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 3/M2/2019

Mémoire

présenté pour obtenir le diplôme de master académique en

Informatique

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

L'utilisation du Deep Learning pour l'extraction du contenu des pages web

---

Par :

MADOUI SOUMIA

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

ZERARKA Med Faouzi	M.C.A	Président
MEADI Med Nadjib	M.C.B	Rapporteur
FEKRAOUI Farah	M.A.A	Examinateur

Dédicace

Je dédie cette thèse à . . .

mes parents,

mes frères & mes sœurs,

tous mes enseignants tout au long de mes études,

tous ma famille,

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

Remerciement

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

Tout d'abord, j'exprime ma gratitude à mon encadreur « Mr. MEADI Mohamed Nadjib » pour ses conseils et orientations en dépit d'un emploi du temps chargé.

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

Enfin, Je remercie chaleureusement ma famille, tous mes amis et collègues qui m'ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire, à tous ceux qui m'ont était une source d'aide ou de motivation importante.

Merci à tous et à toutes…

Résumé

Le problème de l'extraction de contenu est un sujet d'étude depuis le développement du World Wide Web. Son objectif est de séparer le contenu principal d'une page Web, tel que le texte d'un article, du contenu bruyant, tel que les publicités et les liens de navigation.

La plupart des approches d'extraction de contenu fonctionnent au niveau des blocs, c'est-à-dire que la page Web est segmentée en blocs, puis chacun de ces blocs est déterminé à faire partie du contenu principal ou du contenu bruyant de la page Web.

Dans ce projet, On essaye d'appliquer l'extraction de contenu à un niveau plus profond, à savoir les éléments HTML. Au cours de la thèse, On va approfondir la notion de contenu principal, créer un ensemble de données de pages Web dont le contenu a été étiquetés manuellement comme faisant partie du contenu principal ou du contenu bruyant par le web scraping, et on va appliquer l'apprentissage profond (réseau de neurone à convolution) à cet ensemble de données afin de induire un modèle pour séparer le contenu principal et le contenu bruyant. Enfin, ce modèle induite va être évalué à l'aide d'un ensemble de données différent constitué de pages Web étiquetées manuellement par le web scraping aussi.

Mots clés : L'extraction de contenu, apprentissage profond, réseau de neurone à convolution, web scraping, contenu principal, contenu bruyant.

تلخيص

مشكلة استخراج المحتوى هي موضوع الدراسة منذ تطوير الشبكة العالمية. الغرض منه هو فصل المحتوى الرئيسي لصفحة الويب، مثل نص المقال، عن المحتوى الصاخب، مثل الإعلانات وروابط التنقل.

تعمل معظم أساليب استخراج المحتوى على مستوى الكتلة، أي يتم تقسيم صفحة الويب إلى كتل، ثم يتم تحديد كل من هذه الكتل على أنها جزء من المحتوى الرئيسي أو المحتوى الصاخب لصفحة الويب.

في هذا المشروع، نحاول تطبيق استخراج المحتوى على مستوى أعمق، أي على عناصر ال HTML. خلال الأطروحة، سنقوم بتعميق مفهوم المحتوى الرئيسي، وإنشاء مجموعة بيانات لصفحات الويب التي تم تصنيف محتواها يدويًا كجزء من المحتوى الرئيسي أو المحتوى الصاخب عن طريق كشط الويب، ثم تطبيق التعلم العميق (شبكة الخلايا العصبية التلافيفية) لهذه البيانات لتعيين نموذج لفصل المحتوى الرئيسي والمحتوى الصاخب. أخيرًا، يتم تقييم هذا النموذج باستخدام مجموعة مختلفة من البيانات التي تتكون من صفحات ويب يتم تمييزها يدويًا عن طريق تجريف الويب أيضًا.

الكلمات المفتاحية : استخراج المحتوى، التعلم العميق، شبكة الخلايا العصبية التلافيفية، تجريف الويب، المحتوى الرئيسي، المحتوى الصاخب.

Abstract

The problem of content extraction is a subject of study since the development of the World Wide Web. Its goal is to separate the main content of a web page, such as the text of an article, from the noisy content, such as advertisements and navigation links.

Most content extraction approaches operate on the block level, that is, the web page is segmented into blocks, and then each of these blocks is determined to be part of the main content or the noisy content of the Web page.

In this project, we try to apply the content extraction at a deeper level, namely to HTML elements. During the thesis, we investigate the notion of main content more closely, create a dataset of web pages whose elements have been manually labeled as either part of the main content or the noisy content by the web scraping, then we apply the deep learning (convolution neural network) to this data set in order to induce a model for separating the main content and the noisy content. Finally, this induced model is going to be evaluated by a different dataset of manually labeled Web pages using the web scraping also.

Key words: Content extraction, deep learning, convolution neural network (CNN), web scraping, main content, noisy content.

TABLE DES MATIÈRES

Table des matières 01
Liste des Figures 04
Liste des Tableaux 06
Introduction générale 07
Chapitre 01 : Extraction de contenu des pages web 10
I. Introduction 11
II. Extraction de contenu des pages web 11
1. Définition du contenu principal 11
2. Définition du contenu bruyant 12
3. Le Web 13
4. Recherche d'information sur le Web 14
5. Analyse des pages web (Web Mining) 15
5.1. Catégories du Web Mining 17
5.1.1. Exploration de contenu Web 17
5.1.2. Exploration de structure Web 18
5.1.3. Exploration de l'usage Web 18
6. Extraction du contenu web (Web Scraping) 18
6.1. Processus de Web Scraping 19
6.2. Modalités d'extraction 19
6.3. Techniques d'extraction 20
6.3.1. Expression régulière 20
6.3.2. XPath 20
6.3.3. Traverser le DOM. 21
6.3.4. Analyse HTML 21
6.4. Web Scraping et HTML 21

7. Travaux Reliés 22

III. Conclusion 25
Chapitre 02 : Apprentissage profond 26
I. Introduction 27
II. Apprentissage automatique 27
1. Types d'apprentissage automatique 28
1.1. Apprentissage supervisé. 28
1.2. Apprentissage non supervisé 28
III. Apprentissage profond (Deep Learning) 29
1. Historique 30
2. Réseaux de Neurones Artificiels 32
2.1. Neurone biologie 32
2.2. Réseau de Neurones Artificiel (RNA) 33
2.2.1. Définition 33
2.2.2. Neurone formel 33
2.2.3. Fonction d'activation 34
2.2.4. Types de réseaux de neurones 35
2.2.4.1. Réseaux de neurones non bouclés 35
2.2.4.2. Réseaux de neurones bouclés 36
2.2.5. Architecture d'un réseau de neurones artificiel 36
2.2.5.1. Réseaux monocouche 37
2.2.5.2. Réseaux multi-couches 37
2.2.5.2.1. Réseaux multicouche classique 37
2.2.5.2.2. Réseau à connexions locales 38
2.2.5.2.3. Réseau à connexions récurrentes 38
2.2.6. Apprentissage des réseaux de neurones 39
2.2.7. Les Limites de Réseaux de neurones 39
3. Apprentissage Profond 40
4. Architectures d'apprentissage profond 41
4.1. Réseau de Neurone à Convolution 41
4.1.1. Principe d'architecture d'un CNN 41
4.1.2. Les couches de CNN 42
4.1.2.1. La couche de convolution (CONV) 42
4.1.2.2. Couche de pooling 43
4.1.2.3. Couche entièrement connectée 44
4.2. RNN (Recurrent Neural Networks) 44
4.3. LSTM (Long Short Term Memory networks) 45
IV. Conclusion 46

Chapitre 03 : Conception 47
I. Introduction 48
II. Conception globale du système 48
III. Conception détaillée du système 49
1. La phase de préparation de données 49
1.1.Prétraitement de données 50
1.1.1. Web scraping 50
1.1.2. Supprimer les mots vides 51
1.1.3. Lemmatisation 51
1.1.4. Stemming 52
1.2. Word2Vector 52
1.2.1. Approche par sac-de-mots continus (CBOW) 53
1.2.2. Approche par Skip-Gram 53
2. Apprentissage en profondeur (CNN) 54
IV. Conclusion 55
Chapitre 04 : Implémentation 56
I. Introduction 57
II. Langage et l'environnement de la programmation 57
III. La création de la base 58
1. Web scraping 60
2. Préparation de données pour l'apprentissage 63
2.1. Prétraitement de données 63
2.1.1. Convertir les lettres en minuscules 63
2.1.2. Supprimer les mots vides 63
2.1.3. Lemmatisation 64
2.1.4. Stemming 64
2.2. Word2Vector 64
3. Apprentissage de CNN 66
4. Résultats Expérimentaux 68
IV. Conclusion 71

Conclusion générale 72
Bibliographie 74

LISTE DES FIGURES

1 Principe de fonctionnement de l'architecture Client-Serveur 14
2 Le processus d'extraction de connaissances à partir de données 15
3 Une proposition de décomposition du processus de Web Mining 16
4 Web mining catégories 17
5 Un exemple de la courbe de pente du document 23
6 Processus d'apprentissage automatique 27
7 Schéma d'un modèle supervisé 28
8 Schéma d'un modèle non supervisé 29
9 Illustration de l'accroissement d'intérêt pour les réseaux de neurones 31
10 Schéma d'un neurone biologique 32
11 Comparaison entre le neurone biologique et le neurone artificiel 33
12 Neurone formel 34
13 Fonction d'activation seuil 35
14 Fonction d'activation linéaire 35
15 Fonction de sigmoïde 35
16 Réseaux de neurones non bouclés 36
17 Réseaux de neurones bouclés 36
18 Réseau de neurones monocouche 37
19 Réseau multicouche classique 38
20 Réseau à connexion locale 38
21 Réseau à connexions récurrentes 39
22 La différence de performance entre le Deep Learning et la plupart des algorithmes de machine learning en fonction de la quantité de données. 40
23 Comparaison entre l'apprentissage automatique et le deep learning 40
24 Les réseaux de neurones convolutifs 41
25 Une illustration simple de l'opération de convolution à deux dimensions. . 42

26 Exemples du kernels appliqués sur la même image 43
27 Le pooling 43
28 Pooling 44
29 Un réseau de neurones à convolution qui reçoit une image 2D comme entrée 44
30 Un réseau de neurones récurrents 45
31 Conception globale du système de classification 49
32 Le processus de prétraitement 50
33 Le processus de création de la base d'apprentissage (et le même processus sera utilisé pour créer la base de test) 51
34 Modèle général de CBOW 53
35 Modèle général de Skip-Gram 54
36 Modèle général de classification 55
37 Une page web Origine 59
38 Une page web Annoté 59
39 Télécharger le contenu HTML d'une page Web 60
40 Le contenu HTML de la page web 60
41 L'extraction de contenu principal 61
42 Classe main extraite à partir de la page numéro 7 61
43 La base d'apprentissage extraite 62
44 La base de teste extraite 62
45 Les classes 63
46 La base d'apprentissage en minuscule. 63
47 Tokenization des données. 63
48 Suppression des mots vides. 64
49 La lemmatisation 64
50 Stemming. 64
51 Codage de la base d'apprentissage. 65
52 Le modèle Word2Vector. 65
53 Codage de classes. 66
54 Méthode de codage de classe. 66
55 Le modèle de CNN. 67
56 Le résultat du modèle CNN utilisé. 68
57 La méthode fit pour entrainer le modèle. 69
58 L'entrainement du CNN avec epochs=10. 69
59 L'entrainement du CNN avec epochs=15. 70

60 L’entrainement du CNN avec epochs=20. 71

LISTE DES TABLEAUX

Tableau.1 Analogie entre le neurone biologique et le neurone artificiel 33

INTRODUCTION GÉNÉRALE

Introduction générale

Les pages Web (également appelées documents Web) sont les unités de base pour construire le World Wide Web, elle contient des catégories d'informations très diverses. Chaque catégorie d'informations peut avoir différents formats (tels que texte, image, audio, vidéo ...).

Une page Web est composée de «parties» distinctes, qu'on va les appeler le contenu de la page Web. On peut séparer le contenu de la page web en deux classes: Le contenu principal qui représente la partie d'information utile de la page web, et le contenu bruyant qui contient la source d'informations non pertinents de la page web (tels que les publicités et les barres de navigation...). Le processus d'identification du contenu principal d'une page Web est appelé l'extraction du contenu.

L'extraction de contenu principal de la page Web est très utile pour diverses applications. Pour la plupart des pages Web, un utilisateur humain peut identifier facilement et rapidement le contenu principal. Cependant, du point de vue du balisage HTML, le contenu principal et le contenu bruyant sont étroitement imbriqués. Par conséquent, leur séparation représente un grand défi pour les extracteurs automatiques d'informations.

On va faire l'extraction du contenu à manière automatique en utilisant l'apprentissage profond, L'apprentissage profond est un apprentissage réalisé sur un réseau de neurones avec plusieurs couches cachées, il vise à entraîner un système pour qu'il résolve des situations sans que tous les paramètres nécessaires à la résolution du problème n'aient été calculés par le programmeur, afin de prendre une décision correcte concernant le problème donné.

L'objectif de notre travail est de réaliser un système capable de faire l'extraction de contenu principal en utilisant l'apprentissage profond. On va considérer la tâche d'extraction de contenu en tant que problème de classification, qui est un problème courant traiter par l'apprentissage profond. On va construire un modèle capable de séparer le contenu de la page web (contenu principal ou bruit) à l'aide d'algorithme de l'apprentissage profond (réseau de neurone à convolution) basé sur un ensemble de données de pages Web dont le contenu a été étiquetés manuellement comme faisant partie du contenu principal ou du contenu bruyant en utilisant le web scraping. De plus, la performance de modèle construit doit être évaluée avec une base de test.

Organisation du mémoire :

Ce mémoire a été organisé en quatre chapitres :

- le premier chapitre, sera consacré à la description du domaine d'extraction de contenu du page web et ces techniques ainsi que les travaux connexes.
- Dans le deuxième chapitre, on va présenter les notions de base du réseau de neurone et ces différentes architectures et on va essayer de définir le domaine d'apprentissage profond et spécialement les réseaux de neurones à convolution.
- Dans le troisième chapitre, On va exposer la conception globale et détaillée de notre système.
- Dans le dernier chapitre, On va montrer la partie expérimentale de notre travail, discuter les différents résultats obtenus et on va terminer avec une conclusion générale.

CHAPITRE

1

Extraction de contenu des pages web

I. Introduction

Avec le développement du World Wide Web, l'Internet est devenu la source d'information la plus importante. Lorsqu'on parcourt une page Web, une masse d'informations non pertinentes, telles que les publicités, les navigations non pertinentes sont incluses. Ces informations collectivement non pertinentes représentent non seulement un lourd fardeau pour les utilisateurs, mais génèrent également des problèmes pour les applications manipulant ces pages Web tels que les moteurs de recherche. Le processus d'identification du contenu principal d'une page Web est appelé ***extraction du contenu principal*** ou, plus brièvement, ***extraction du contenu***.

