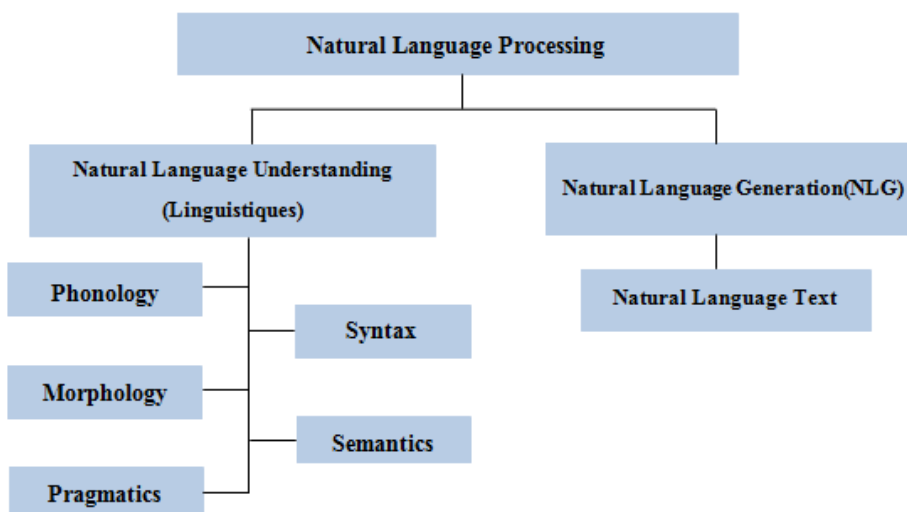


Figure 1.3.2 Classification générale du traitement du langage naturel. [12]

## Natural Language Understanding (NLU)

Qui vise à comprendre le langage naturel ; essaye de comprendre le sens d'un texte donné. La nature et la structure de chaque mot dans le texte doivent être comprises pour NLU.

Pour comprendre la structure, NLU tente de résoudre l'ambiguïté suivante en langage naturel [35]:

- Ambiguïté lexicale - les mots ont plusieurs sens ;
- Ambiguïté syntaxique - Phrase comportant plusieurs arbres d'analyse ;
- Ambiguïté sémantique - Phrase à significations multiples

## Natural Language Generation (NLG)

C'est le processus de production automatique de texte à partir de données structurées dans un format lisible avec des phrases et des phrases significatives, permet à une machine de s'exprimer de manière naturelle pour un être humain [35].

Génération de langage naturel divisée en trois étapes proposées:

- Planification du texte - la commande du contenu de base dans les données structurées est effectuée.
- Planification des phrases - les phrases sont combinées à partir de données structurées pour représenter le flux d'informations.
- Réalisation - les phrases grammaticalement correctes sont finalement produites pour représenter le texte.

### 1.3.3 Taches de traitement du langage naturel

Dans cette section, nous présentons brièvement les quatre tâches standard du NLP.

#### Part-Of-Speech Tagging

Est une partie cruciale du traitement du langage naturel. Elle consiste à étiqueter les mots avec une partie du discours, a pour objectif de classer chaque mot avec un signe unique indiquant son rôle syntaxique(à associer à chaque mot d'un texte sa classe morphosyntaxique), par exemple: nom, un verbe, un adjectif,...etc. Le POS constitue la base de la résolution d'entité nommée, de l'analyse des sentiments, de la réponse aux questions, et l'ambiguïté du sens des mots( [20] ;[27]).

#### Exemple

Exemple de Part-Of-Speech Tagging (la figure 1.3.3.a).

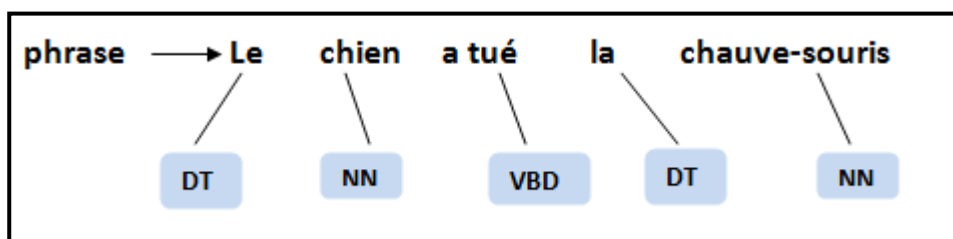


Figure 1.3.3.a Exemple de Part-Of-Speech Tagging.

## Liste du Tag & description

La *figure 1.3.3.b* ci-dessous représente des quelques Tags et leurs descriptions.

Tag	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun

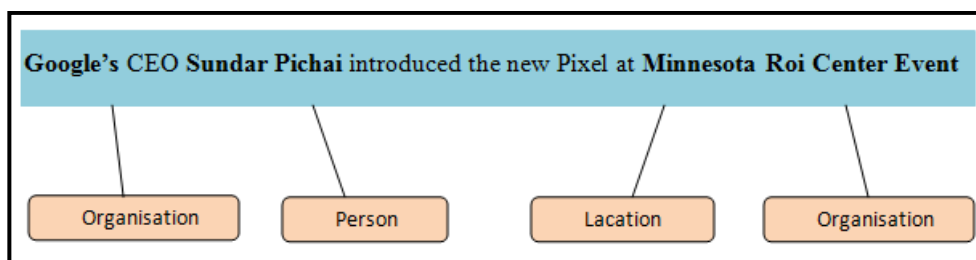
*Figure 1.3.3.b Tags et leurs descriptions.*

## Named Entity Recognition (NER)

Les étiqueteurs NER (ou reconnaissance d'entités nommées en français) marque les éléments atomiques de la phrase en catégories plus grandes telles que (noms de personnes, noms d'organisations ou d'entreprises, noms de lieux, quantités, distances, valeurs, dates, etc.) ([21];[27]).

### Exemple

Exemple de Named Entity Recognition (la *figure 1.3.3.c*).



*Figure 1.3.3.c Exemple de la tache NER.*

## Semantic Role Labeling (SRL)

SRL vise à donner un rôle sémantique à un constituant syntaxique d'une phrase[27].

## Parsing ou Chunking

Est le processus qui consiste à déterminer la structure syntaxique d'un texte en analysant ses mots constitutifs sur la base d'une grammaire sous-jacente (du langage).

L'exemple de grammaire ci-dessous, où chaque ligne indique une règle de la grammaire à appliquer à un exemple de phrase «Tom a mangé une pomme».



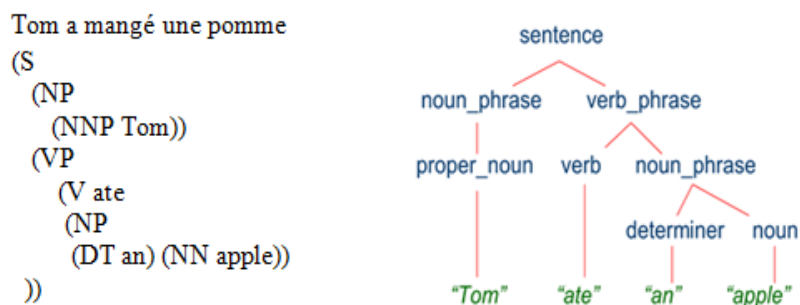


Figure 1.3.3.d Exemple de la tache Parsing.

### 1.3.4 Objectif

Le domaine du traitement du langage naturel (NLP) vise à convertir le langage humain en une représentation formelle facile à manipuler par les ordinateurs pour étudier des problèmes fondamentaux du traitement de la langue naturelle, ce qui est bien adapté à la modélisation des données textuelles afin d'en extraire des informations et, éventuellement, de représenter les mêmes informations différemment ([7] ; [13] ; [15]).

### 1.3.5 Domaines d'application du NLP

Le traitement du langage naturel (NLP) a récemment attiré beaucoup d'attention sur l'analyse du langage humain, couvrant divers domaines tels que la traduction automatique, la détection de courrier électronique indésirable, les questions médicales, les réponses aux questions, le traitement et la synthèse de textes en langage naturel, la reconnaissance vocale, l'extraction d'informations, l'analyse de sentiment([12];[13]).

#### Exemple

- Analyse de sentiments (Sentiment Analysis en anglais);
- Classification de document( Document classification en anglais);
- Spam – ham mail classification ;
- Resume shortlisting ;
- Document summarization.
- Produire de texte.

#### Applications du NLP avec Text Mining

- Spell Checking (vérification orthographique en français) ;
- Information extraction ;
- Keyword Recherche.

## 1.4 L'apprentissage profond et le traitement de langage naturel

Au cours des dix dernières années, le traitement du langage naturel est devenu l'un des domaines clés de l'intelligence artificielle avec des réseaux de neurones et des techniques d'apprentissage approfondi améliorant les performances dans plusieurs tâches.

Le traitement du langage naturel est l'un des sujets le plus en vogue dans le domaine des technologies de la parole et du langage. L'apprentissage automatique peut également comprendre comment effectuer des tâches importantes du NLP ([8] ; [11]).

L'arrivée à maturité d'analyse des données avec la convergence de nouvelles technologies telles que l'intelligence artificielle, la narration automatique, l'activation vocale, le traitement de texte et d'images, a fait de Deep Learning le composant central pour l'utilisation du traitement du langage naturel dans les applications[38].

### 1.4.1 Le traitement du langage naturel classique

Auparavant, les informaticiens divisaient un langage en ses formes grammaticales, telles que des parties de discours, des phrases, etc., en utilisant des algorithmes complexes, c.-à-d. le NLP traitait des données statiques (la figure 1.4.1). Aujourd'hui, l'apprentissage en profondeur est la clé pour effectuer les mêmes exercices([9] ;[25]).