La plupart d'utilisateur des pages web recherchent le contenu principal et ne souhaitent généralement pas le contenu non pertinent. Il y a plusieurs applications pour faire l'analyse et l'extraction de contenu pertinent des pages web. Pour l'analyse du page web, on a ***L'exploration Web*** (Web Mining) et pour l'extraction, on a le ***Web scraping. L'exploration Web*** est l'application de techniques d'exploration de données (Data Mining) permettant de découvrir et d'extraire automatiquement des connaissances à partir des pages Web. Et le ***Web scraping*** est une technique d'extraction du contenu de pages Web, via un script ou un programme, Cette technique se concentre principalement sur la transformation de données non structurées (format HTML) sur le web en données structurées (base de données ou fichier csv).

II. Extraction de contenu des pages web

L'objectif de l'extraction de contenu des pages web est de séparer le contenu principal d'une page Web (tel que le texte), du contenu bruyant (tel que les publicités et les liens de navigation…). Le contenu des pages web peut être présenté sous déférents formes par exemple (textes, images, vidéos…etc). L'extraction du contenu principal d'une page Web est très utilisée par plusieurs applications. L'une des applications est ***l'exploration web*** (en anglais :***web Mining***) et On a aussi est le ***web scraping***…

1. Définition du contenu principal

Le contenu principal est présenté comme «la partie d'une page Web qui rend la page Web intéressante pour l'utilisateur», mais cette définition est plutôt vague. Il est difficile de donner une définition formelle précise.Le problème est que les utilisateurs peuvent avoir des intérêts différents dans la page Web.Par exemple, de nombreux utilisateurs préfèrent lire uniquement le résumé de l'article, en ignorant le corps de l'article. De nombreux utilisateurs sont également intéressés par les liens vers des articles connexes fournis par la page Web, que la plupart des algorithmes d'extraction de contenu classent comme non principaux.

À cause de la difficulté d'identifié le contenu principal de la page web, on va le considéré comme les parties qui ne sont pas bruyants dans la page web. La justification de cette définition est que le contenu bruyant est plus facile à définir que le contenu principal [32].

2. Définition du contenu bruyant

Le contenu bruyant d'une page Web est constitué de tous ses contenus qui ne sont pas principaux. Cependant, les contenus bruyants peuvent être subdivisés en types distincts. Dans la liste suivante, nous essayons de fournir une catégorisation exhaustive de tous les contenus possibles d'une page Web qui ne seront pas considérés comme principaux.

Publicité(Advertisement) : C'est le type de contenu bruyant le plus évident. De nombreuses pages Web incluent des publicités payantes de produits commerciaux, qui sont parfois liées au sujet de la page Web (marketing ciblé).

Navigation : La plupart des sites Web incluent un menu de navigation (ou une barre). Il consiste en des liens vers certaines pages Web (généralement importantes ou fréquemment consultées) du site Web, telles que la page d'accueil.

PagesWebPromotionnelles : Il s'agit notamment de liens vers des pages Web autres que la page Web actuelle. Les liens peuvent faire référence à :

- des pages Web sur le même sujet que la page Web actuelle ou sur un sujet similaire.
- les pages Web qui sont actuellement à la mode; qui est fréquemment lu, partagé ou commenté.

Les pages Web référées peuvent résider sur le même site Web que la page Web d'origine ou sur un autre site Web.

Informationslégales : Cette catégorie comprend des contenus tels que les avis de droits d'auteur et les avis de confidentialité.

Informationsnonpertinentes : Certaines pages Web incluent des informations supplémentaires, telles que les prévisions météorologiques ou les indices boursiers, qui peuvent être liées ou non au sujet de la page Web.

Sourcesetréférences : Certaines pages Web fournissent une liste des sources d'informations qu'elles contiennent ou des références pour une lecture ultérieure.

Elémentsd'entrée : Ce sont les éléments qui reçoivent des entrées de la part de l'utilisateur, telles que les zones de texte et les cases à cocher. Cette catégorie comprend également des éléments permettant à l'utilisateur d'effectuer toute action, tels que les boutons J'aime, Partager, Imprimer et Envoyer. Bien que ces éléments puissent être importants pour l'utilisateur, mais il a été décidé de les traiter comme un contenu non principal, car l'extraction de contenu traite les pages Web en tant que sources d'informations et ne traite pas de leur aspect interactif [32].

Cette liste est utile (en termes d'extraction de contenu) car chaque catégorie peut être facilement identifiée par un observateur humain. Par exemple, il est facile de décider si un certain contenu appartient ou non à la catégorie de publicité.

3. Le Web

Le Web a été inventé entre les années **1989-1991** par **Tim Berners Lee** qui à cette époque a travaillé au CERN (Centre Européen de la Recherche Nucleaire, ou laboratoire européen pour la physique des particules) en Suisse. Le Web (ou World Wide Web, WWW, W3, Toile) est un système hypertexte public contenant des documents liés entre eux par des hyperliens permettant de passer automatiquement d'un document à l'autre. Selon CERN, leWorld Wide Web est défini comme une "initiative de recherche d'informations hypermédia à grande surface visant à donner un accès universel à un vaste ensemble de documents". En d'autres termes, c'est la plus grande source d'information qui est facilement accessible et consultable. Il se compose de milliards de documents interconnectés (appelés pages Web) qui sont rédigés par des millions de personnes, dont l'accès à ses documents est très simple en utilisant un réseau mondial appelé Internet [25].

Un utilisateur s'appuie sur un programme (appelé client) pour se connecter à une machine distante (appelée serveur) où les données sont stockées. La navigation à travers le Web se fait au moyen d'un programme client appelé navigateur, par exemple : Netscape, Internet Explorer, Firefox,... etc. Les navigateurs Web envoient des requêtes à des serveurs distants en utilisant L'URL (en anglais : Uniform Resource Locator), et il affiche le contenu sur l'écran du côté client. Les fichiers tombent dans quelques types principaux:

- **HTML** : contient le contenu principal de la page.
- **CSS** : ajoutez un style pour rendre la page plus jolie.
- **JS** : Les fichiers Javascript ajoutent de l'interactivité aux pages Web.
- **Images** : Les formats d'image, tels que JPG et PNG, permettent aux pages Web d'afficher des images.

Fig1 : Principe de fonctionnement de l'architecture Client-Serveur [25].

Une fois que tous les fichiers ont été reçus par notre navigateur, il rend la page et nos l'affiche. Et lorsque nous effectuons au l'extraction du contenu Web, nous intéressons au contenu principal de la page Web donc nous examinons le code HTML.

HTML (HyperText MarkupLanguage)

L'**HTML** est un langage informatique utilisé sur l'internet. Ce langage est utilisé pour créer des pages web. L'acronyme signifie *HyperText Markup Language*, ce qui signifie en français "*langage de balisage d'hypertexte*". Cette signification porte bien son nom puisqu' effectivement ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage.

4. Recherche d'information sur le Web

La Recherche d'informations dans le Web (RIW) a sa racine dans la recherche d'information (RI) classique. Sur le Web, les documents sont des pages Web. Il est évident de dire que la recherche dans le Web est l'application la plus importante de la RI [25, 34].

La Recherche d'Information (RI) peut être définie comme une activité dont la finalité est de localiser et de délivrer un ensemble de documents à un utilisateur en fonction de son besoin en informations. L'opération de la RI est réalisée par des outils informatiques appelés Systèmes de Recherche d'Information (SRI), ces systèmes ont pour but de mettre en correspondance une représentation du besoin de l'utilisateur (requête) avec une représentation du contenu des documents au moyen d'une fonction de comparaison (ou de correspondance).

Généralement, la recherche documentaire passe par les étapes suivantes :

- L'analyse des besoins d'information.
- La préparation de la recherche en cernant le sujet et la formulation de la requête de recherche traduisant les besoins d'information.
- Le choix des outils de recherche les plus convenable.
- Le lancement de la recherche et le traitement des résultats obtenus [34, 35].

5. Analyse des pages web (Web Mining)

En **1996**, c'est **Etzioni** qui a inventé le terme «web mining». Selon **Oren Etzioni*Web Mining*** est l'utilisation de techniques d'exploration de données (Data Mining) qui permet de découvrir et d'extraire automatiquement des informations utiles à partir de documents et de services World Wide Web [29]. ***Data Mining*** désigne un processus d'extraction des connaissances à partir de grandes quantités de données après les collectes, les nettoyées, les traitées, et les analysées.

Le processus d'exploration de données est de manière générale composé de plusieurs étapes comme illustré sur la Figure 2.

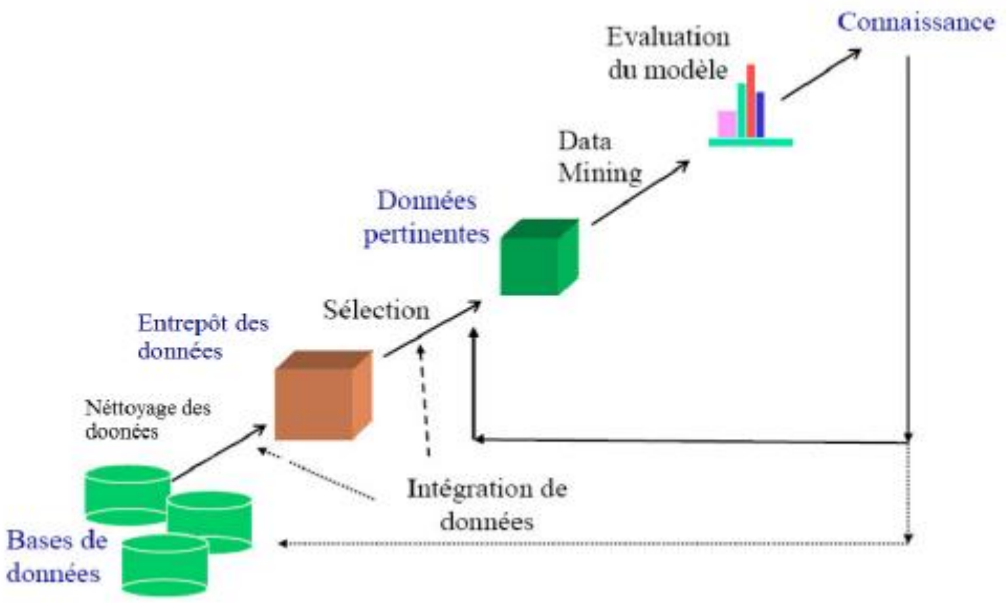


Fig2 : Le processus d'extraction de connaissances à partir de données [25].

- **Nettoyage des données** : Les données que nous avons collectées à partir de toutes les sources ne sont pas propres et peuvent contenir des erreurs, des valeurs manquantes, des données bruitées ou incohérentes. Par conséquent, nous devons appliquer différentes techniques pour nous débarrasser de ces anomalies.
- **Intégration de données** : est un outil capable d'extraire les données dans différentes sources, de les adapter et de les charger dans un entrepôt de données, les données sont collectées et combinées à partir de sources de données ayant chacune une structure et une définition de données distincte.
- **Sélection** : Dans cette étape, les données pertinentes pour la tâche de datamining à accomplir sont sélectionnées, ces données sont stockées dans des bases de données.
  Nous sélectionnons uniquement les données que nous pensons utiles pour l'exploration de données.

- **Transformation** : les données même après le nettoyage ne sont pas prêtes pour l'extraction. Il faut les transformer en formes approprié à l'exploration. Les techniques utilisées sont le lissage, l'agrégation, la normalisation, la généralisation… etc.
- **Datamining** : Dans cette étape, nous sommes prêts à appliquer des techniques d'exploration de données aux données pour découvrir des modèles intéressants. Il existe plusieurs techniques appliquées sur les données comme : le regroupement (clustering), la classification et l'analyse d'association ...
- **Evaluation du modèle** : Cette dernière étape identifie les modèles intéressants parmi les modèles générés, en se basant sur des mesures d'intérêt et sur l'avis de l'expert. La connaissance découverte peut ainsi être prise en compte dans un processus de prise de décision.
- **Connaissance** : Cette étape est bénéfique car il est utile d'utiliser les connaissances acquises pour prendre de meilleures décisions.

Similaire à **Kosala et Blockeel**, **Qingyu Zhang et Richard s. Segall** suggère de décomposer l'exploration Web en sous-tâches suivantes:

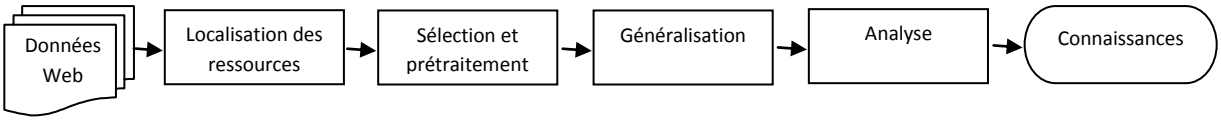


Fig3 : Une proposition de décomposition du processus de Web Mining [25].

Localisation des ressources: Cette tâche comprend principalement la représentation des documents, l'indexation et la recherche et récupération des données, qui sont soit en ligne ou hors ligne, à partir des différentes ressources sur le Web. Ces ressources peuvent être des articles de presse, des forums, des blogs et le contenu textuel des documents HTML obtenus en supprimant les balises HTML, ...etc.

Sélection d'informations et prétraitement: extraire et prétraiter automatiquement des informations spécifiques à partir de ressources Web récemment découvertes. Lorsque les documents ont été récupérés le défi consiste à extraire automatiquement des connaissances et d'autres informations requises sans intervention humaine. Un système de prétraitement robuste est nécessaire pour extraire tout type de connaissances à partir d'une grande collection des données non structurées.

Généralisation: découverte de modèles généraux sur des sites Web individuels et sur plusieurs sites.

Analyse: L'analyse est un problème fondée sur les données qui suppose qu'il existe suffisamment de données disponibles, afin que des informations potentiellement utiles puissent

être extraites et analysées. Puisque le Web est un média interactif, les humains jouent un rôle important dans le processus de découverte d'informations ou de connaissances sur le Web. Ce rôle est particulièrement important pour la validation et/ou l'interprétation des modèles (connaissances) extraits qui ont lieu dans cette phase.

Visualisation: Présenter les résultats d'une analyse interactive de manière visuelle, facile à comprendre [25, 29].

Kosala et Blockeel, qui effectuent des recherches dans le domaine de l'exploration Web et suggèrent les trois ***catégories d'extraction Web*** en fonction du type de données à extraire.

5.1. Catégories du Web Mining

L'exploration Web est divisée en trois catégories principales en fonction du type de données: exploration de contenu Web (Web Content Miningen anglais), exploration de structure Web (Web Structure Miningen anglais) et exploration d'utilisation Web (Web Usage Mining en anglais).

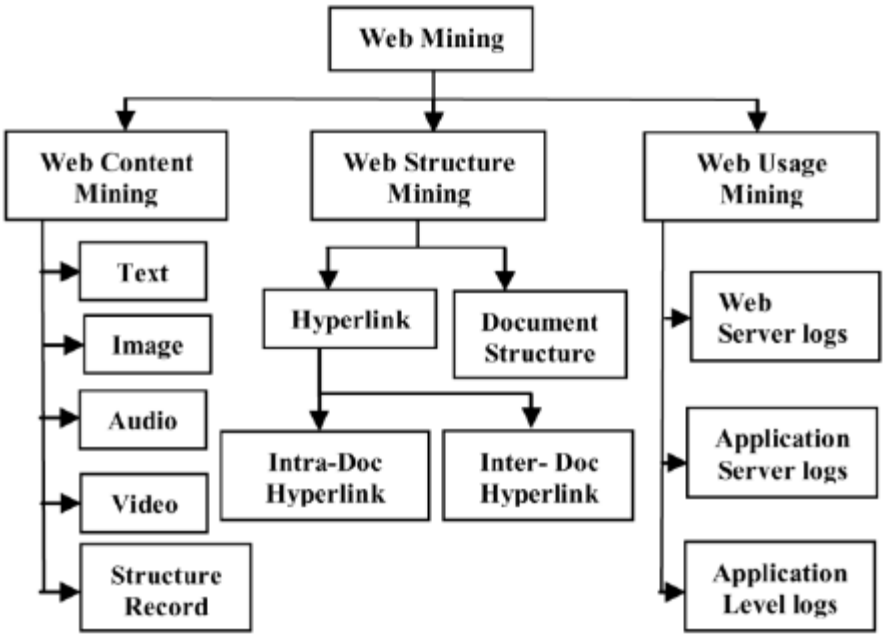


Fig4 : Web mining catégories [29].

5.1.1. Exploration de contenu Web (Web Content Mining)

Il s'agit de découvrir des informations utiles ou des connaissances à partir du contenu des pages Web. L'exploration de contenu Web analyse le contenu des ressources Web. Ce type d'exploration met en œuvre des techniques de traitement automatique du langage naturel « *natural language processing (NLP)* » et de recherche d'information « *Information Retrieval (IR)* ». La ***recherche d'information*** est l'un des domaines de recherche proposant une

gamme de méthodes populaires et efficaces, principalement statistiques, pour l'exploration de contenu Web.

L'exploration de contenu Web est une forme de L'exploration de texte appliquée au domaine Web. En règle générale, un document Web comprend des données telles que du texte, de l'audio, de la vidéo, des images et des hyperliens. La recherche dans l'exploration de contenu Web englobe la découverte de ressources à partir du Web, la catégorisation et la classification de documents et l'extraction d'informations à partir de pages Web [29, 30].

5.1.2. Exploration de structure Web (Web Structure Mining)

La structure de Web est un graphe typique consiste en des pages Web en tant que nœuds et des hyperliens en tant qu'arêtes reliant des pages connexes, basé sur la topologie des hyperliens et générer les informations, telles que la similitude et la relation entre différents sites Web. L'exploration de structure Web est le processus de découverte d'informations sur la structure à partir du Web pour identifier les documents pertinents [30].

5.1.3. Exploration de l'usage Web (Web Usage Mining)

L'exploitation de l'usage du Web consiste à déterminer ce que les utilisateurs recherchent sur Internet. En analysant ces flux de clics, on cherche à découvrir des informations utiles sur le comportement de l'utilisateur lors de la navigation sur le Web [8]. Il existe trois phases d'exploration de l'utilisation du Web. Les trois phases sont [9]:

- **Prétraitement**: il aide à extraire les données brutes des ressources Web et traite ensuite les données.
- **Découverte de modèles**: après le prétraitement des données, celles-ci sont utilisées pour la découverte de modèles.
- **Analyse de modèle**: après avoir découvert le modèle, celui-ci est analysé puis vérifié. Si le modèle est correct, il est mis en œuvre sur le Web pour extraire les informations de ce dernier.

6. Extraction du contenu web (Web Scraping)

Le ***Web Scraping*** est une technique permettant d'extraire des données du Web et de les enregistrer dans un système de fichiers (CSV, JSON ou XML) ou une base de données pour les récupérer ou les analyser ultérieurement. Généralement, les données Web sont grattées à l'aide du protocole HTTP (Hypertext Transfer Protocol) ou via un navigateur Web. Ceci est accompli

soit manuellement par un utilisateur, soit automatiquement par un robot ou un robot d'indexation Web (***Web crawling***) [33].

6.1. Processus de Web Scraping

Le processus de récupération des données sur Internet (en anglais : The process of scraping data) peut être divisé en deux étapes séquentielles. *Acquérir des ressources Web*, puis *extraire les informations souhaitées* à partir des données acquises. Plus précisément, un programme de grattage Web commence par composer une requête HTTP afin d'acquérir des ressources à partir d'un site Web ciblé. Cette demande peut être formatée dans une URL contenant une requête GET ou dans un message HTTP contenant une requête POST. Une fois la demande reçue et traitée avec succès par le site Web ciblé, la ressource demandée sera extraite du site Web, puis renvoyée au programme de scrap Web. La ressource peut être sous plusieurs formats, tels que des pages Web construites à partir de HTML, des sources de données au format XML ou JSON ou des données multimédia telles que des images, des fichiers audio ou vidéo. Une fois les données Web téléchargées, le processus d'extraction continue d'analyser, de reformater et d'organiser les données de manière structurée. Il existe deux modules essentiels d'un programme de nettoyage Web: un module permettant de composer une requête HTTP, tel que ***Urllib2*** ou ***sélénium***, et un autre permettant d'analyser et d'extraire des informations de code HTML brut, telles que ***Beautiful Soup*** ou ***Pyquery***.

- ***Urllib2*** : définit un ensemble de fonctions permettant de traiter les requêtes HTTP, telles que l'authentification, les redirections, les cookies, etc…
- ***Selenium*** : est un wrapper de navigateur Web permettant de créer un navigateur Web, tel que Google Chrome ou Internet Explorer, et d'activer les utilisateurs à automatiser le processus de navigation sur un site Web par programmation.

Concerne l'extraction de données :

- ***Beautiful Soup*** : est conçu pour extraire le contenu d'un document HTML et XML. Il fournit des fonctions Pythonic pratiques pour naviguer, rechercher et modifier un arbre d'analyse; une boîte à outils pour décomposer un fichier HTML et extraire les informations souhaitées via ***lxml*** ou ***html5lib***[33].

6.2. Modalités d'extraction

On peut faire une distinction de base concernant le web scraping selon la modalité d'extraction des données :

- **Extraction manuelle** : une personne navigue le web pour extraire informations pertinentes aux intérêts depuis les pages qu'elle visite. La pratique la plus commune pour ce type d'extraction et le simple copier/coller.
- **Extraction semi-automatique** : une personne utilise un logiciel ou une application web pour aspirer/nettoyer les éléments d'une ou plusieurs pages web pertinents aux intérêts.
- **Extraction automatique** : l'extraction se fait de manière totalement automatique grâce à l'émulation par une machine d'un navigateur web qui visite des pages et qui est capable de suivre les différents liens afin de générer automatiquement un corpus de pages liées entre elles.

6.3. Techniques d'extraction

Dans le cadre d'extractions automatiques ou semi-automatiques, il est nécessaire d'identifier dans les documents analysés les données d'intérêt afin de les séparer de l'ensemble du contenu. Voici une liste non exhaustive de techniques qui peuvent être utilisées :

- Expressions régulières
- XPath
- Traverser le DOM
- Analyse HTML

6.3.1. Expressions régulières

Les expressions régulières sont une fonctionnalité disponible pratiquement dans tout langage de programmation et qui permet d'identifier des patterns à l'intérieur de contenu textuel. Grâce à une syntaxe qui permet de combiner plusieurs règles d'analyse en même temps, il est possible d'extraire de manière ponctuelle des éléments qui correspondent aux critères définis dans l'expression régulière. Le mécanisme consiste à rechercher toutes les ressemblances entre la chaîne de caractères à trouver (le pattern) à l'intérieur du contenu cible de l'analyse. Les expressions régulières sont une technique très puissante d'extraction de contenu car elles s'appliquent indépendamment de la structure du document analysé. Cette puissance nécessite cependant une syntaxe assez complexe qui n'est pas très intuitive.

6.3.2. XPath

XPath est un standard du W3C (l'organisme qui s'occupe des standards du Web) pour trouver des éléments dans un document XML. Ce langage exploite la structure hiérarchique des

nœuds (et attributs) d'un document XML et nécessite par conséquent une structure de document très précise, ce qui n'est pas forcément le cas dans les pages HTML.

6.3.3. Traverser le DOM

Cette technique est similaire à XPath car elle exploite également la structure hiérarchique d'une page web à travers le DOM (Document Object Model). Il s'agit encore une fois d'un standard W3C qui permet d'accéder au contenu des différentes balises d'une page HTML grâce à leur positionnement hiérarchique dans la page.

6.3.4.Analyse HTML

De nombreux sites Web ont de grandes collections de pages générées dynamiquement à partir d'une source structurée sous-jacente telle qu'une base de données. Les données de la même catégorie sont généralement codées dans des pages similaires par un script ou un modèle commun. Dans l'exploration de données, un programme qui détecte de tels modèles dans une source d'informations particulière, extrait son contenu et le traduit sous une forme relationnelle, est appelé un wrapper. Les algorithmes généraux de wrapper supposent que les pages d'entrée d'un système d'induction de wrapper se conforment à un modèle commun et qu'elles peuvent être facilement identifiées en termes de modèle commun d'URL. En outre, certains langages de requête de données semi-structurés, tels que XQuery peuvent être utilisés pour analyser des pages HTML et pour extraire et transformer le contenu de la page.

6.4. Web Scraping et HTML

Les difficultés dans l'extraction de données à partir d'une page web sont liées au langage HTML lui-même c'est-à-dire le contenu du page web. Le HTML5 introduit des nouvelles balises qui ont principalement un intérêt sémantique.

Les balises de structuration du contenu

Dans la perspective des développeurs de pages web il y a souvent des éléments qui sont présents dans la plupart des sites : une entête qui suggère l'argument du site, un menu de navigation qui permet d'accéder aux différentes ressources disponibles, le contenu principal de la page (c'est-à-dire ce qui la rend "unique" par rapport aux autres pages du site), et ainsi de suite. Avant l'introduction de balises spécifiques en HTML5, les développeurs utilisaient pour chacun de ces éléments récurrents une balise « **div »** à laquelle ils associaient une classe souvent avec

une relation sémantique avec le contenu, par exemple « **div class="header"** » pour l'entête, « **div class="navigation"** » pour le menu, et ainsi de suite. Cette pratique était cependant loin de représenter un standard : d'une part, chaque développeur pouvait choisir son propre nom de classe (par exemple dans sa propre langue maternelle), et de l'autre les classes sont des éléments de style et en accord avec les bonnes pratiques de développement, il faudrait séparer le contenu de la forme. Dans cette perspective, HTML5 introduit des balises telles que : ***header, nav, aside, footer, main, article, section***…

7. Travaux Reliés

De nombreux chercheurs ont développé plusieurs approches d'extraction de contenu Web basées sur différentes méthodes. Parmi ceux-ci, quelques recherches importantes sur l'extraction de contenu Web sont présentées dans cette section :

1. Ashraf F et al ont proposé un système dans lequel des techniques de regroupement (Cluster) ont été utilisées pour un Extraction d'Information automatique à partir de documents HTML contenant des données semi-structurées. Au moyen d'informations spécifiques à un domaine fournies par l'utilisateur, le système proposé a analysé et segmenté les données d'un document HTML, les a divisées en clusters comportant des éléments analogues et a estimé une règle d'extraction en fonction du modèle d'occurrence des jetons de données. Ensuite, la règle d'extraction a été utilisée pour affiner les clusters, et finalement, la sortie a été démontrée [31].

2. L'extraction du texte corps (Body Text Extraction) a été introduite et décrite par **Finn et al** comme méthode d'identification du contenu textuel principal d'une page Web, qu'ils qualifient de corps de texte principal. L'algorithme BTE est basé sur l'observation que le corps du texte d'une page Web est principalement constitué de texte et très peu de balisage.

BTE commence par affecter tous les jetons de la source HTML de la page Web dans l'une des deux catégories suivantes: jetons de balises HTML et jetons de mots. Par conséquent, la page Web est vue comme une séquence $\{B_i\}$ de bits, avec $B_i = 1$ lorsque le jeton initial est une balise et $B_i = 0$ lorsque le premier jeton est un mot. Cette séquence peut être représentée par la courbe de pente du document, comme illustré à la figure 5. Un point (x; y) situé sur la courbe nous indique essentiellement: Dans les x premiers jetons de la page Web, il y a des jetons de balise y. Par conséquent, les segments à faible pente, généralement appelés plateaux, correspondent aux parties de la source de la page Web contenant un petit nombre de balises HTML [32].

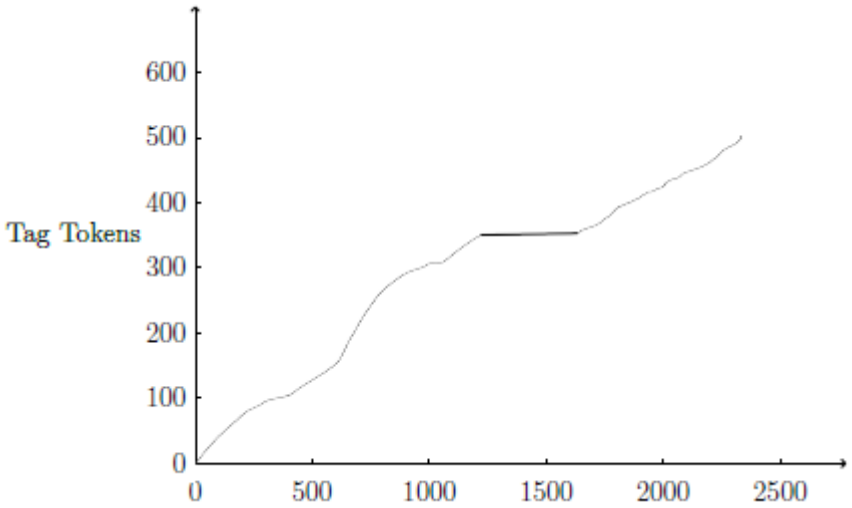


Fig5 : Un exemple de la courbe de pente du document

3. Extraction de contenu basée sur la vision (Vision-Based Content Extraction) :

Cai et al a introduit l'algorithme de segmentation de pages basé sur la vision (VIPS). Il tente de simuler l'approche d'un utilisateur humain pour comprendre la structure du contenu d'une page Web. Un utilisateur humain ne voit pas le balisage HTML ou le DOM de la page Web; Tout ce qu'elle voit, c'est le rendu visuel de la page. VIPS tente donc d'utiliser les mêmes repères spatiaux et visuels qui donnent des indications à un utilisateur humain sur la structure du contenu de la page Web.