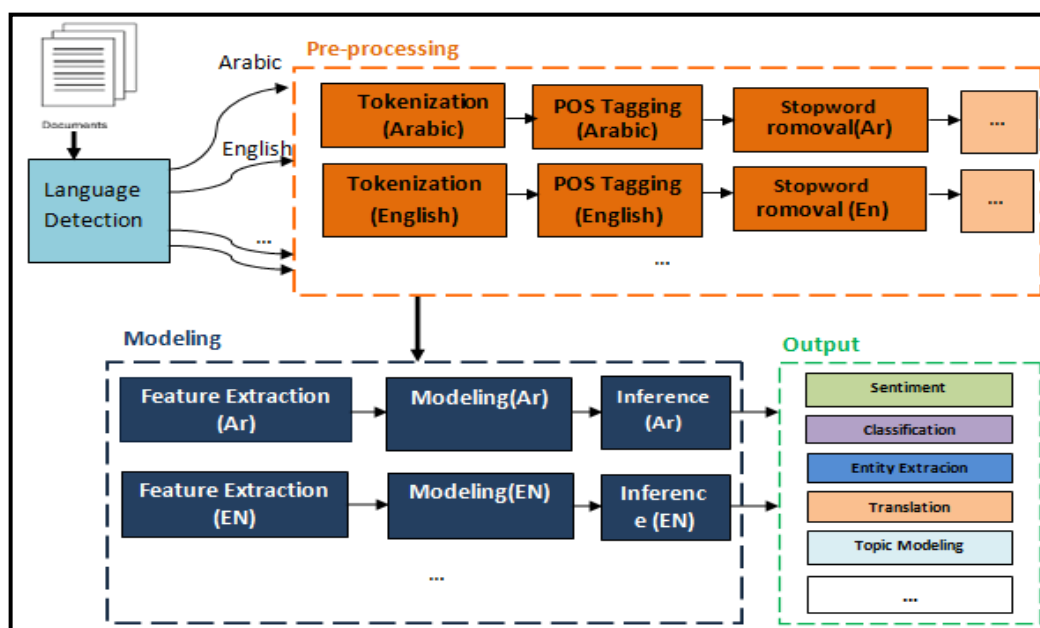


Figure 1.4.1 illustration une processus classique du NLP. [37]

### 1.4.2 Le traitement du langage naturel avec l'apprentissage profond

“ Des vagues d'apprentissage profond ont déferlé sur les rives de la linguistique informatique depuis plusieurs années maintenant, mais 2015 semble être l'année où toute la force du tsunami a frappé les grandes conférences sur le traitement du langage naturel (NLP) ”. Selon Dr. Christopher D. Manning Déc. 2015; [39].

Depuis quelques années les réseaux neuronaux se montrent très efficaces et ont été appliqués avec succès dans toutes les tâches de traitement automatique des langues (TAL)([17] ;[18]). exemple illustré dans la figure 1.4.2.

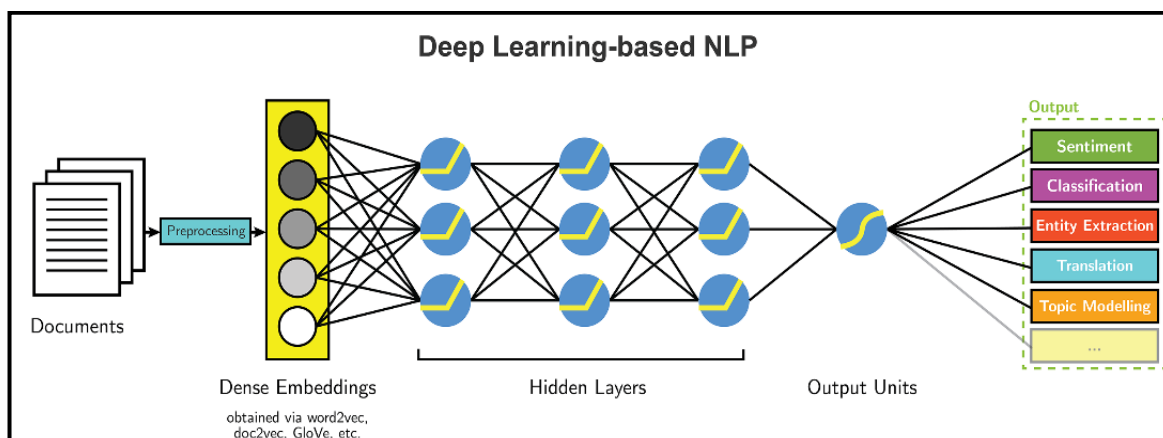


Figure 1.4.2 Exemple d'application du NLP avec le Deep learning. [37]

### 1.4.3 Les algorithmes de l'apprentissage profond avec le NLP

La figure 1.4.3 ci-dessous représente des quelques algorithmes et leur utilisation avec le NLP.

Algorithme	Utilisation du NLP
Neural Network - NN (feed)	<ul style="list-style-type: none"> <li>▪ Part-of-speech Tagging</li> <li>▪ Tokenization</li> <li>▪ Named Entity Recognition</li> <li>▪ Intent Extraction</li> </ul>
Recurrent Neural Networks -(RNN)	<ul style="list-style-type: none"> <li>▪ Machine Translation</li> <li>▪ Question Answering System</li> <li>▪ Image Captioning</li> </ul>
Recursive Neural Networks	<ul style="list-style-type: none"> <li>▪ Parsing sentences</li> <li>▪ Sentiment Analysis</li> <li>▪ Paraphrase detection</li> <li>▪ Relation Classification</li> <li>▪ Object detection</li> </ul>
Convolutional Neural Network - (CNN)	<ul style="list-style-type: none"> <li>▪ Sentence/ Text classification</li> <li>▪ Relation extraction and classification</li> <li>▪ Spam detection</li> <li>▪ Categorization of search queries</li> <li>▪ Semantic relation extraction</li> </ul>

Figure 1.4.3 Algorithmes d'apprentissage et leur utilisation avec le NLP. [36]

### 1.5 Travaux Connexes

De très nombreux travaux ont appliqué les techniques de l'apprentissage profond aux données textuelles. On n'en cite que quelques-uns. Dans cette section, nous présentons brièvement des travaux basés sur l'apprentissage profond et le traitement du langage naturel.

### 1.5.1 Classification automatique de textes

Les cas d'usages les plus fréquents en analyse textuelle sont la classification de texte; la classification du texte est l'une des tâches les plus importantes du traitement automatique du langage. Il s'agit du processus de classification des chaînes de texte ou des documents dans différentes catégories (la *figure 1.5.2*). La classification de texte a diverses applications, telles que la détection du sentiment de l'utilisateur à partir d'un Tweet, la classification d'un courrier électronique en tant que Spam ou Ham[33].

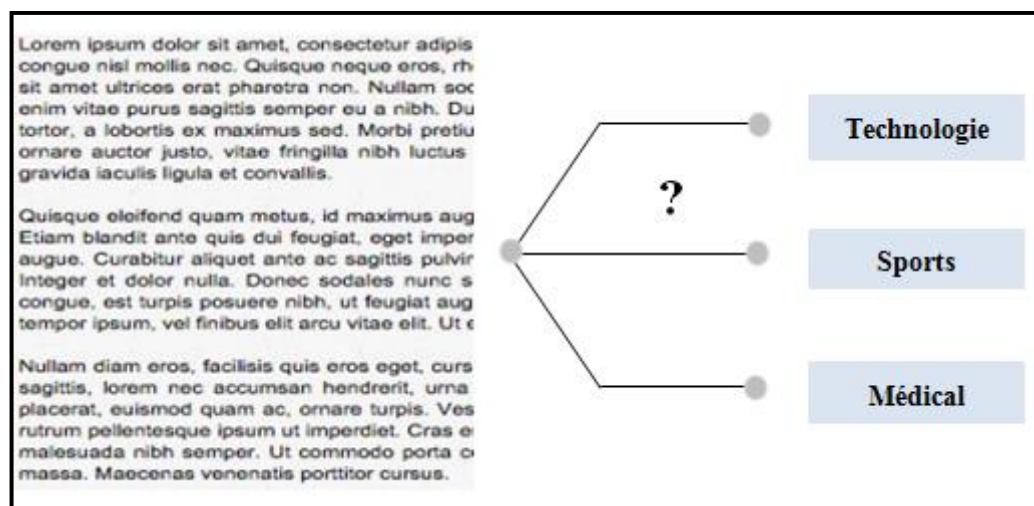


Figure 1.5.2 Exemple de la classification automatique de textes.

### 1.5.2 Analyse de sentiments

Bien que la langue ou les expressions de langue contiennent sans aucun doute des ambiguïtés dans la nature, peu de recherches ont été menées ces dernières années dans des domaines pertinents, en partie à cause de la prolifération des techniques d'apprentissage profond du traitement automatique du langage.

## 1.6 Conclusion

L'apprentissage profond est le domaine le plus émergent de l'apprentissage automatique et a apporté une contribution importante dans divers domaines de recherche. Cela a permis de surmonter les inconvénients des méthodes traditionnelles en rendant les systèmes moins complexes et plus rapides. L'apprentissage profond a été utilisé avec le traitement automatique du langage dans plusieurs domaines de recherche, ce qui est très prometteur et constitue un succès. Dans ce chapitre nous avons exposé le technique de l'apprentissage profond, ainsi que ses avantages, et ses limites, le traitement du langage naturel, enfin les techniques de l'apprentissage profond avec le traitement du langage naturel.

## ***Chapitre 02 Analyse de données textuelles***

---

## 2.1 Introduction

L'émergence du Big Data où le stockage des données numériques n'est plus un problème et où les sources de données disponibles se multiplient fait de l'analyse des données textuelles un enjeu crucial. Un des aspects centraux du prétraitement est de transformer ces données textuelles peu structurées. La plupart des outils du traitement de langage naturel (TLN) permettent le traitement des données textuelles.

Dans ce chapitre, nous aborderons quelques concepts tels que les concepts de données textuelles qui constitue la base de notre projet, ainsi que les méthodes et techniques de prétraitement des données textuelles. Ce chapitre est divisé en 3 parties : une première traite des concepts de base, domaines d'application. Une seconde partie traite de l'exploration, du nettoyage et de la normalisation du texte et des différents types de transformations qui vont vous permettre de mieux comprendre vos données textuelles. La dernière partie sera consacrée aux niveaux d'analyse de texte.

## 2.2 Concepts de base

Dans cette section on va expliquer quelques notions.

### 2.2.1 Données structurées

Les données structurées sont des données dont les éléments sont adressables pour une analyse efficace. Il est organisé dans un référentiel formaté qui est généralement une base de données.

#### Exemple

ID	Nom	Age
1	Hamid	45
2	Khalid	44
3	Samira	25
4	Laila	22

Figure 2.2.1 : Exemple sur les données structurées.