VIPS est appliqué de manière récursive à l'arborescence DOM de la page Web. La première étape de VIPS est l'extraction par bloc. À partir du nœud racine vers le bas (initialement, le nœud racine est l'élément <html>), chaque nœud DOM est inspecté pour vérifier s'il représente un seul bloc visuel. Si tel est le cas, le bloc est ajouté à un pool de blocs. Si le nœud contient plusieurs blocs visuels, les enfants de ce nœud sont inspectés de la même manière jusqu'à ce que tous les blocs de la (sous-) page en cours soient extraits et ajoutés au pool de blocs. La question de savoir si un nœud DOM représente un seul bloc visuel ou si elle doit être divisée davantage dépend de plusieurs considérations [32].

4. Wrapper : Un wrapper est une procédure (programme) permettant d'extraire des enregistrements de base de données d'une source d'information donnée, en particulier d'une page Web. De nombreuses pages Web incluent des contenus générés de manière dynamique, obtenus à partir d'une requête sur une base de données interne, par exemple des pages Web décrivant les spécifications du produit. Les wrappers tentent de restaurer ces informations sous leur forme relationnelle. Il existe trois façons de construire des wrappers.

Codage manuel : Les wrappers peuvent être créés par une personne familiarisée avec les balises des pages Web contenant les données. Par exemple, un wrapper peut recevoir pour instruction de récupérer le contenu de certaines cellules de tableau contenant les données pertinentes.

Wrapper induction : L'apprentissage automatique supervisé est utilisé pour obtenir les règles d'extraction. Cela nécessite un ensemble de formation de web pages avec les données pertinentes étiquetées manuellement dans chaque page Web.

Extraction automatique des données : L'apprentissage automatique non supervisé est utilisé au lieu de l'apprentissage supervisé pour obtenir les règles d'extraction. Cela évite de devoir étiqueter manuellement les données dans les pages Web.

Il est noté qu'un wrapper spécifique est conçu pour une source d'informations spécifique. Les tâches d'extraction de wrappers et de contenu se chevauchent, mais elles ne sont pas identiques. La différence réside dans les données à extraire. Les wrappers recherchent des données structurées ou semi-structurées dans une page Web, qui est généralement extraite puis utilisée comme entrée dans une base de données relationnelle. Au contraire, l'extraction de contenu implique l'identification de tout le contenu principal d'une page Web, généralement constituée de données non structurées [32].

III. Conclusion

Dans ce chapitre, On a présenté une vue générale sur l'extraction de contenu des pages web et les technique utilisées pour cette extraction.

L'extraction du contenu principal est le processus d'identification le contenu principal d'une page Web, la plupart d'utilisateur des pages web recherchent le contenu principal et ne souhaitent généralement pas le contenu non pertinent de la page web. Pour faire l'extraction de contenu du page web, On peut appliquer plusieurs techniques. Parmi les techniques utilisées, On trouve le Web mining et le web scraping qu'On a détaillé dans ce chapitre.

CHAPITRE

APPRENTISSAGE PROFOND

I. Introduction

L'apprentissage automatique (en anglais : Machine Learning) est un champ d'étude de l'intelligence artificielle. Bien que l'apprentissage automatique soit un domaine de l'informatique, il diffère des approches informatiques traditionnelles. En effet dans cette dernière, les algorithmes sont des ensembles d'instructions explicitement programmées utilisées par les ordinateurs pour calculer ou résoudre des problèmes.

Par conséquent, L'apprentissage profond (en anglais: Deep Learning) est une technique d'apprentissage automatique. L'apprentissage profond utilise des algorithmes inspirés de la structure et de la fonction du cerveau, appelés Réseaux de Neurones Artificiels. En d'autres termes. Les algorithmes d'apprentissage profond s'apparentent à la manière dont le système nerveux est structuré de manière à ce que chaque neurone se connecte et transmette des informations à l'autre. Cette technique d'apprentissage automatique a considérablement amélioré les résultats dans de nombreux domaines tels que la vision par ordinateur, la reconnaissance de la parole et la traduction automatique…etc.

Dans ce chapitre On va présenter tout d'abord les notions en relation avec l'apprentissage profond.

II. Apprentissage automatique

En général, l'objectif de l'apprentissage automatique est de comprendre la structure des données et de les intégrer dans des modèles qui peuvent être compris et utilisés par tout le monde [1]. L'apprentissage automatique est 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 [5].

L'apprentissage automatique facilite l'utilisation des ordinateurs dans la construction de modèles à partir de données d'échantillonnage afin d'automatiser les processus de prise de décision en fonction des données saisies [1].

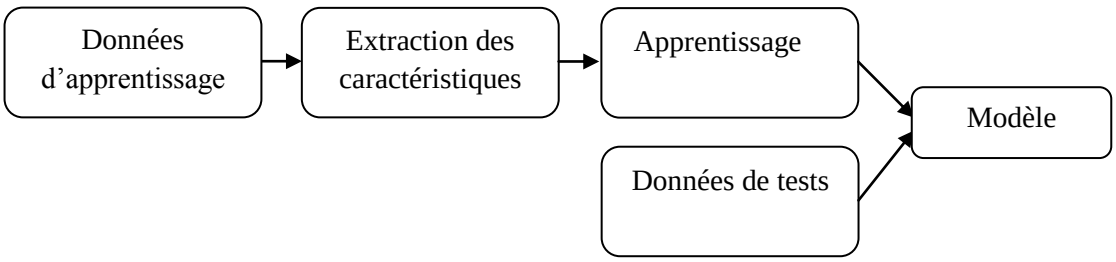


Fig6 : Processus d'apprentissage automatique [5].

1. Types d'apprentissage automatique

Les algorithmes d'apprentissage peuvent se catégoriser selon le mode d'apprentissage qu'ils emploient :

1.1. Apprentissage supervisé

Dans l'apprentissage supervisé, les classes sont prédéterminées et les exemples connus, le système apprend à classer selon un modèle de classification ou de classement. Le but de cette méthode est que l'algorithme puisse «apprendre» en comparant sa sortie réelle avec les sorties « désirées» pour trouver des erreurs et modifier le modèle résultant.

L'apprentissage supervisé utilise donc des modèles pour prédire les valeurs d'étiquettes sur des données non étiquetées [1].

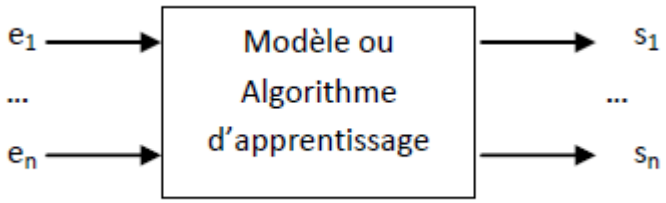


Fig7 : Schéma d'un modèle supervisé [15]

Quelques algorithmes d'apprentissage supervisé

- Machine à vecteurs de support (SVM)
- Réseau de neurones (RNA)
- Méthode des k plus proches voisins (KPP)
- Arbre de décision
- Classification naïve bayesienne… [17].

1.2. Apprentissage non supervisé

Il vise à concevoir un modèle structurant l'information. La différence ici est que les catégories ou encore les classesdes données d'apprentissage ne sont pas connus, c'est ce que l'on cherche à trouver [15].

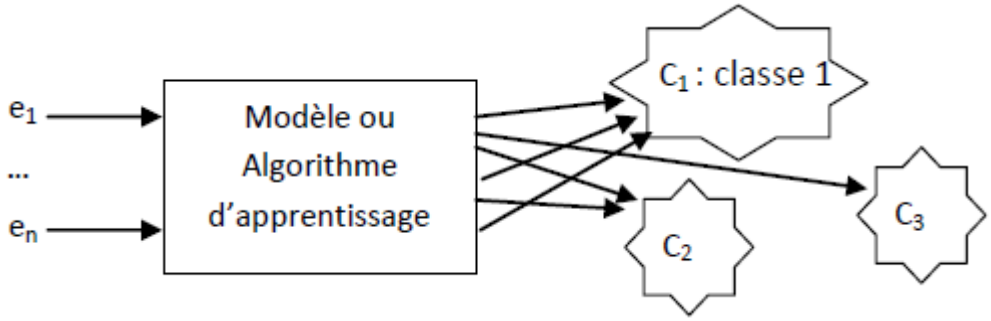


Fig8 : Schéma d'un modèle non supervisé [15].

Dans l'apprentissage non supervisé, les données sont non étiquetées, de sorte que l'algorithme d'apprentissage trouve tout seul des points communs parmi ses données d'entrée. L'objectif de l'apprentissage non supervisé peut être aussi simple que de découvrir des modèles cachés dans un ensemble de données, mais il peut aussi avoir un objectif d'apprentissage des caractéristiques, qui permet à la machine intelligente de découvrir automatiquement les représentations nécessaires pour classer les données brutes [1].

Quelques algorithmes d'apprentissage non supervisé

- Clustering (segmentation, regroupement) : construire des classes automatiquement en fonction des exemples disponibles
- Règles d'association : analyser les relations entre les variables ou détecter des associations
- Réduction de dimensions [18].

III. Apprentissage profond (Deep Learning)

Apprentissage profond est une technique d'apprentissage automatique, il vise à entraîner un système pour qu'il résolve des situations sans que tous les paramètres nécessaires à la résolution du problème n'aient été calculés par le programmeur. L'objectif est d'entraîner un algorithme avec des paramètres variables (une « boîte noire ») afin de prendre une décision correcte concernant une tâche donnée [11].

L'apprentissage profond est un apprentissage réalisé sur un réseau de neurones avec plusieurs couches cachées. Le principe d'apprentissage profond repose sur un apprentissage hiérarchique couche par couche. Entre chaque couche interviennent des transformations non linéaires et chaque couche reçoit en entrée la sortie de la couche précédente.

Il existe plusieurs manières de construire un réseau de neurones profond, notamment le CNN (Convolution Neural Network), DBN (*Deep Belie fNetwork*), RNN (Recurrent Neural Network) … [19].

1. Historique

En **1873 :** Introduction du neurone **Groupings** comme les premiers modèles de réseaux de neurones par **Alexander Bain**.

En **1943 :** les neurologues **Warren McCulloch** et **Walter Pitts** menèrent les premiers travaux sur les réseaux de neurones à la suite de leur article fondateur : " What the frog's eye tells to the frog's brain". Ils constituèrent un modèle simplifié de neurone biologique communément appelé neurone formel. Ils montrèrent également théoriquement que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes.

Les travaux de **McCulloch** et **Pitts** n'ont pas donné d'indication sur une méthode pour adapter les coefficients synaptiques. Cette question au cœur des réflexions sur l'apprentissage a connu un début de réponse grâce aux travaux du physiologiste canadien **Donald Hebb** sur l'apprentissage en **1949** décrits dans son ouvrage "The Organization of Behaviour". **Hebb** a proposé une règle simple qui permet de modifier la valeur des coefficients synaptiques en fonction de l'activité des unités qu'ils relient. Cette règle aujourd.hui connue sous le nom de «règle de Hebb» est presque partout présente dans les modèles actuels, même les plus sophistiqués.

En **1957 : Franck Rosenblatt** introduit le modèle du Perceptron. Il construit le premier neuro-ordinateur basé sur ce modèle et l'applique au domaine de la reconnaissance de formes

En **1960 : B. Widrow**, un automaticien, développe le modèle Adaline (Adaptative Linear Element). Dans sa structure, le modèle ressemble au Perceptron, cependant la loi d'apprentissage est différente. Celle-ci est à l'origine de l'algorithme de rétropropagation de gradient très utilisé aujourd'hui avec les Perceptrons multicouches.

En **1969: Marvin Lee Minsky** et **Seymour Papert** publient un ouvrage qui met en exergue les limitations théoriques du perceptron. Limitations alors connues, notamment concernant l'impossibilité de traiter par ce modèle des problèmes non linéaires. Ils étendent implicitement ces limitations à tous modèles de réseaux de neurones artificiels.

En **1974 : Paul Werbos** introduit la retro propagation.

En **1980 : TeuvoKohonen** introduit des cartes auto organisatrices et **Kunihiko Fukushima** introduit du Neocognitron, qui a inspiré les réseaux de neurone convolutif

En **1982**, **John Joseph Hopfield**, physicien reconnu, a introduit un nouveau modèle de réseau de neurones (complètement récurrent).

En **1983** : La Machine de Boltzmann est le premier modèle connu apte à traiter de manière satisfaisante les limitations recensées dans le cas du perceptron. Mais l'utilisation pratique

s'avère difficile, la convergence de l'algorithme étant extrêmement longue (les temps de calcul sont considérables).

Une révolution survient alors dans le domaine des réseaux de neurones artificiels : une nouvelle génération de réseaux de neurones, capables de traiter avec succès des phénomènes non-linéaires ; le perceptron multicouche ne possède pas les défauts mis en évidence par **Marvin Minsky**. Proposé pour la première fois par **Werbos**, le Perceptron Multi-Couche apparait en **1986** introduit par **Rumelhart**, et, simultanément, sous une appellation voisine, chez **Yann le Cun**. Ces systèmes reposent sur la rétropropagation du gradient de l'erreur dans des systèmes à plusieurs couches, chacune de type Adaline de **Bernard Widrow**, proche du Perceptron de **Rumelhart.**

En **1986 : Michael I. Jordan** définition et introduction des réseaux de neurones récurrents.

En France, elle est à l'image du congrès Neuro-Nîmes qui a pour thème les réseaux neuro-mimétiques et leurs applications. Créé en 1988, le chiffre de ses participants croit chaque année et reflète bien l'intérêt que le monde scientifique et industriel (50% des participants) porte au connexionnisme (Fig9).

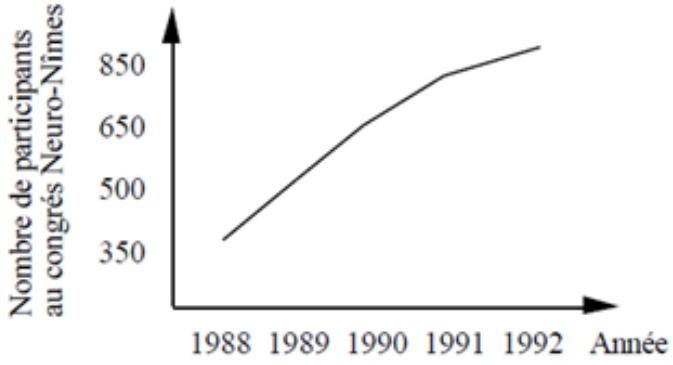


Fig9 : Illustration de l'accroissement d'intérêt pour les réseaux de neurones :
évolution du nombre de participants au congrès Neuro-Nîmes

En **1990 : Yann LeCun** introduit de LeNet et montra la capacité des réseaux de neurones profond.

1995-2005 : développement des **SVM**, perte d'intérêt pour les réseaux de neurones.

En **2006** : premières architectures **profondes** de réseaux de neurones Deep belief Network par **Geoffrey Hinton.**

En **2009 : Salakhutdinov et Hinton** introduisent des Deep Boltzmann Machines.

En **2012** : résultats en reconnaissance d'objets (Toronto, ImageNet) et de la parole (Microsoft) démontre le potentiel de technologie disruptive de l'apprentissage profond par **Alex Krizhevsky.**
En **2014** : explosion d'investissements privés en apprentissage automatique, en particulier en apprentissage profond [2,5, 6, 8].

2. Réseaux de Neurones Artificiels

Les réseaux de neurones artificiels (RNA) sont inspirés de la méthode de travail du cerveau humain qui est totalement différente de celle d'un ordinateur. Le cerveau humain se base sur un système de traitement d'information parallèle et non linéaire, très compliqué, ce qui lui permet d'organiser ses composants pour traiter, d'une façon très performante et très rapide, des problèmes très compliqués tel que la reconnaissance des formes [21].

2.1 Neurone biologie

Un neurone est une cellule du système nerveux spécialisée dans la communication et le traitement d'informations.

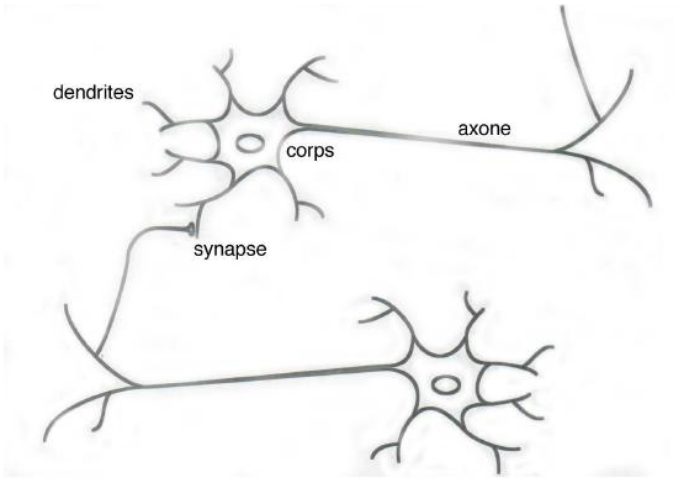


Fig10 : Schéma d'un neurone biologique [3].

Les neurones reçoivent les signaux (impulsions électriques) par des extensions très ramifiées de leur corps cellulaire (les dendrites) et envoient l'information par de longs prolongements (les axones). Les impulsions électriques sont régénérées pendant le parcours de l'axone. La durée de chaque impulsion est de l'ordre d'1 ms et son amplitude d'environ 100 mV. Les contacts entre deux neurones, de l'axone à une dendrite, se font par l'intermédiaire des synapses.

2.2. Réseau de Neurones Artificiel (RNA)

2.2.1. Définition

Un RNA (Réseau de Neurones Artificiels) est un ensemble de neurones formels associes en couches (ou sous-groupes) et fonctionnant en parallèle. Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurone samont. A chacune de ces entrées est associé un poids w représentant la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones avals [8, 10].

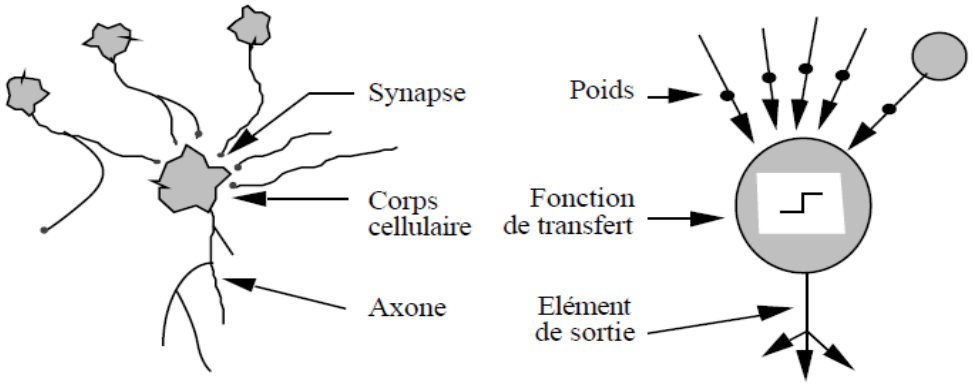


Fig11 : Comparaison entre le neurone biologique et le neurone artificiel.

Neurone biologie	Neurone artificiel
Synapses	Poids de connexions
Axones	Signal de sortie
Dendrites	Signal d'entrée
Somma	Fonction d'activation

Tableau.1. Analogie entre le neurone biologique et le neurone artificiel

2.2.2. Neurone formel

Un neurone formel (ou simplement "neurone") est une fonction algébrique non linéaire et bornée, dont la valeur dépend de paramètres appelés coefficients ou poids. Les variables de cette fonction sont habituellement appelées "entrées" du neurone, et la valeur de la fonction est appelée sa "sortie". Un neurone peut être représenté graphiquement comme indiqué sur la Fig8 [7].

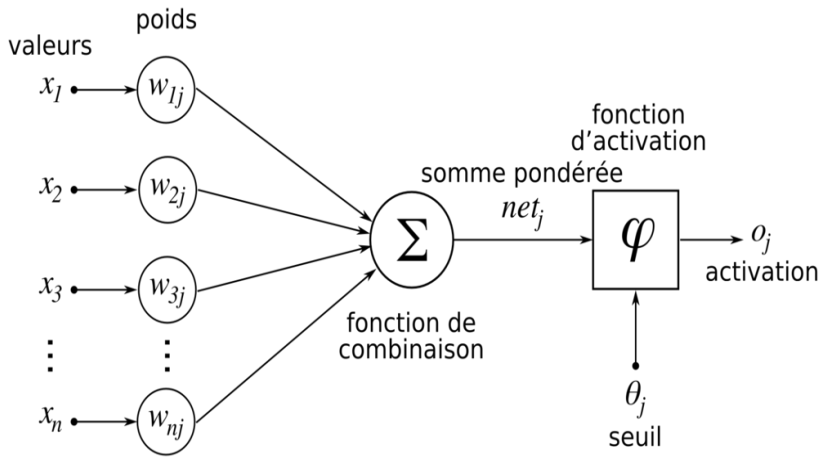


Fig12 : Neurone formel [2].

- Toutes les entrées (x_j) : sont directement les entrées du système.
- Biais ($b=w_0$) : les entrées qui sont toujours mises à 1.
- Poids (W_i) : sont les facteurs multiplicateurs qui affectent l'influence de chaque entrée sur la sortie de neurone.
- Noyau (Somme pondérée + Fonction d'activation) : $F(x)=\sum(X_iW_i)-W_0$.

2.2.3. Fonction d'activation

La fonction d'activation (ou fonction de transfert) sert à convertir le résultat de la somme pondérée des entrées d'un neurone en une valeur de sortie, cette conversion s'effectue par un calcul de l'état du neurone en introduisant une non-linéarité dans le fonctionnement du neurone. Le biais b joue un rôle de seuil, quand le résultat de la somme pondérée dépasse ce seuil, l'argument de la fonction de transfert devient positif ou nul; dans le cas contraire, il est considéré négatif. Finalement si le résultat de la somme pondérée est:

- en dessous du seuil, le neurone est considéré comme non-actif
- aux alentours du seuil, le neurone est considéré en phase de transition.
- au-dessus du seuil, le neurone est considéré comme actif [20].

Différentes fonctions de transfert pouvant être utilisées comme fonction d'activation du neurone. Les trois les plus utilisées sont les fonctions «seuil», «linéaire» et «sigmoïde» [3].

- **Fonction de seuil**

Comme son nom l'indique, la fonction seuil applique un seuil sur son entrée. Plus précisément, si l'entrée est négative (ne passe pas le seuil) la fonction retourne alors la valeur 0 (faux), si l'entrée est positive ou nulle (dépasse le seuil) la fonction retourne 1 (vrai) [3].

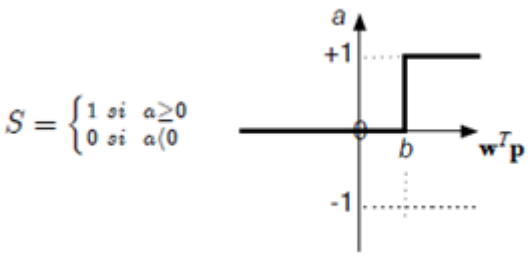


Fig13 : Fonction d'activation seuil [3].

- **Fonction linéaire**

C'est une fonction simple de la forme: f(x) = ax ou f(x) = x. En général, l'entrée passe à la sortie sans une très grande modification ou alors sans aucune modification. On reste donc dans une situation de proportionnalité.

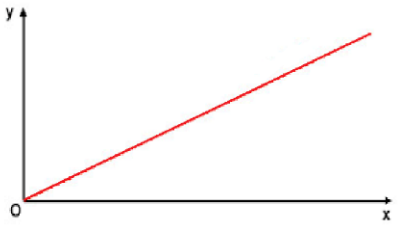


Fig14 : Fonction d'activation linéaire [3].

- **Fonction de sigmoïde**

Elle est la plus utilisée car elle introduit de la non-linéarité, mais c'est aussi une fonction continue, différentiable. Une fonction sigmoïde est définie par :

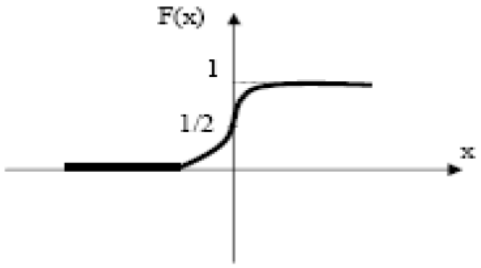


Fig15 : Fonction de sigmoïde.

2.2.4. Types de réseaux de neurones

On distingue deux grands types d'architectures de réseaux de neurones : les réseaux de neurones *non bouclés* et les réseaux de neurones *bouclés.*

2.2.4.1. Réseaux de neurones non bouclés

Un réseau de neurones non bouclé est représenté par un ensemble de neurones "connectés" entre eux, l'information circulant des entrées vers les sorties sans "retour en arrière"; si l'on

représente le réseau comme un graphe dont les nœuds sont les neurones et les arêtes les "connexions" entre ceux-ci, le graphe d'un réseau non bouclé est acyclique [7].

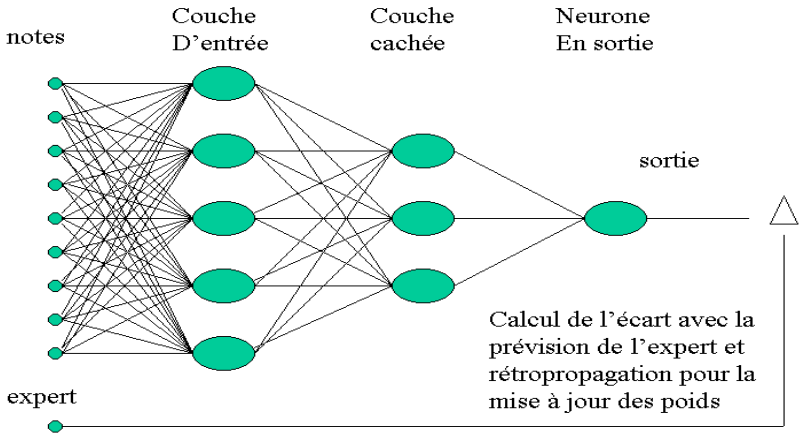


Fig16 : Réseaux de neurones non bouclés [10].

2.2.4.2. Réseaux de neurones bouclés

Contrairement aux réseaux de neurones non bouclés dont le graphe de connexions est acyclique, les réseaux de neurones bouclés peuvent avoir une topologie de connexions quelconque, comprenant notamment des boucles qui ramènent aux entrées la valeur d'une ou plusieurs sorties [7].

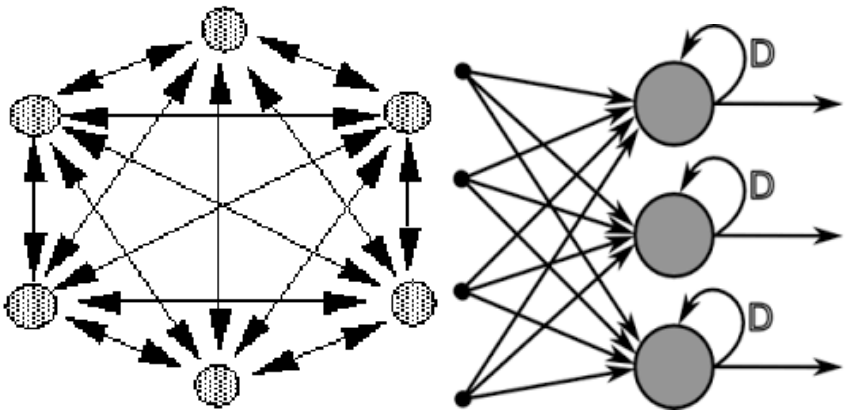


Fig17 : Réseaux de neurones bouclés [8].

2.2.5. Architecture d'un réseau de neurones artificiel

Dans les architectures des réseaux de neurones, on remarque qu'il existe deux types de la structure. La structure d'un réseau de neurone à monocouche (Perceptron monocouche) et la structure d'un réseau de neurone multicouche (Perceptron multicouche).

2.2.5.1. Réseaux monocouche

Un réseau monocouche est un réseau de neurones contenant n neurones en entrée et m neurones en sortie. Les neurones d'entrées soient entièrement connectés aux neurones en sortie par une couche modifiable de poids [10].