### 2.2.2 Données non structurées

Les données non structurées sont des données qui ne sont pas organisées de manière prédéfinie ou qui ne possèdent pas de modèle de données prédéfini.

#### Exemple

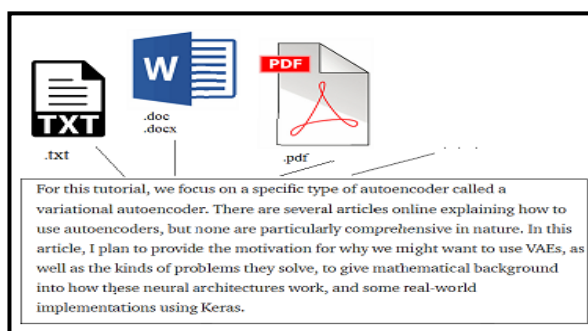


Figure 2.2.2 : Exemple de données non structurées.

### 2.2.3 Données textuelles

#### Définition

Les données textuelles sont une source d'information qui permettrait d'extraire de la connaissance, tels que le contenu d'un document textuel (les articles, les livres...), le contenu d'un page web, etc.

**Données textuelles = données non structurées**

### 2.3 Domaines d'application

L'analyse de données textuelles entre dans de nombreux domaines tels que :

- Fouille de données (Text Mining) ;
- Analyse de sentiments.

Voici quelques exemples d'applications basées sur l'analyse de texte pour extraire des données et atteindre un certain résultat :

#### Résumé automatique

Résumé d'un document. Recherche des phrases les plus représentatives dans un document.

#### Résumé d'un corpus

Recherche du document le plus représentatif dans un corpus ou recherche des phrases représentatives à partir de plusieurs documents. (La notion de distance entre phrases, entre documents, joue un rôle important).

#### Identification des tendances

Les données arrivent en flux. S'appuyer sur la temporalité (par exemple textes sur les réseaux sociaux). Analyser l'évolution de la popularité des thèmes, identifier les sujets émergents (Topic Detection en anglais).

#### Analyse des liens

Analyser les relations (cooccurrences) entre les termes, qui peuvent être des personnages (nous approfondirons cette idée dans l'analyse des réseaux sociaux).

#### Et autres

- L'indexation automatique;
- L'extraction et la cartographie de concepts;
- La classification automatique;
- Le rapprochement entre textes.

### 2.4 Problème d'analyse de données textuelles

Le problème principal dans la fouille de données textuelles est le fossé sémantique, c'est-à-dire l'écart entre l'interprétation qu'un ordinateur peut obtenir automatiquement à partir d'un texte et la signification de ce même texte pour un humain (de la catégorie ciblée par le texte).

## 2.5 Les opérations principales d'analyse de données textuelles

### 2.5.1 Collecte des données textuelles

Certaines étapes, comme la collecte de données font partie de tout projet de fouille de données textuelles et présentent des particularités qui dépendent de la nature des données. Dans cette section on va exposer la phase la plus importante dans l'analyse des textes.

### 2.5.2 Le prétraitement du texte

#### 2.5.2.1 Qu'est-ce que le prétraitement du texte?

La préparation des données va jouer un rôle fondamental dans l'analyse de textes et l'extraction de l'information. Prétraiter un texte signifie simplement le mettre sous une forme appropriée (la figure 2.5.2.1).

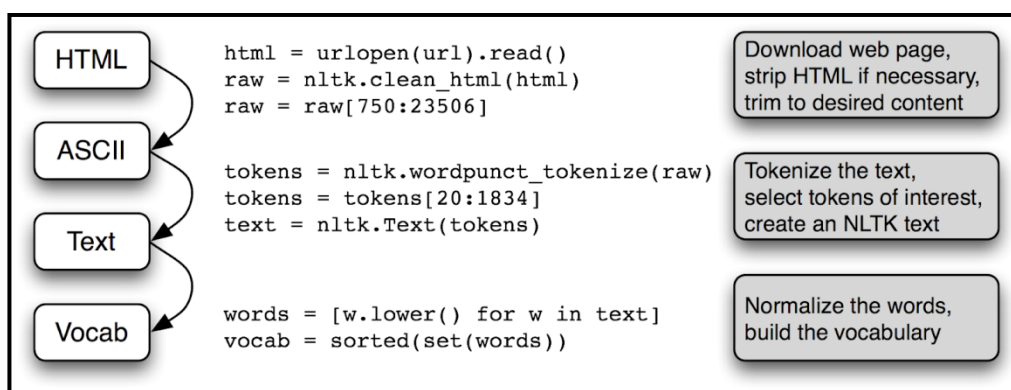


Figure 2.5.2.1 Exemple du prétraitement d'un texte.

#### 2.5.2.2 Les opérations du prétraitement

Dans cette section, nous mentionnerons les processus de traitement tribal du texte, qui sont présents dans tout processus de traitement, quel que soit le but du traitement, pas tous, mais la plupart d'entre eux sont des processus que nous suivons dans le traitement des textes.

Lorsque vous travaillez sur différents aspects du traitement de langage naturel( NLP), vous pouvez vous retrouver souvent confronté à l'une des tâches suivantes:

- Tokenization.
- Normalisation, Stemming et lemmatisation.
- Nettoyage des mots vides.
- ...

L'analyse de données textuelles passe par plusieurs étapes brièvement décrites dans la suite.

#### Lowercasing

Mettre les données textuelles en minuscule, L'objectif de cette opération est d'obtenir toutes les données dans un format uniforme et de nous assurer que «NLP» et «nlp» sont traités de la même manière, afin d'éliminer la redondance des mots [20].



## Tokenization

Est une étape obligatoire du prétraitement du texte pour tout type d'analyse. Il existe de nombreuses bibliothèques pour effectuer la création de jetons comme NLTK, SpaCy et TextBlob [20].

## Stemming

Est utile pour traiter les problèmes de parcimonie ainsi que pour normaliser le vocabulaire, a pour un objectif normaliser les termes dans sa forme de base ou racine (la *figure 2.5.2.2.a*).

Le mot d'origine	Le mot après Stemming
connect, connected, connection, connections, connects	connect

*Figure 2.5.2.2.a Exemple de l'opération Stemming.*

## Lemmatisation

Regroupe différentes formes infléchies d'un terme, appelé Lemma, en quelque sorte similaire à Stemming, car il mappe plusieurs termes dans une racine commune (la *figure 2.5.2.2.b*).

La lemmatisation ne présente aucun avantage significatif par rapport à l'acquisition de bases à des fins de recherche et de classification de texte, mais vous pouvez toujours l'essayer pour voir son impact sur votre métrique de performance.

Le mot d'origine	Le mot après Lemmatisation
goose, geese	goose

*Figure 2.5.2.2.b Exemple de l'opération Lemmatisation.*

## Suppression de mots vides

Les mots vides (ou Stopwords en anglais), sont des mots très courants qui n'ont pas de sens ou qui ont un sens inférieur à ceux d'autres mots-clés (sont des mots ou des phrases descriptives qui caractérisent vos documents) dans le texte par exemple "le", "la" ou "des", etc. .ils possèdent la grande partie du texte et les supprime, on se concentre sur les mots importants [20].

## Normalisation (Standardisation du texte)

La normalisation de texte est le processus de transformation de texte en une forme canonique (standard) (la *figure 2.5.2.2.c*). Par exemple, les mots goood et gud peuvent être transformés en good, sa forme canonique.

Token	Token normalisée
2moro 2mrrw 2morrow 2mrw	Tomorrow
b4	before
:) :-)	smile

Figure 2.5.2.2.c Exemple de normalisation d'un Token.

### Suppression du bruit

La suppression du bruit consiste à supprimer des caractères, des chiffres et des morceaux de texte pouvant gêner votre analyse de texte. L'élimination du bruit est l'une des étapes les plus essentielles du prétraitement du texte. On peut remplacer des caractères spéciaux et des caractères numériques par des caractères blancs.

Cette étape est très importante car la ponctuation n'ajoute aucune information ni valeur supplémentaire. Par conséquent, la suppression de toutes ces instances contribuera à réduire la taille des données et à accroître l'efficacité des calculs[20].

## 2.5.3 La représentation du texte

### 2.5.3.1 Méthodes standard

#### Sac-de mot

Sac de mots (ou Bag of Word en anglais) est une description de document très utilisé en recherche d'information. Son principe se résume en 3 phases :

La Tokenisation.

La constitution d'un dictionnaire global qui sera en fait le vocabulaire.

L'encodage des chaînes de caractères par rapport au vocabulaire constitué précédemment.

#### A one-hot encoding

chaque mot d'un texte correspond à un élément vectoriel. Le principal avantage des vecteurs de mots **one-hot encoding** est qu'ils capturent la cooccurrence de mots binaires (elle est également décrite comme un sac de mots) ce qui est suffisant pour effectuer un large éventail de tâches de la PNL, y compris la classification de texte et permettent de résoudre certains des problèmes tel que n-gram et TF-IDF.

#### Tf-idf

TF-Idf (en anglais Term Frequency-inverse document frequency) est le résultat d'un calcul, dans l'algorithme des moteurs de recherche, permettant d'obtenir un poids, une évaluation de la pertinence

d'un document par rapport à un terme, en tenant compte de deux facteurs : la fréquence de ce mot dans le document (TF) et le nombre de documents contenant ce mot (IDF) dans le corpus étudié. On peut calculer par la formule illustrée dans la *figure(2.5.3.a)*

$$TF \times IDF(t_k, d_j) = Occ(t_k, d_j) \times \log \frac{Nb_{doc}}{Nb_{doc}(t_k)}$$

K :terme ;  
 J : document  
 Nb : le nombre de documents  
 Nb<sub>doc</sub> : le nombre de documents citent le terme t<sub>k</sub>

### 2.5.3.2 Word Embedding

Cette technique permet de représenter chaque mot d'un dictionnaire par un vecteur de nombre réelle correspondant. Ceci facilite l'analyse sémantique des mots. Cette nouvelle représentation à ceci de particulier que les mots apparaissant dans des contextes similaires possèdent des vecteurs correspondants qui sont relativement proches.

## 2.6 Niveaux d'analyse de texte

Après avoir préparé les données de texte (texte, page Web, texte d'image, livre numérique, etc.), l'étape suivante consiste à analyser le texte au niveau supérieur. Dans la section suivante, nous afficherons différents niveaux d'analyse de texte.