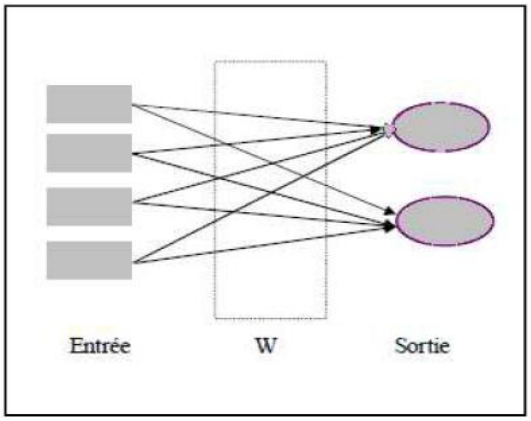


Fig18 : Réseau de neurones monocouche.

2.2.5.2. Réseaux multi-couches

Le Perceptron multicouche est une extension du perceptron monocouche qui dispose d'une ou de plusieurs couches cachées. Les neurones y sont arrangés en couches successives : la première couche qui forme 1e vecteur des données d'entrées est appelée couche d'entrée tandis que la dernière couche qui produit les résultats est appelée couche de sortie. Toutes les autres couches qui se trouvent au milieu sont appelées couches cachées.Les neurones de la couche d'entrée sont connectés uniquement à la couche suivante tandis que les neurones des couches cachées ont la particularité d'être connecté à tous les neurones de la couche précédente et de la couche suivante, par contre il n'y a pas de connexions entre les neurones d'une même couche.

Contrairement au Perceptron monocouche la présence d'une couche cachée dans le Perceptron multicouche facilite la modélisation des relations non linéaires entre les entrées et la sortie.

Le choix du nombre de couches cachées dépend généralement de la complexité du problème à résoudre, en théorie une seule couche cachée peut être suffisante pour résoudre un problème donné mais il se peut que le fait de disposer de plusieurs couches cachées permette de résoudre plus facilement un problème complexe.

2.2.5.2.1. Réseaux multicouche classique

Dans un réseau multicouche classique, il n'y a pas de connexion entre neurones d'une même couche et les connexions ne se font qu'avec les neurones de la couche aval [10].

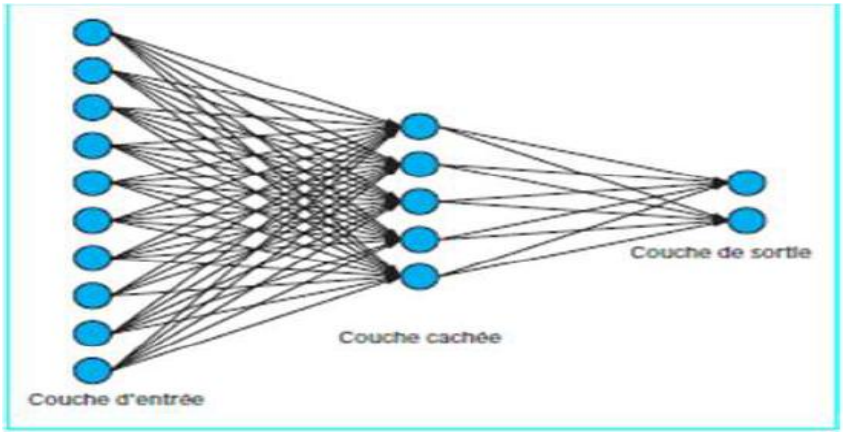


Fig19 : Réseau multicouche classique [10].

2.2.5.2.2. Réseau à connexions locales

C'est aussi un réseau multicouche, mais tous les neurones d'une couche amont ne sont pas connectés à tous les neurones de la couche aval. Nous avons donc dans ce type de réseau de neurones un nombre de connexions moins important que dans le cas du réseau de neurones multicouche classique [10].

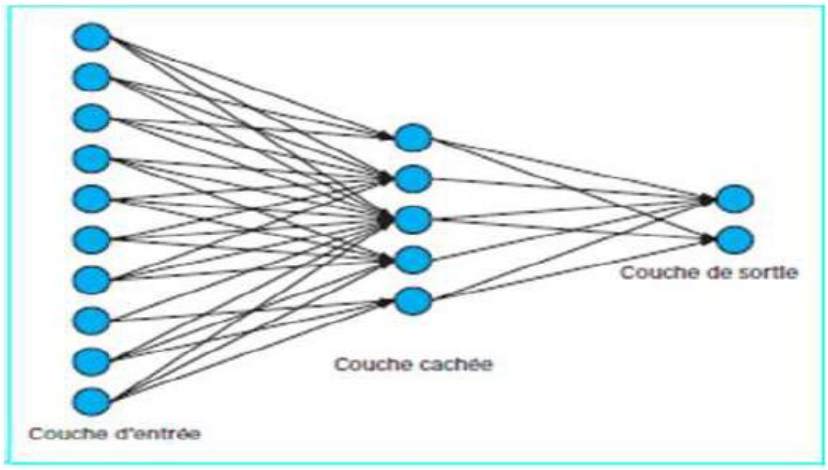


Fig20 : Réseau à connexion locale [10].

2.2.5.2.3. Réseau à connexions récurrentes

Un réseau de ce type signifie qu'une ou plusieurs sorties de neurones d'une couche aval sont connectées aux entrées des neurones de la couche amont ou de la même couche. Ces connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouche. Les réseaux à connexions récurrentes sont des réseaux plus puissants car ils sont séquentiels plutôt que combinatoires comme l'étaient ceux décrits précédemment [10].

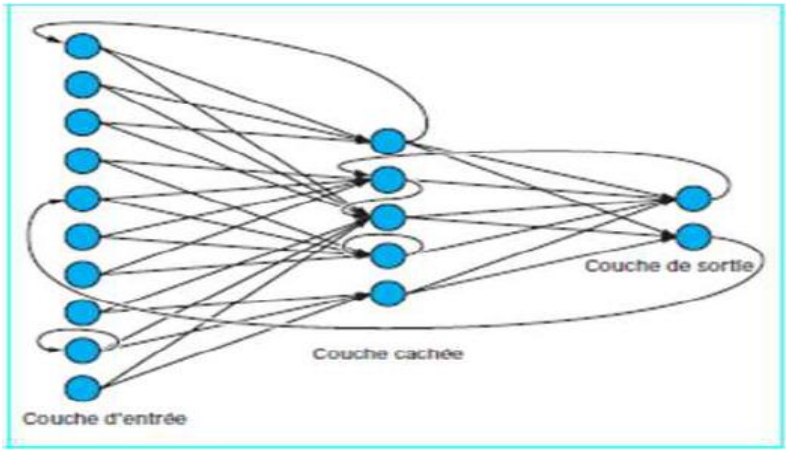


Fig21 : Réseau à connexions récurrentes [10]

2.2.6. Apprentissage des réseaux de neurones

La tâche principale des réseaux de neurones artificiels est l'apprentissage pour la classification, qui est réalisée par un processus itératif d'adaptation des poids W_i pour arriverà la meilleure fonction permettant d'avoir $f(x_i) = y_i$, $\forall$ i = 1..N. Les valeurs des Wi sont initialisées aléatoirement, et corrigées selon les erreurs entre les yi obtenus et attendus.

Dans un réseau de neurones multicouches, la correction se fait dans le sens inverse du sens de propagation des données ce qui est appelé la rétro-propagation (en anglais : back-propagation). À chaque présentation d'un exemple d'apprentissage au réseau, on passe par deux étapes :

- Dans l'étape de propagation, les valeurs du vecteur d'entrée (l'exemple) sont reçues dans la couche d'entrée et propagées d'une couche à l'autre jusqu'à la sortie où un vecteur de sortie (les y_i) est obtenu.
- Dans la phase de rétro-propagation, les W_i sont ajustés de la dernière couche jusqu'à la première de manière à rapprocher les yi obtenus de ceux attendus.

Ces deux étapes sont répétées avec chaque exemple d'apprentissage pour obtenir à la fin un réseau de neurones artificiel entrainé. L'utilisation d'un RNA entrainé, se fait par l'injection des valeurs du vecteur de l'exemple à classifier, dans l'entrée et recevoir sa classe à la sortie par propagation [21].

2.2.7. Les Limites de Réseaux de neurones

- Boite noire : très difficile (impossible) d'analyser et comprendre le fonctionnement en face d'un problème donné. Les réseaux de neurones ne fournissent pas les explications concernant leurs résultats
- Difficulté de choisir la structure (type, nombre de nœuds, organisation, connexions,...etc) la mieux adaptée au problème [21].

- Non optimalité de l'architecture : Il n'existe pas encore de moyens permettant de définir l'architecture optimale du réseau de neurones. En effet, le réseau qui apparaît optimal d'une façon globale ne délivre pas toujours les résultats les plus pertinents.

3. Apprentissage profond

Une des grandes différences entre l'apprentissage profond et les algorithmes d'apprentissage automatique traditionnelles c'est qu'il s'adapte bien, plus la quantité de données fournie est grande plus les performances d'un algorithme d'apprentissage profond sont meilleurs. Contrairement à plusieurs algorithmes de l'apprentissage automatique classiques qui possèdent une borne supérieure à la quantité de données qu'ils peuvent recevoir des fois appelée "plateau de performance", les modèles d'apprentissage profond n'ont pas de telles limitations (théoriquement) et ils sont même allés jusqu'à dépasser la performance humaine dans des domaines comme le traitement d''image.

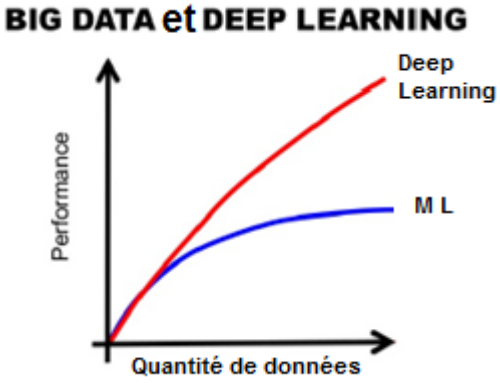


Fig22: La différence de performance entre l'apprentissage profond et la plupart des algorithmes de machine learning en fonction de la quantité de données [5].

Autre différence entre les algorithmes d'apprentissage automatique traditionnels et les algorithmes d'apprentissage profond c'est l'étape de l'extraction de caractéristiques. Dans les algorithmes d'apprentissage automatique traditionnels l'extraction de caractéristiques est faite manuellement, c'est une étape difficile et coûteuse en temps et requiert un spécialiste en la matière alors qu'en apprentissage profond cette étape est exécutée automatiquement par l'algorithme [5].

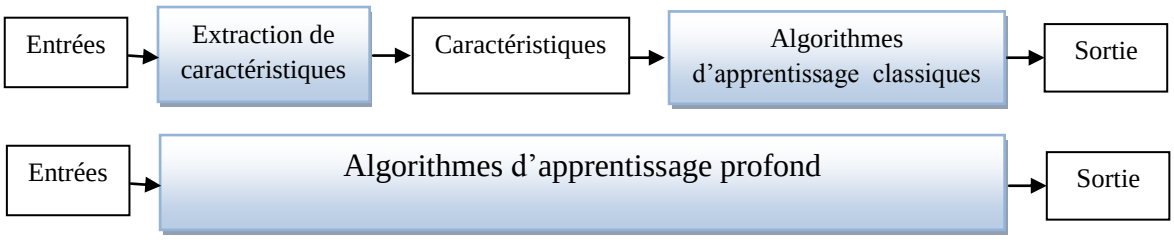


Fig23 : Comparaison entre l'apprentissage automatique et l'apprentissage profond

4. Architectures d'apprentissage profond

Il existe différentes architectures possibles pour construire un réseau de neurones profond :

4.1. Réseau de Neurone à Convolution

Réseaux de neurones à convolution (Convolution Neural Network (CNN)) sont considérés comme un type spécialisé de réseau de neurones pour le traitement de données ayant une topologie semblable à une grille, qui peuvent être considérées comme une grille 1D (Vecteur) et grille 2D de pixels (Matrice). Les réseaux à convolution ont connu un succès considérable dans les applications pratiques nous citons par exemple reconnaissance de l'image et de la vidéo, les systèmes de recommandations et le traitement du langage naturel… [5]. Le nom « réseau de neurones à convolution» indique que le réseau emploie une opération mathématique appelée *convolution*. La convolution est une opération linéaire spéciale [12].

4.1.1. Principe d'architecture d'un CNN

Les réseaux de neurones à convolution sont à ce jour les modèles les plus performants pour classer des images**.** Ils comportent deux parties bien distinctes.

La première partie d'un CNN est la partie convolutive à proprement parler. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers d'une succession de filtres ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions (Fig21). Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. En fin, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN.

Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée de couches entièrement connectées (perceptron multicouche). Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image[22].

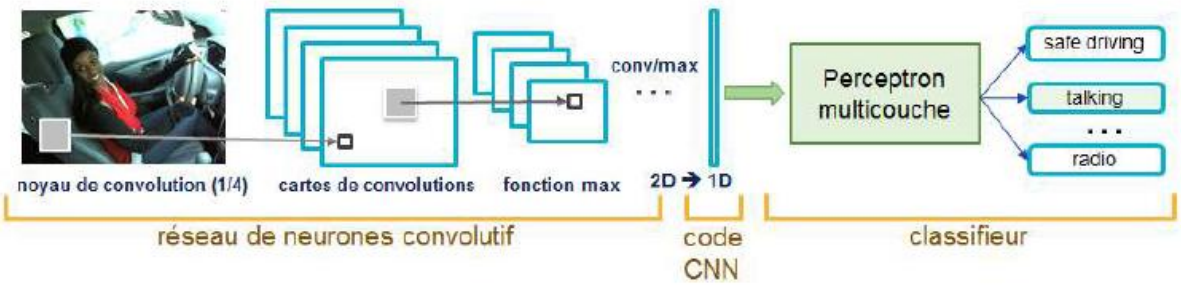


Fig24 : Les réseaux de neurones convolutifs [22].

4.1.2. Les couches de CNN

4.1.2.1. La couche de convolution (CONV)

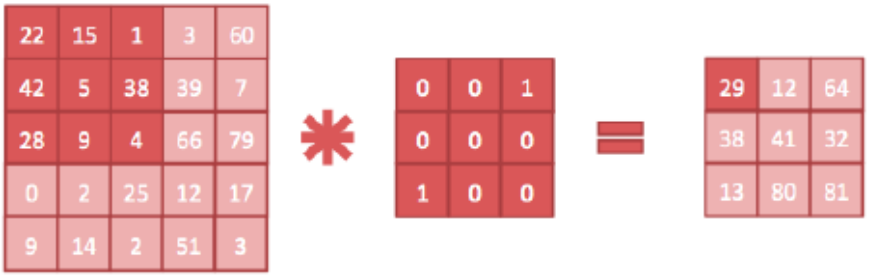


Fig25 : Une illustration simple de l'opération de convolution à deux dimensions [4].

Comme le montre la figure 25, la matrice la plus à gauche est la matrice d'entrée. Celui du milieu est généralement appelé matrice du noyau ‘kernel‘. La convolution est appliquée à ces matrices et le résultat est présenté comme la matrice la plus à droite. Le processus de convolution est un produit élément par élément suivi d'une somme, comme illustré dans l'exemple.

L’opération de convolution est généralement appelée noyau ‘kernel‘. Par différents choix de noyaux ‘kernel‘, différentes opérations sur les images pourraient être réalisées. Les opérations incluent généralement identité, détection des contours, flou, netteté, etc. En introduisant des matrices aléatoires comme convolution opérateur, certaines propriétés intéressantes pourraient être découvertes [4].

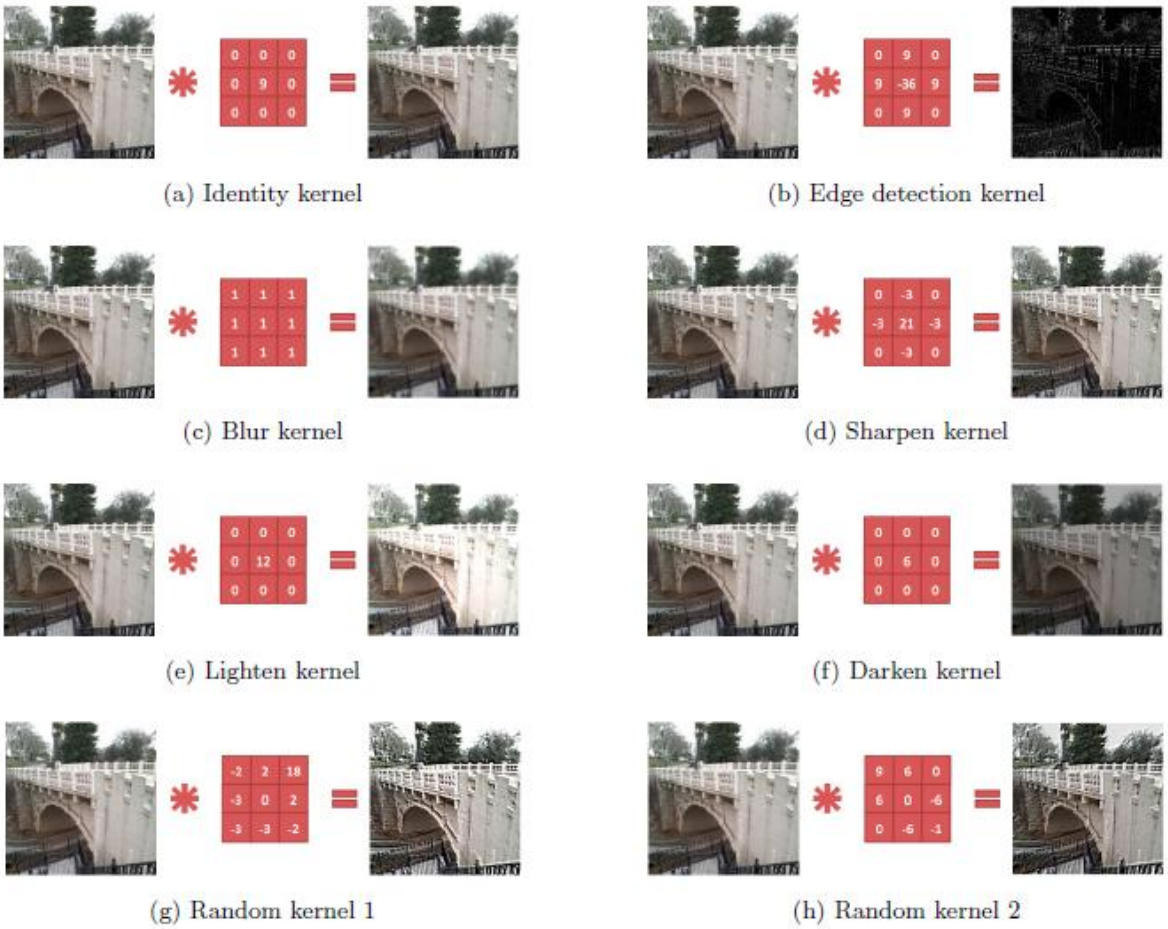


Fig26 : Exemples du kernels appliqués sur la même image [4].

4.1.2.2. Couche de pooling

Un autre outil très puissant utilisé par les CNNs s'appelle le Pooling. Le Pooling est une méthode permettant de prendre une large image et d'en réduire la taille tout en préservant les informations les plus importantes qu'elle contient [12].Il y a plusieurs façons de faire cette mise en commun, comme prendre le **maximum** ou la **moyenne**, ou une **combinaison linéaire** apprise des neurones dans le bloc. Par exemple, la figure 27 montre max pooling sur une fenêtre 2×2.

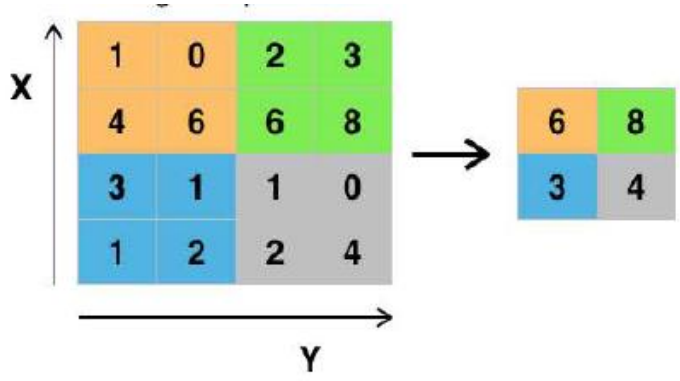


Fig27 : Le pooling [12].

Au final, une couche de pooling est simplement un traitement de pooling sur une image ou une collection d'images. L'output aura le même nombre d'images mais chaque image aura un nombre inférieur de pixels. Cela permettra ainsi de diminuer la charge de calculs.

L'image ci-dessous montre un exemple de Pooling. La tuile a ici des dimensions de 3 par 3. L'image de 9 par 9 pixels de départ est réduite en une image de 7 par 7 pixels [13].

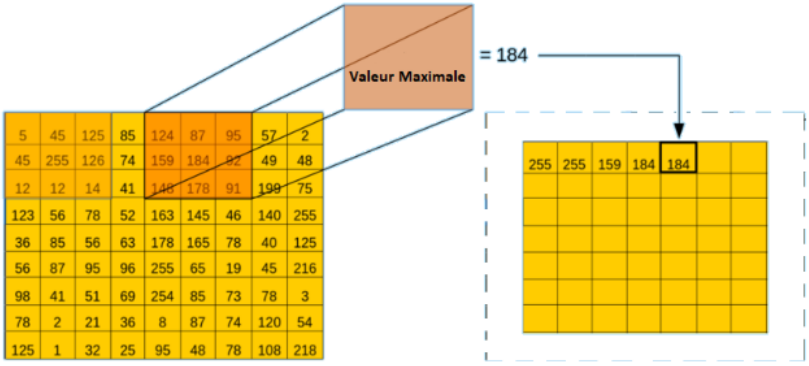


Fig28 : Pooling [13].

4.1.2.3. Couche entièrement connectée

Après plusieurs couches de convolution et de max-pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées. Les neurones dans une couche entièrement connectée ont des connexions vers toutes les sorties de la couche précédente [22, 13].

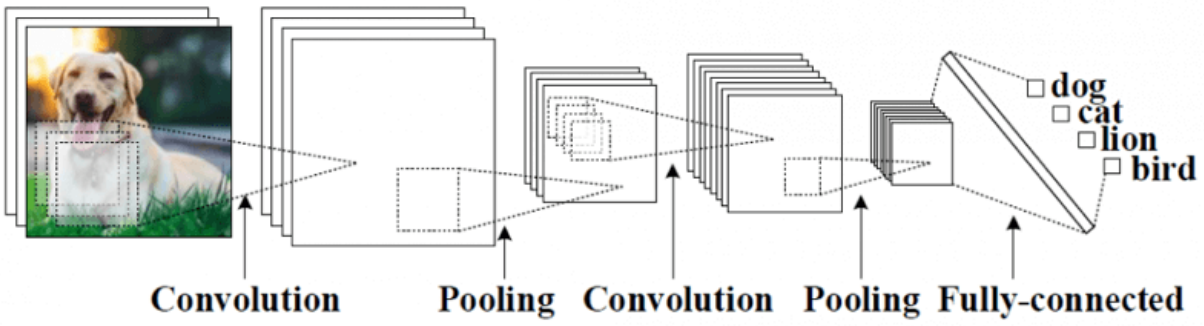


Fig29 : Un réseau de neurones à convolution qui reçoit une image 2D comme entrée.

4.2. RNN (Recurrent Neural Networks)

Un réseau de neurones récurrents est un réseau de neurones artificiels présentant des connexions récurrentes. Un réseau de neurones récurrents est constitué d'unités (neurones) interconnectés interagissant non-linéairement et pour lequel il existe au moins un cycle dans la structure.

Les architectures de réseaux neuronaux récurrents peuvent prendre de nombreuses formes différentes. Un type commun consiste en un perceptron multicouche standard (MLP) plus des boucles ajoutées. Pour les architectures simples et les fonctions d'activation déterministes,

l'apprentissage peut être réalisé à l'aide de procédures de descente de gradient similaires à celles conduisant à l'algorithme de rétro-propagation pour les réseaux à rétroaction.

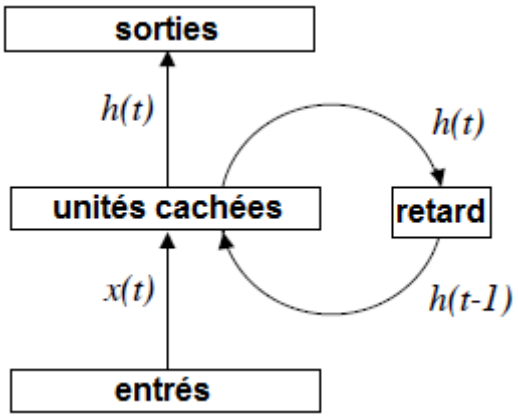


Fig30 : Un réseau de neurones récurrents [23].

Notez que le temps t doit être discrétisé, les activations étant mises à jour à chaque pas de temps. Une unité de délai doit être introduite pour conserver les activations jusqu'à leur traitement au prochain pas de temps [23].

4.3. LSTM (Long Short Term Memory networks)

Les réseaux de mémoire à long terme - généralement appelés simplement "LSTM" - sont un type particulier de RNN, capable d'apprendre des dépendances à long terme. Ils ont été introduits par Hochreiter&Schmidhuber (1997).

LSTM peut non seulement traiter des points de données uniques (tels que des images), mais également des séquences complètes de données (telles que la parole ou la vidéo). Par exemple, LSTM est applicable à des tâches telles que la reconnaissance de l'écriture manuscrite ou la reconnaissance vocale.

Une unité LSTM commune est composée d'une cellule, d'une porte d'entrée, d'une porte de sortie et d'une porte d'oubli. La cellule se souvient des valeurs sur des intervalles de temps arbitraires et les trois portes régulent le flux d'informations entrant et sortant de la cellule.

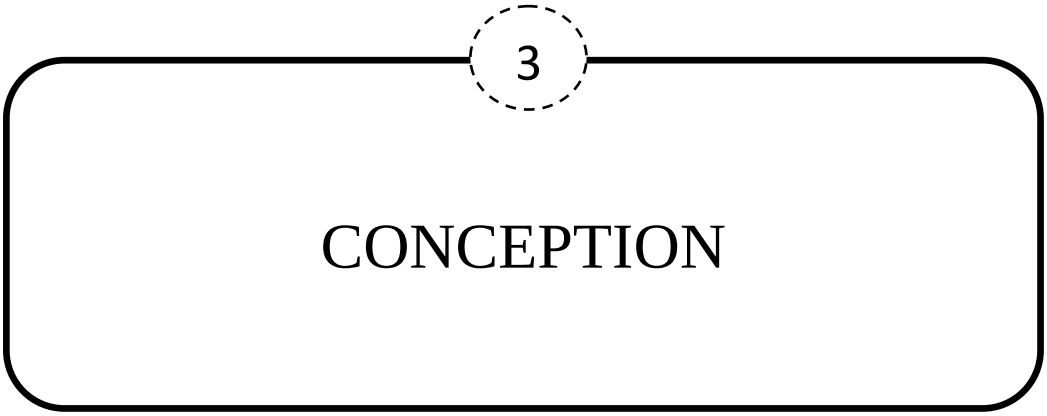
IV. Conclusion

Dans ce chapitre, On a présenté les notions importantes qui sont en relation avec l'apprentissage profond. Les systèmes d'apprentissage profond commencent à surpasser non seulement les méthodes classiques, mais également les capacités humaines dans diverses tâches telles que la classification des images ou la reconnaissance des visages...

Avec des quantités énormes de puissance de calcul, les machines peuvent maintenant reconnaître des objets et traduire la parole en temps réel. L'intelligence artificielle devient finalement intelligente.

Il est prédit que de nombreuses applications de l'apprentissage profond affecteront votre vie dans un avenir proche. En fait, ils ont déjà un impact. Au cours des cinq à dix prochaines années, les outils de développement, les bibliothèques et les langages d'apprentissage approfondi deviendront des composants standards de chaque boîte à outils de développement logiciel.

CHAPITRE



CONCEPTION

I. Introduction

Un document Web comprend généralement de nombreux types d'informations. On trouve, le contenu principal qui transmet les informations primaires, et des contenus bruyants tels que des publicités, des en-têtes, des pieds de page, des décorations, des informations de copyright, des menus de navigation, etc. La présence de contenus bruyants peut affecter les performances d'applications telles que les moteurs de recherche, crawlers et Web miners…etc. Par conséquent, extraire le contenu principal du document Web et supprimer le contenu bruyant est un processus important.

Dans ce chapitre, On va formuler la tâche d'extraction de contenu en tant que problème de classification pour le traiter ensuite par l'apprentissage profond.

Notre Objectif est de réaliser un système capable d'extraire le contenu principal à partir d'une masse des pages web en utilisant l'apprentissage profond (Réseaux de Neurones à convolution profond). On va prendre le problème d'extraction du contenu comme un problème de classification (on va classer le contenu de la page web vers contenu principal ou contenu bruyant).

II. Conception globale du système

Globalement pour construire un modèle de classification de contenu des pages web (contenu principal ou contenu bruyant) notre système suit les étapes suivantes :

- La première étape est une **préparation** des données (prétraitement + représentation vectorielle des données).
- La deuxième étape est **l'apprentissage du Réseaux de neurone à convolution (CNN)**, en utilisant comme entrée le résultat de la première étape.