### 2.6.1 L'analyse syntaxique

C'est une étape pour comprendre et reconnaître la structure des phrases.

Des règles syntaxiques (la grammaire), Identifier les relations syntaxiques entre les mots (*la figure 2.6.1*).

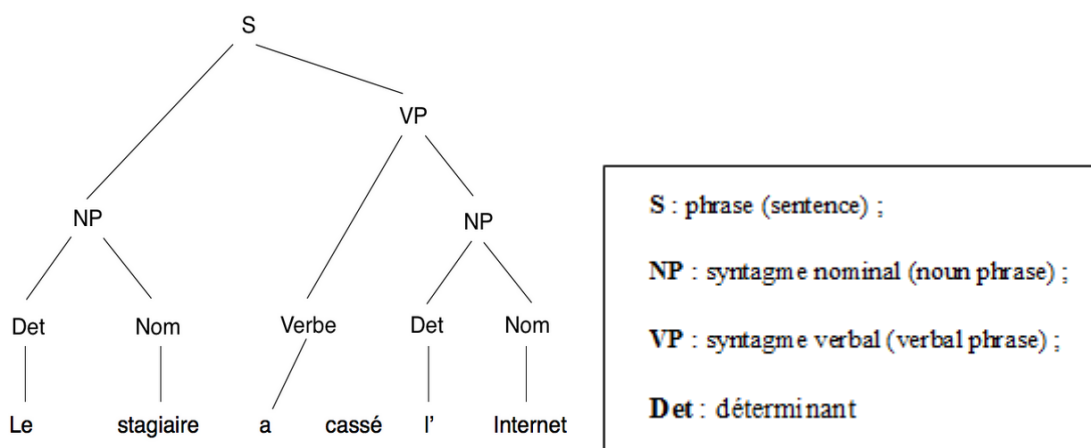


Figure 2.6.1 Exemple d'un analyse de phrase syntaxiquement

### **2.6.2 L'analyse sémantique**

Comme indiqué précédemment, c'est l'étape ou le problème le plus difficile auquel nous sommes confrontés lors de l'analyse de données textuelles.

L'analyse sémantique est l'étape de l'extraction du sens des phrases. Cette étape compliquée en raison de l'ambiguïté de la langue. Par exemple un même mot peut avoir plusieurs sens possibles.

### **2.6.3 L'analyse pragmatique**

Est l'étape qui vient compléter l'analyse sémantique. En analysant les mots et phrases proches les uns des autres.

## **2.7 Conclusion**

Tant que l'être humain aura besoin de sauvegarder des données numériques dans tous les domaines de la vie, les développeurs continueront à rechercher et à améliorer les méthodes de traitement des données textuelles , ce qui est important pour l'extraction d'informations de manière simple et efficace. Dans ce chapitre nous avons présenté les étapes les plus importants pour nettoyer les données textuelles ainsi que les définir méthodes de représentation de texte.

## *Chapitre 03 Conception du système*

---

### 3.1 Introduction

Dans une expérience d'analyse de texte, vous devez généralement ; nettoyer et prétraiter un jeu de données de texte ; extraire des vecteurs de fonctions numériques du texte prétraité ; former le modèle d'apprentissage et noter et valider le modèle. Dans ce chapitre on va expliquer l'architecture détaillée de notre système explication, quelques processus, la méthode utilisée pour la transformation de données textuelles, enfin l'algorithme choisi pour l'implémentation.

### 3.2 Conception globale

Dans cette section, nous allons essayer de donner un vue général de notre système.

#### 3.2.1 Architecture générale

Dans notre système, nous traiterons les données textuelles qui seront divisées en deux parties (données d'entraînement et données de test) et passeront par la phase de prétraitement (\*), comme illustré à la *figure 3.2.1*. Ensuite, nous créons un modèle d'apprentissage qui prend les entrées de données traitées à l'étape (\*) et s'entraîne sur une partie des données (données d'entraînement). Le système final consistera à insérer un nouveau document textuel et à en extraire les mots les plus importants.

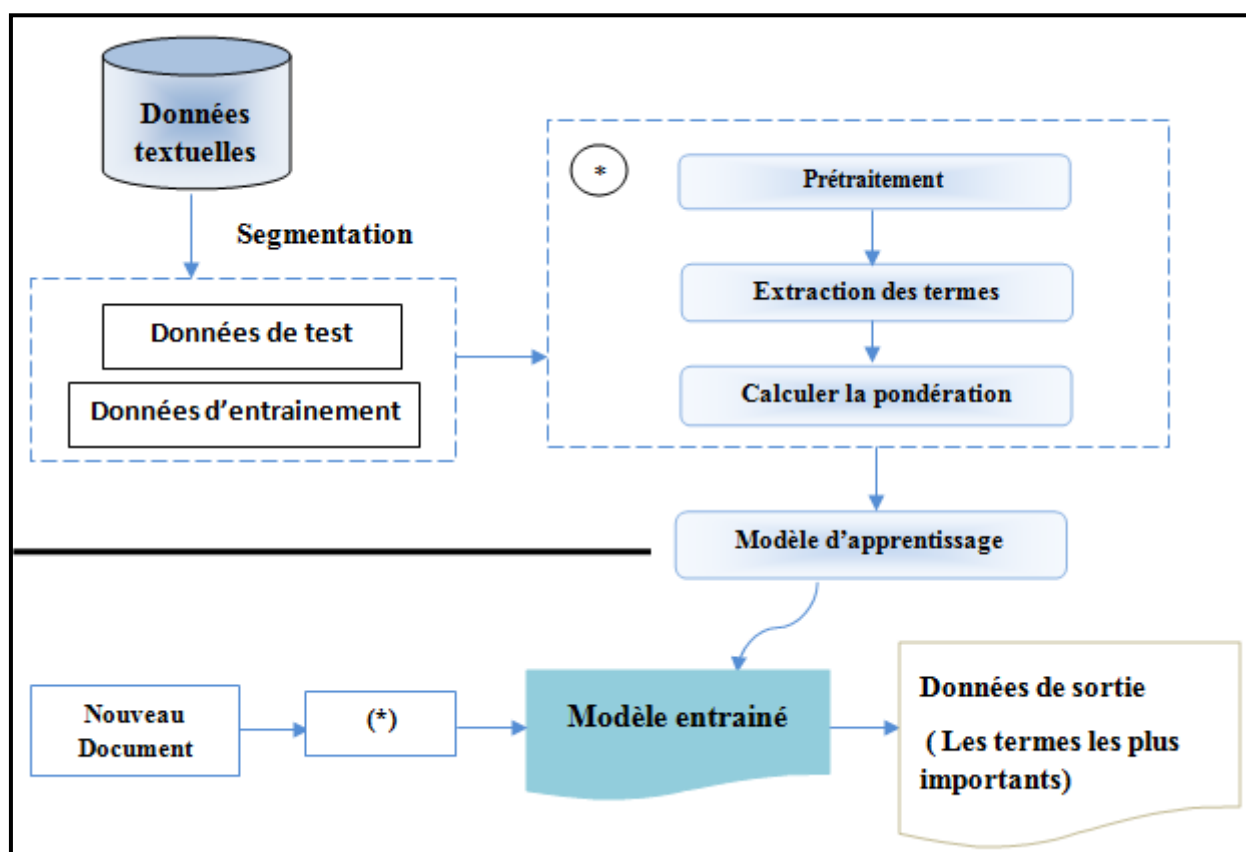


Figure 3.2.1 : Représentation de l'architecture générale du système.

### 3.3 Conception détaillée

Dans cette section, nous allons essayer d'expliquer l'architecture détaillée de notre système.

#### 3.3.1 Architecture détaillée

L'architecture détaillée de notre système illustré à la *figure 3.3.1*, et on va l'expliquer dans la section suivante.

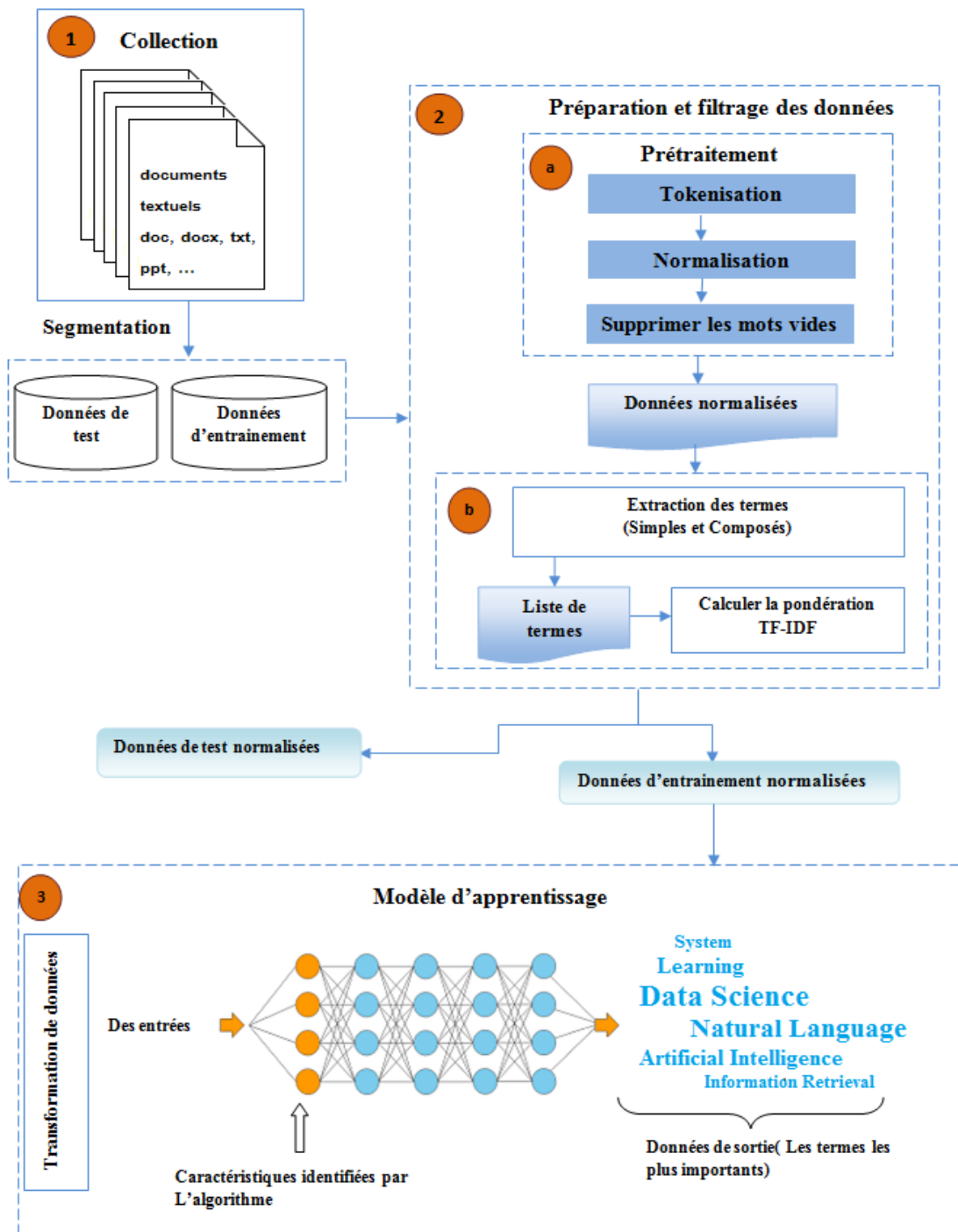


Figure 3.3.1 : Représentation de l'architecture détaillée de système.

### 3.3.2 Explication

#### Phase 1: Collection de données

Il s'agit de créer un modèle à base d'un corpus de documents, ce corpus est partitionné en deux ensembles : l'ensemble d'apprentissage et celui de test.

Cela consiste, en premier temps, à entraîner ce modèle avec l'ensemble d'apprentissage, et une fois appris, nous testerons son efficacité avec l'ensemble test. Dans certains cas, nous terminerons ce processus par une étape de validation du modèle avec un ensemble de nouveaux documents.

#### Corpus( documents textuels)

L'entraînement d'un modèle en apprentissage profond dépend de la collecte du plus grand nombre de données (collection, corpus ou Dataset en anglais), de plus de 2000 documents textes de différents domaines, de la technologie, du sport, etc.

#### Phase 2: Préparation de données

##### Phase 2.a Prétraitement

Chaque document de la collection de données va passer par les 3 étapes (Tokenisation, normalisation, la suppression de Stopword) et on va obtenir des résultats comme illustre la figure 3.3.2.a.

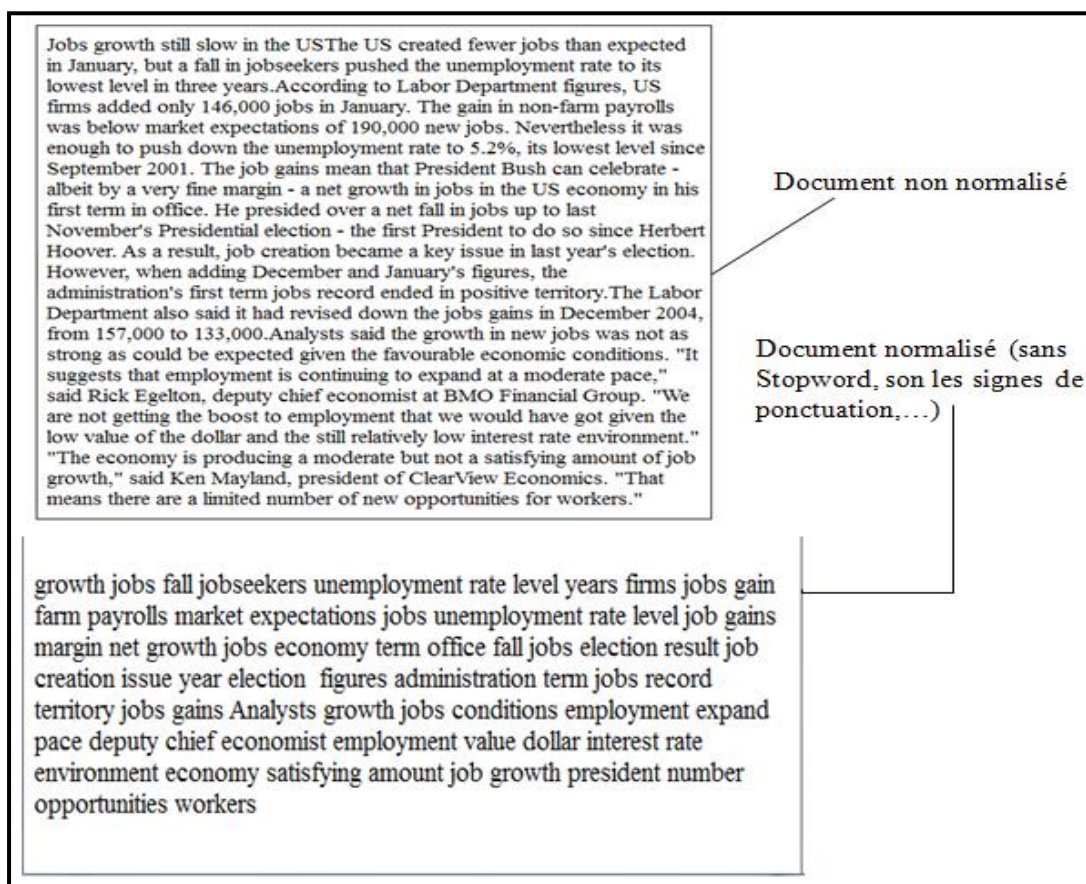


Figure 3.3.2.a : Exemple d'un document normalisé (sans Stopword).



### Phase 2.b L'extraction des termes importants

A partir des documents textuels nettoyés et normalisés, on va obtenir les termes (simples et composés) extraits de la collection de données ( la *figure 3.3.2.b*).

Les termes-clés sont les mots ou les expressions poly-lexicales qui représentent le contenu principal d'un document. Ils sont utiles pour diverses applications, telles que l'indexation automatique ou le résumé automatique, mais ne sont pas toujours disponibles. De ce fait, nous nous intéressons à l'extraction automatique de termes-clés et, plus particulièrement, à la difficulté de cette tâche lors du traitement de documents appartenant à certaines disciplines scientifiques.[3]

```

trade
rule
meeting
nation
status
finance
minister
policy
subsidy
compete
reform
organisation
account
world
order
integration
economy
. . .
globalisation
term
part
group
third
population
conference
exchange
rate
country
shock
channel

```

*Figure 3.3.2.b : Illustration de la liste de termes extraits.*

Enfin, des termes-clés candidats y sont ensuite sélectionnés, puis ordonnés par importance et enfin, les  $k$  plus importants sont sélectionnés en tant que termes-clés.

### Calculer la pondération TF-IDF

Pour ordonner les termes par importance, on va calculer la pondération TF-IDF des termes de la collection à partir des formules ci-dessous. et créer la matrice Docs-termes (la *figure 3.3.2.c*)

$$TF(i, j) = \frac{\text{occurrence de term}(i) \text{ dans doc}(j)}{\text{nombre de terms dans le document } (j)}$$

$$IDF(i) = \log\left(\frac{N}{\text{nombre de documents citent le term}(i)}\right) \quad N: \text{ le nombre totale de documents}$$

$$tfidf = Tf * IDF$$

	topic1	topic2	topic3
akz	0.00	0.00	1.00
ds	0.00	1.00	0.00
glp	0.57	0.00	0.43
lega	0.00	1.00	0.00
pcs	0.00	0.00	1.00
pdv	1.00	0.00	0.00
pev	1.00	0.00	0.00
pls	1.00	0.00	0.00
prd	0.21	0.79	0.00
pss	0.00	0.78	0.22
pst	1.00	0.00	0.00
udc	0.00	1.00	0.00
udf	0.49	0.51	0.00
verts	0.00	0.00	1.00

Figure 3.3.2.c : Exemple matrice Docs-termes avec la pondération *tf-idf*.

Chaque terme a une pondération supérieure aux autres pondérations, ce qui sera important dans le document auquel il appartient.

A fin de cette étape, nous avons obtenu des données prêtes pour entraîner du modèle d'apprentissage.

### Phase 3: Modèle d'apprentissage

La dernière étape du cadre d'apprentissage consiste à former un modèle à l'aide des fonctionnalités créées à l'étape précédente (étape de la préparation des données).

Le modèle d'apprentissage profond contient principalement trois types de couches:

- la couche d'entrée,
- plusieurs couches cachées,
- et la couche de sortie.

#### Phase 3.a: Transformation de données

Comme tous les modèles d'apprentissage nécessitant une transformation de données dans la couche d'entrée,

Comme nous le savons déjà, les machines ou les algorithmes ne peuvent pas comprendre les caractères / mots ou les phrases, ils ne peuvent prendre que des nombres en entrée qui incluent également des fichiers binaires. Mais la nature inhérente des données textuelles est non structurée et bruyante, ce qui rend impossible toute interaction avec les machines[20].

Les performances et la précision des algorithmes d'apprentissage automatique et d'apprentissage approfondi dépendent fondamentalement du type de technique d'ingénierie de caractéristiques utilisée[20].

Donc, pour le codage de données il doit appliquer le petit processus ci-dessous ( la figure 3.3.2.d). pour chaque document nettoyé.

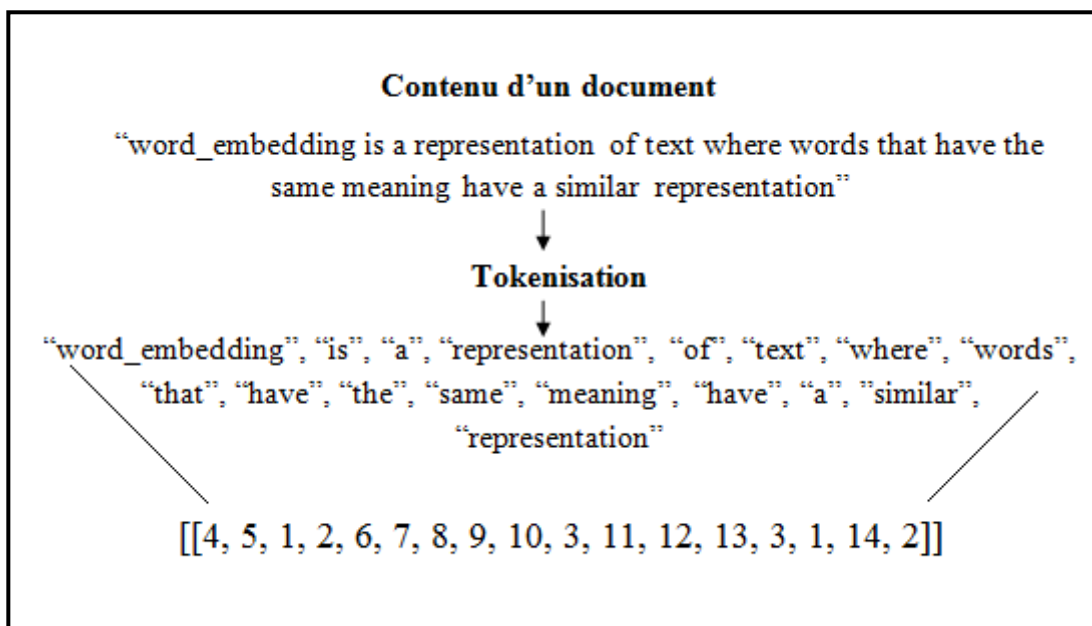


Figure 3.3.2.d illustration de transformation de données.

### Phase 3.b: Codage de données

Dans chaque tâche du traitement de langage naturel, la première et la plus importante étape est de parvenir à une représentation efficace du texte.

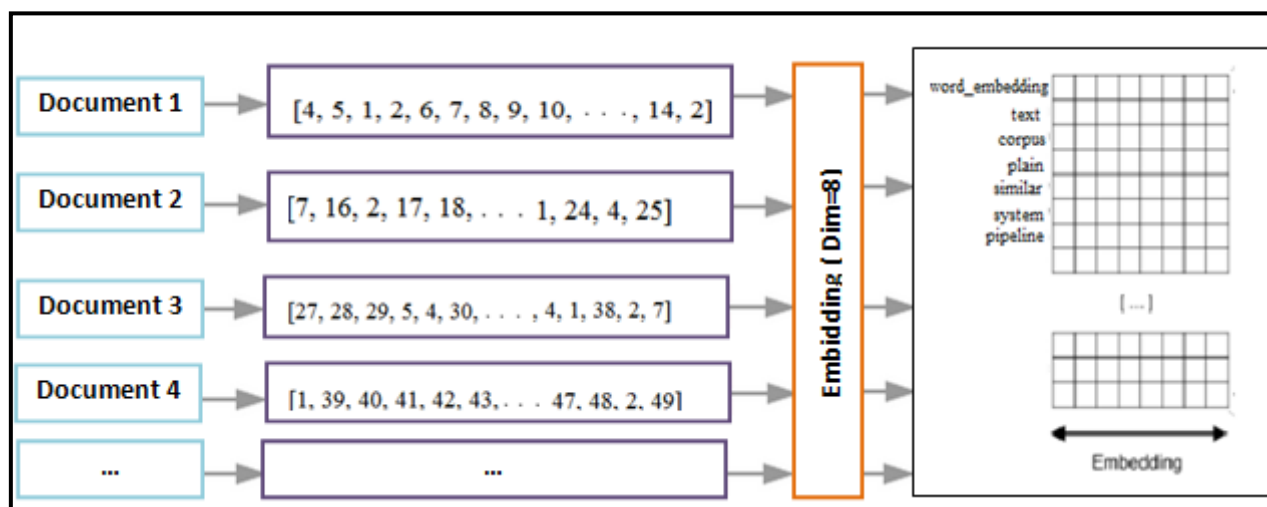


Figure 3.3.2.e représentation de texte.

## Embedding

### Définition

Avant de pouvoir modéliser un problème à travers un algorithme de Deep Learning, il est souvent nécessaire d'effectuer un bon nombre de transformations sur les données. par le choix de l'algorithme utilisé. la première couche apprend à transformer des séquences d'index de mots en vecteurs d'inclusion de mots au cours du processus d'apprentissage, de sorte que chaque index de mot soit mappé sur un vecteur dense de valeurs réelles.

**Exemple**

**Output dim=4**

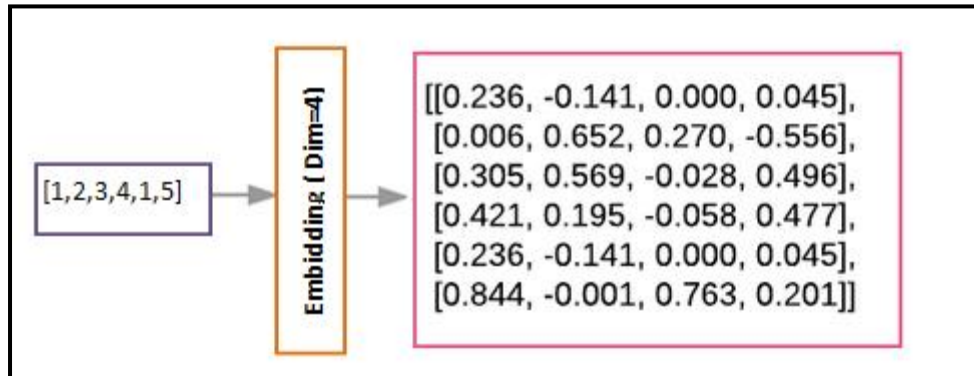


Figure 3.3.2.f Exemple d'Embedding.

**Phase 3.c Choix l'algorithme**

Il existe de nombreux choix de modèles d'apprentissage profond qui peuvent être utilisés pour former un modèle final. Nous allons implémenter l'algorithme LSTM (Long Short-Term Memory en anglais)

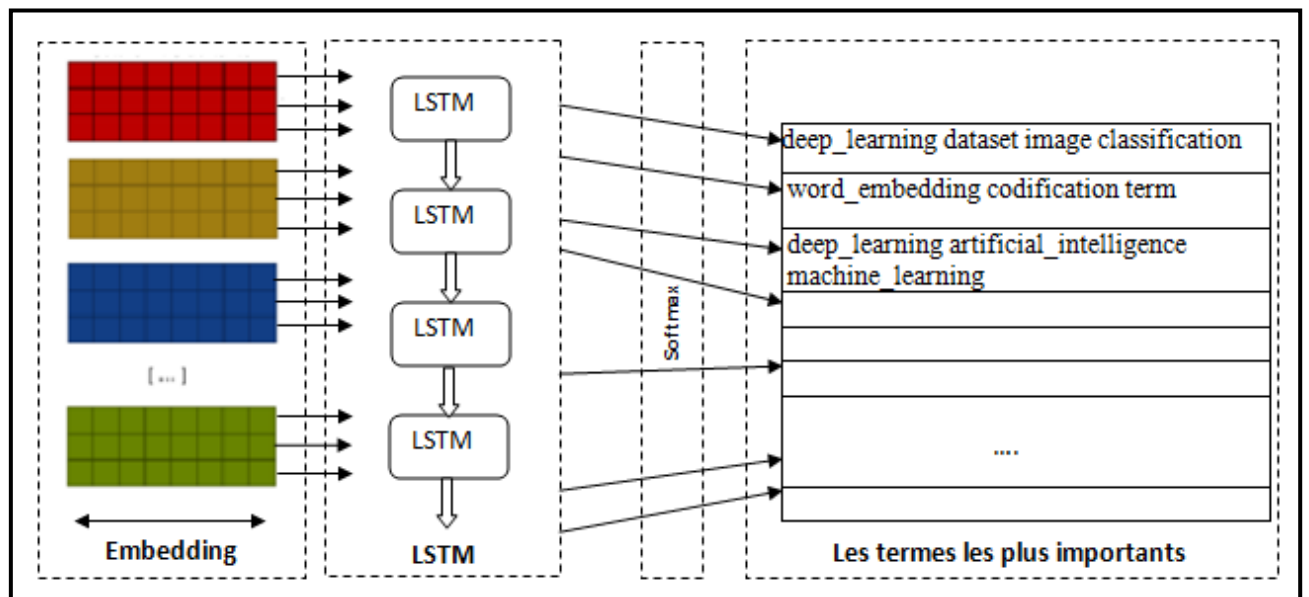


Figure 3.3.2.g illustration détaillé du modèle d'apprentissage créée.

**3.4 Conclusion**

L'étape la plus importante pour atteindre les résultats souhaités en utilisant l'apprentissage profond est la bonne connaissance de la sélection des données appropriées et la meilleure façon de les représenter. Dans ce chapitre, nous allons exposer la conception de notre système avec l'explication de méthode de transformation de notre données ainsi que l'algorithme d'apprentissage choisi.

## *Chapitre 04 Implémentation*

---

## 4.1 Introduction

Il ne fait aucun doute que le plus important pour résoudre un problème dans le domaine d'informatique est un bon choix pour le langage de développement. Le développement du domaine d'informatique est devenu un problème multiple, et l'accès à la résolution des méthodes les plus simples est très important. Nous avons donc choisi le langage de programmation optimal qui nous a considérablement aidés à obtenir de bons résultats. Dans ce chapitre on va définir le langage choisi pour résoudre notre problème, l'environnement de développement, et on va détailler l'implémentation de chaque partie de notre application.

## 4.2 Choix du langage de programmation

Lorsque vous choisissez un langage de programmation qui se spécialise dans l'apprentissage profond, il doit considérer les compétences répertoriées dans les offres d'emploi actuelles ainsi que les bibliothèques disponibles dans différentes langues qui peuvent être utilisées pour les processus d'apprentissage profond. Python est le langage de programmation le plus recherché dans le domaine de l'apprentissage automatique et l'apprentissage profond. Python est suivi par Java, puis par le R, puis C ++. Python [35].

### Python (langage)

Python est un langage de programmation, interprété car, avant de pouvoir les exécuter, un logiciel spécialisé se charge de transformer le code du programme en langage machine, multi-paradigme et multiplateformes, est placé sous une licence libre. qui vous permet de travailler rapidement et d'intégrer les systèmes plus efficacement. Python peut être utilisé pour gérer des données volumineuses et effectuer des calculs complexes. Il existe ce qu'on appelle des bibliothèques 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.



*Figure 4.2 Logo en langage python.*

Ce choix a été motivé par les raisons suivantes :

- L'une des principales langues parmi les langues appropriées pour la programmation de problèmes d'apprentissage profond.
- Il dispose un grand nombre de bibliothèques pour le traitement du langage naturel, telles que NLPnet, NLTK, ... .
- Un langage simple, productif et utilisable dans presque tous les domaines et systèmes.

### 4.3 L'environnement de développement

Pour l'environnement de développement, nous avons utilisé l'environnement du PyCharm(version Community 2019.1.1).



Figure 4.3 l'environnement du PyCharm.

#### 4.3.1 Le prétraitement d'un document

Le prétraitement est faite à l'aide la bibliothèque NLTK

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

La figure (ci-dessous) représente le processus de prétraitement d'un document avec quelques fonctions prédéfinies en python pour le traitement du langage naturel.

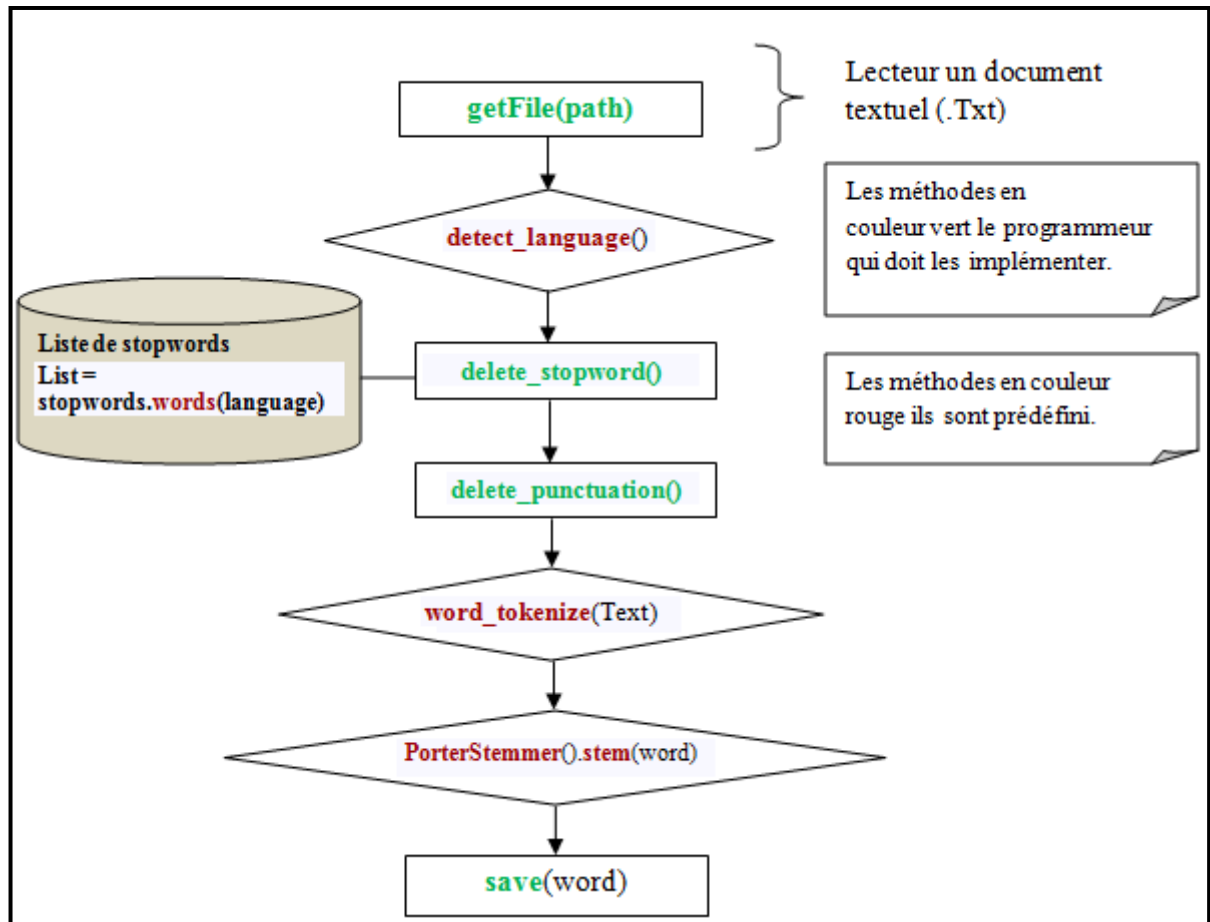


Figure 4.3.1 processus de l'exécution du prétraitement d'un document (.Txt)

### 4.3.2 Les étapes de la création du modèle

Le processus de création d'un modèle d'apprentissage doit suivre les étapes suivantes (figure 4.3.2.a), en commençant par le chargement des données jusqu'à la validation du modèle.

```

    ~~~~~ Processus de Code ~~~~~

    ==> import libraries
    ==> Dataset Preparation
        -loading a dataset
        -pre-processing
        -split the dataset (training / validation)
    ==> Converting text to features
    ==> Build Model Training
    ==> Training Model
    ==> Save Model
  
```

Figure 4.3.2.a processus de la création du modèle d'apprentissage.



### 4.3.2.a Importer les bibliothèques

Avant de créer le modèle, nous devons importer un ensemble de bibliothèques disponibles dans le package **Keras** en python. Tel que :

```
from keras.models import Sequential
from keras.layers import Dense, Embedding, Activation
```

### 4.3.2.b Importer de données

Comme mentionné dans les chapitres précédents, le processus d'entraînement un modèle d'apprentissage profond nécessite une grande quantité de données.

On a utilisé la collection de données disponibles en (Source [41]), et on a s'appliquer le processus mentionné dans la section précédent (la *Section 4.2.1*) et passe à l'étape de segmentation de données.

**Pseudo-code d'apprentissage un modèle**

---

```

// S est un ensemble,
Découper S en 2 parties S1, S2 (S1=80% de S ; S2=20% de S)
Construire un modèle M avec l'ensemble S1  //(*)
Evaluer une mesure d'erreur e de M avec S2
Si la valeur d'erreur e est grande faire
    Changer quelques paramètres de l'algorithme et retour à (*)

```

*Figure 4.3.2.b processus d'entraînement un modèle.*

### 4.3.2.c Transformation de données

A l'aide les deux bibliothèques en Keras ci-dessous on a transformer notre données

```
from tensorflow.python.keras.preprocessing import sequence
from tensorflow.python.keras.preprocessing import text
```

### 4.3.2.d La création du modèle

L'étape la plus importante de notre projet est l'étape de la création du modèle, la *figure 4.3.2.d* ci-dessous représente des quelques instructions de la création du modèle.

Pour obtenir un taux de reconnaissance du modèle plus élevé il doit changer quelques paramètres à chaque itération d'entraînement et vérifier le taux, tel que le nombre de couches cachés ...

```
EMBEDDING_DIM=4
print("Build Modele ...")

model =Sequential()
model.add(Embedding(200, 100, input_length=500))

...

model.add(Activation('sigmoid'))
model.add(Dense(3, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Figure 4.3.2.c illustration des quelques instructions de la création du modèle.

#### 4.4 Présentation de la fenêtre d'application

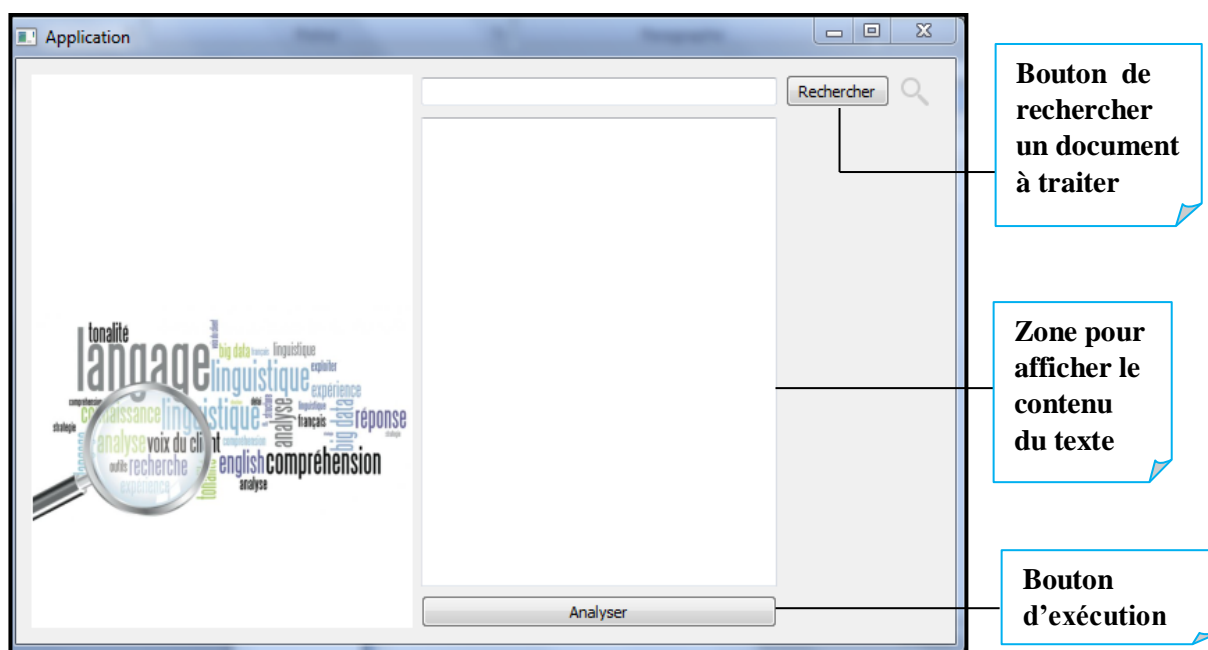


Figure 4.4 fenêtre d'application.

#### 4.5 Conclusion

Les modèles actuels d'apprentissage profond reposent sur un bon traitement des données et sur le choix de l'algorithme approprié, mais les spécialistes ne sont pas encore parvenus à définir des paramètres statiques ni un modèle général permettant de résoudre tous les problèmes similaires, mais ils ont atteint le soi-disant apprentissage par transfert. Ce chapitre, nous avons expliqué comment nous traitons des données textuelles volumineuses et sélectionnons un modèle qui convient à nos données.

## 4.6 Annexes

Dans cette Section on va cite les packages et les bibliothèques utilisés dans l'implémentation.

Nous allons utilisé Python la version 3.5

- Pour le traitement du langage naturel nous allons utilisé (Python Natural Language Processing) la grande bibliothèque NLpnet

Command d'installer en python pip install nlpnet et ses dépendances

**Nltk**

Pip install nltk

Nltk.download()

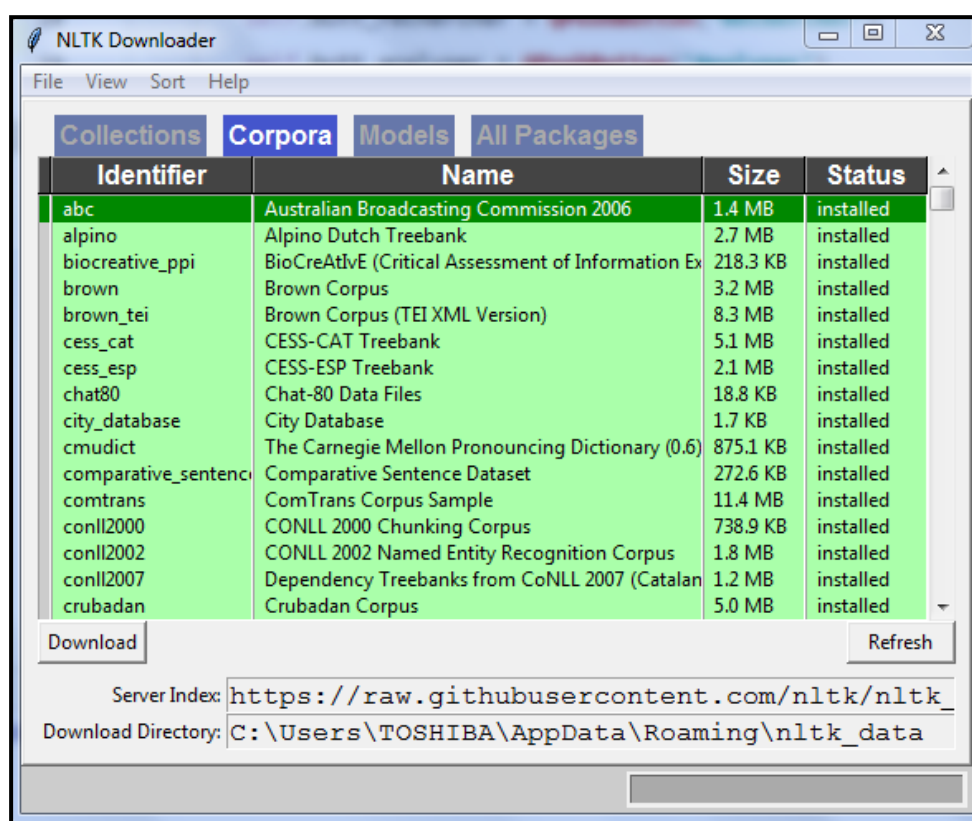


Figure 4.6 dépendances de la bibliothèque NLTK.

**Numpy** pip install numpy

- Deep Learning en Python

Les liens suivants et les installations de paquets de base fourniront à l'utilisateur l'environnement nécessaire à l'apprentissage en profondeur.

pip install keras(<http://keras.io/>)

```
pip install Tensorflow(http://tensorflow.org/)
```

```
pip intall scikit learn
```

- Pour l'interface graphique en Python

```
Pip install PyQt5
```

```
Pip install PyQt5-tools
```

## *Conclusion générale*

L'apprentissage profond a révolutionné la plus récente ces dernières années, particulièrement son entrée dans plusieurs domaines, a réalisé le progrès significatif dans tous les domaines et, particulièrement avec le traitement de langage naturel, nous ne parlons pas de l'analyse de sentiment et l'extraction d'informations très précisément ..., sans mentionner profondément l'apprentissage et comment effectif il est pour des développeurs pour atteindre ce niveau de développement scientifique.

Les techniques d'apprentissage profonds ont été appliqués avec succès à de nombreuses tâches du le traitement de langage naturel , conduisant à des systèmes efficaces, mais parfois à la taille du réseau (nombre accru de couches et de neurones) et au temps de formation sont interdits pour une utilisation efficace. Malgré cet inconvénient, il y a encore peu de travaux de recherche sur les moyens de trouver des moyens plus efficaces de former un réseaux de neurones profonds ou de trouver une structure optimale (sans former des centaines de réseaux différents) [8].

Ce mémoire a passé en revue d'utilisation de l'apprentissage profond avec le traitement du langage naturel pour extraire les termes les plus importants des documents textuels, notamment les problèmes de langage naturel, la difficulté de les analyser et l'extraction du contenu le plus important des textes.

Parce que l'apprentissage profond est un domaine en constante innovation, il est important de garder à l'esprit que les algorithmes, les méthodes et les approches continueront de changer.

# Références

- [1] MATALLAH H, 2011. *Classification Automatique de Textes Approche Orientée Agent*, Mémoire de Magister. Université Aboubekr BELKAID-Tlemcen.
- [2] Yoon Kim, 2014. *Convolutional Neural Networks for Sentence Classification* New York University.
- [3] Adrien Bougouin, Florian Boudin, Béatrice Daille, 2014. *Influence des domaines de spécialité dans l'extraction de termes-clés*. Marseille.
- [4] Yoav Goldberg, 2015. *A Primer on Neural Network Models for Natural Language Processing*.
- [5] Andreas V., Douglas T., Milco W., Jeremy A., Katie G., Lei X., Holly W., 2017. *Final Report on Natural Language Processing*. Version: 1.0 (final)
- [6] Satchee Nene, 2017 *Deep Learning for Natural Language Processing*. International Research Journal of Engineering and Technology (IRJET), Dept. of Computer Engineering, VESIT, Maharashtra, India.
- [7] Silvia F., Eric S., Juan M., Torres M., 2007. *Énergie textuelle de mémoires associatives*.
- [8] Christopher W., Gerasimos S, Gerhard W., *Flexible Deep Neural Network structure with application to Natural Language Processing*. Department of Knowledge Engineering, Maastricht University, The Netherlands.
- [9] Ahammad Fahad, Abdulsamad Ebrahim Yahya, *Inflectional Review of Deep Learning on Natural Language Processing*. International Conference on Smart Computing and Electronic Enterprise. (ICSCEE2018) (IEEE).
- [10] Annina S., Mahima S, S. Venkatesan<sup>3</sup>, D.R. Ramesh Babu, *An Overview of Machine Learning and its Applications*. International Journal of Electrical Sciences & Engineering (IJESE).
- [11] Fateme Behzadi, *Natural Language Processing and Machine Learning: A Review*. International Journal of Computer Science and Information Security. (IJCSIS)
- [12] Diksha K., Aditya K., Kiran K., Sukhdev S., *Natural Language Processing: State of The Art, Current Trends and Challenges*.
- [13] Ronan C., Jason W., *A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning*.
- [14] Andrew W. Trask, 2019. *grokking Deep Learning*.
- [15] Alexis C., Holger S., Yann Le Cun, 2016. *Very Deep Convolutional Networks for Natural Language Processing*.
- [16] Vincent Bouchet, 2017. *Mémoire de Master Machine learning en fnance*.

# Références

- [17] Yoann D., Marco D., Isabelle T., 2017. *Réseaux neuronaux profonds pour l'étiquetage de séquences*.
- [18] Jiwei L. , Xinlei C., Eduard H., Dan J., *Visualizing and Understanding Neural Models in NLP*.
- [19] Gobinda G. Chowdhury, *Natural Language Processing* . Dept. of Computer and Information Sciences, University of Strathclyde, Glasgow, UK
- [20] Akshay K., Adarsha S., *Natural Language Processing Recipes Unlocking Text Data with Machine Learning and Deep Learning using Python*
- [21] Taweh Beysolow, *Applied Natural Language Processing with Python Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing*
- [22] Richard S., Yoshua B., Christopher D. Manning, *Deep Learning for NLP*.
- [23] Vincent B., 2017. *Machine Learning en Finance Vers de nouvelles stratégies*.
- [24] Josh Patterson & Adam Gibson , 2017. *Deep Learning A Practitioner's Approach*, 1ère (Ed), O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472 , Mike Loukides & Tim McGovern, 532 p, (pp. 28)
- [25] Palash G., Sumit P. & Karan J., 2018. *Deep Learning for Natural Language Processing: Creating Neural Networks with Python*, India , Celestin Suresh John, 290p, (pp. 1)
- [26] Yann LeCun, Yoshua Bengio & Geoffrey Hinton., may 2015 *REVIEW Deep learning*
- [27] Ronan C., Jason W., Léon B., Michael K., Koray K. , Pavel K., 2011. *Natural Language Processing (almost) from Scratch*
- [28] De <https://link.springer.com/article/>.
- [29] De <https://interstices.info/la-revolution-de-lapprentissage-profond/>.
- [30] De <https://www.gralon.net/articles/materiel-et-consommables/materiel-informatique-et-consommable-informatique/article-comprendre-le-deep-learning-en-6-questions-10956.htm> .
- [31] De <https://www.saagie.com/fr/blog/qu-est-ce-que-le-deep-learning/> .
- [32] De <https://fr.mathworks.com/discovery/deep-learning.html> .
- [33] De <https://stackabuse.com/text-classification-with-python-and-scikit-learn/>
- [34] De <https://www.zdnet.com/article/what-is-deep-learning-everything-you-need-to-know/>
- [35] De <https://www.supinfo.com/articles/single/6041-machine-learning-introduction-apprentissage-automatique>
- [36] De <http://www.fullai.org/short-history-artificial-intelligence/> .

## *Références*

- [37] De [https://www.oreilly.com/library/view/python-natural language/](https://www.oreilly.com/library/view/python-natural-language/).
- [38] De <http://blogshells.com/how-deep-learning-and-data-science-work-with-natural-language-processing/>
- [39] De <http://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning/>
- [40] De <https://www.gralon.net/articles/materiel-et-consommables/materiel-informatique-et-consommable-informatique/article-comprendre-le-deep-learning-en-6-questions-10956.htm>.
- [41] De <https://www.kaggle.com/13nnys/useful-text-preprocessing-on-the-datasets>