On peut schématiser la conception comme suit :

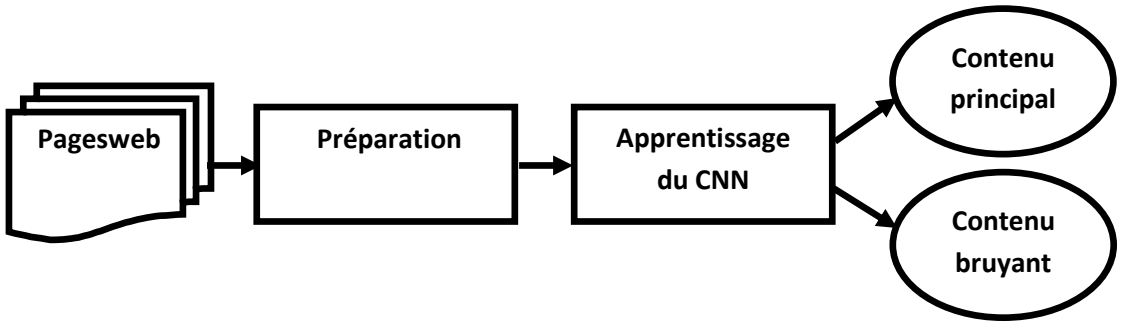


Fig31 : Conception globale du système de classification.

III. Conception détaillée du système

Dans la conception détaillée, On va expliquer profondément les étapes de la conception globale.

1. La phase de préparation de données

La phase de préparation de données comprend deux étapes :

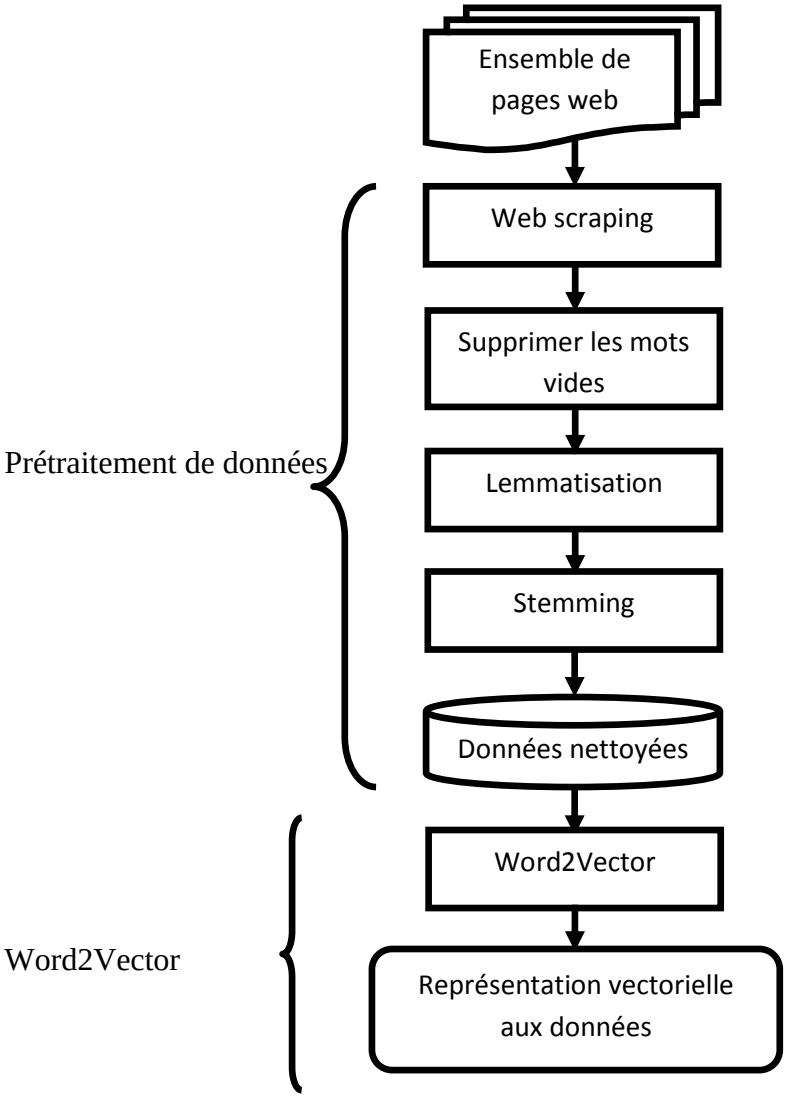


Fig32 : Le processus de prétraitement.

1.1. Prétraitement de données

L'entrée du système est un ensemble des pages web, dont chacune composé d'un contenu principal et un contenu bruyant. Le prétraitement est très important dans le processus de classification car la construction du modèle est basée sur les données préparées. En effet les pages web qui ne sont pas préparées correctement peuvent donner un modèle non performant.

1.1.1. Web scraping

Le web scraping (parfois appelé harvesting) est une technique d'extraction du contenu de pages Web, via un script ou un programme, dans le but de le transformer pour permettre son utilisation dans un autre contexte.

Pour chaque page web, On va créer un ***fichier csv*** contenant les deux classes avec le web scraping. La dernière étape de la création du la base d'apprentissage consiste à concaténer les fichiers CSV individuels résultant de le web scraping de chaque page Web, comme le montre la figure 33.

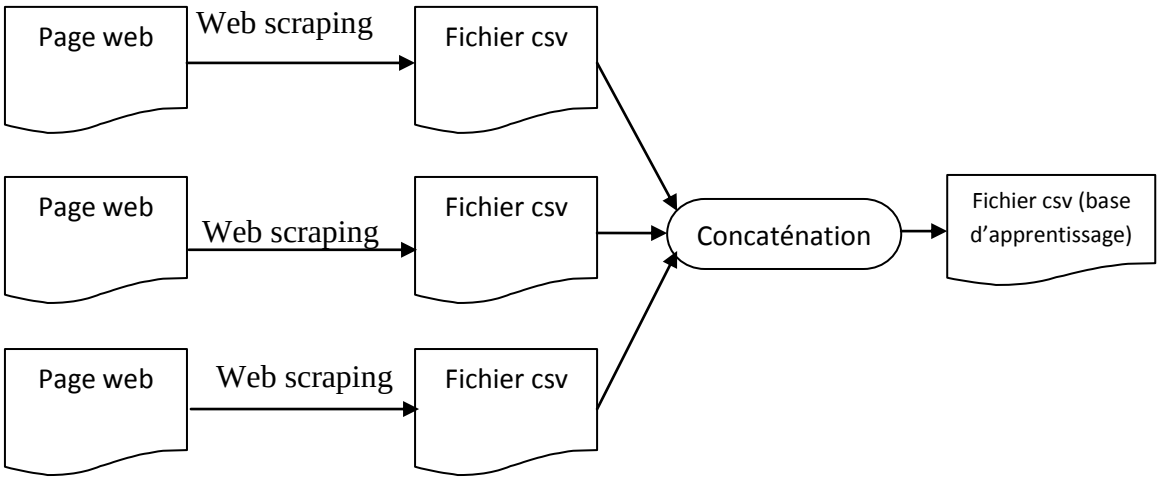


Fig33 : Le processus de création de la base d'apprentissage (et le même processus sera utilisé pour créer la base de test).

1.1.2. Suppression des mots vides

Un **mot vide** (en anglais : **stop word**) est un mot qui est tellement commun qu'il est inutile de l'indexer ou de l'utiliser dans une recherche. Elimination de mots vides consiste à supprimer tous les mots standards dans le contenu du page web extraite par le web scraping, ce sont des mots très communs et utilisés dans pratiquement tous les textes. Leur présence peut dégrader la performance de l'algorithme de classification en termes de coût et en termes de précision de la classification. Notre base est en anglais donc On peut prendre comme exemple pour l'anglais la liste suivante: I, me, my, myself, we, our, ours, ourselves, you, your, yours, yourself, yourselves, he, him, his, himself,…etc.

1.1.3. Lemmatisation

La lemmatisation est par définition une action consistant à l'analyse lexicale d'un texte avec pour but de regrouper les mots d'une même famille. On parle ici de donner la forme canonique d'un mot ou d'un ensemble de mots : Chacun de ces mots d'un contenu donné se trouve réduit en une entité appelée en lexicologie lemme ou encore « forme canonique d'un mot ». Les lemmes d'une langue utilisent plusieurs formes en fonction :

- du genre (masculin ou féminin),

- de leur nombre (un ou plusieurs),
- leur personne (moi, toi, eux…),
- de leur mode (indicatif, impératif…)

Il existe généralement plusieurs formes pour un même lemme.

- Quelques exemples sont plus évocateurs pour présenter la lemmatisation :

L'adjectif grand existe sous quatre formes : grand, grande, grands et grandes. La forme canonique de tous ces mots est **grand**.

1.1.4. Stemming

Stemming est un procédé de transformation des mots en leur radical ou racine. La racine d'un mot correspond à la partie du mot restante une fois que l'on a supprimé son préfixe et suffixe, à savoir son radical. Un programme informatique de stemming est appelé un racinisateur. Les algorithmes les plus connus ont été développés par **Julie Beth Lovins** en **1968** et **Martin Porter 1980**.

Par exemple, en anglais, le stemming de «fishing», «fished», «fish» et «fisher» donne «fish».

1.2. Word2Vector

Le modèle ***word2vec*** et ses applications ont récemment attiré l'attention de la communauté de l'apprentissage automatique. ***Word2vec*** est une méthode fondée sur des réseaux de neurones artificiels. Cette méthode propose deux architectures : l'architecture "sac de mots continus " (CBOW) et l'architecture "Skip-gram". Ces deux architectures se présentent sous la forme de réseaux de neurone simple. Ils sont constitués de trois couches : une couche d'entrée, une couche cachée et une couche de sortie. La couche d'entrée contient soit un "sac-de-mots" (CBOW), soit un mot seul (Skip-gram). La couche cachée correspond à la projection des mots d'entrée dans la matrice des poids. Cettematrice est partagée par tous les mots (matrice globale). Enfin, la couche desortie est composée de neurones "softmax". Pour des raisons de complexitéalgorithmique due à la couche "softmax". Le couplage de ces fonctions avec lasimplicité de ces réseaux leur permettent d'être entraînés sur des très grandesquantités de textes, et ainsi d'obtenir des modélisations de meilleure qualitéque les modèles plus complexes à base de récurrence ou de convolution [36, 37].

1.2.1. Approche par sac-de-mots continus (CBOW)

L'architecture CBOW est un réseau de neurones devant prédire un mot à partir de son contexte. La couche d'entrée représente la présence ou l'absence des mots dans le contexte de manière binaire (i.e. 1 pour la présence, 0 pour l'absence).

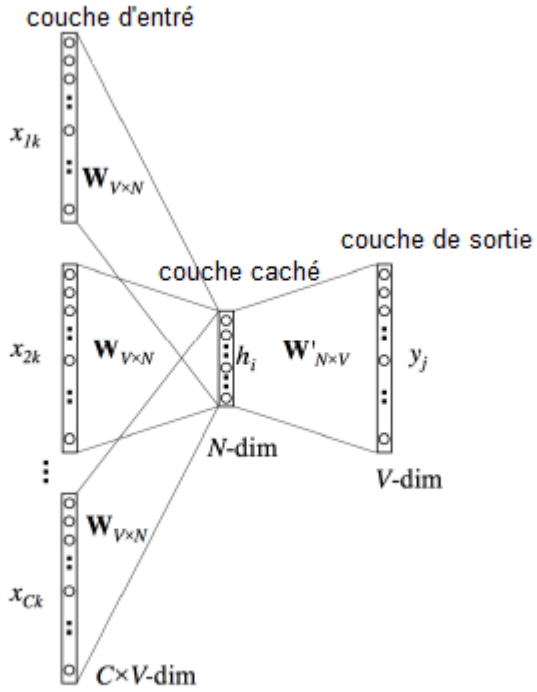


Fig34 : Modèle général de CBOW [36].

Chaque mot dans le contexte est projeté dans la matrice des poids du modèle. La somme (ou la moyenne) de ces représentations passe ensuite par la couche de sortie. Enfin, le modèle compare sa sortie avec le mot seul et corrige sa représentation par rétro-propagation du gradient [37].

1.2.2. Approche par Skip-Gram

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

Comparativement au CBOW, cette architecture permet une meilleure modélisation des mots peu fréquents et permet de mieux capturer les relations sémantiques [36].

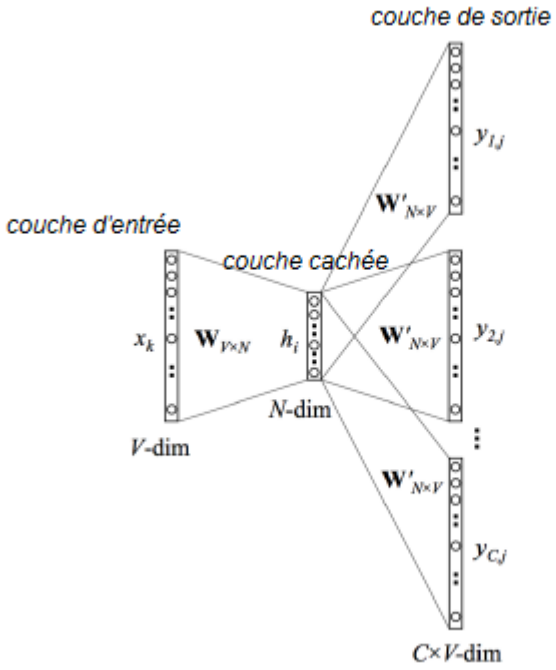


Fig35 : Modèle général de Skip-Gram [36].

2. Apprentissage en profondeur (CNN)

L'étape suivante représente l'apprentissage d'un réseau de neurone profond (CNN),un CNN est simplement un empilement de plusieurs couches de *convolution*, *pooling* et *fully-connected*. L'entrée est un espace vectoriel du contenu de la page web et la sortie est la classe de contenu. Dans le problème de classification, ce vecteur contient les probabilités d'appartenance aux classes. Un CNN a été implémenté parce qu'il a été prouvé qu'il est possible d'obtenir des résultats impressionnants en utilisant des convolutions sur la couche d'entrée pour calculer la sortie.

Créer un nouveau réseau de neurones convolutif est coûteux en termes d'expertise, de matériel et de quantité de données annotées nécessaires.Il s'agit d'abord de fixer l'architecture du réseau, c'est-à-dire le nombre de couches, leurs tailles et les opérations matricielles qui les connectent. L'apprentissage consiste alors à optimiser les coefficients du réseau pour minimiser l'erreur de classification en sortie. Cet apprentissage peut prendre plusieurs semaines pour les meilleurs CNN, avec de nombreux GPU travaillant sur des pages web annotées.

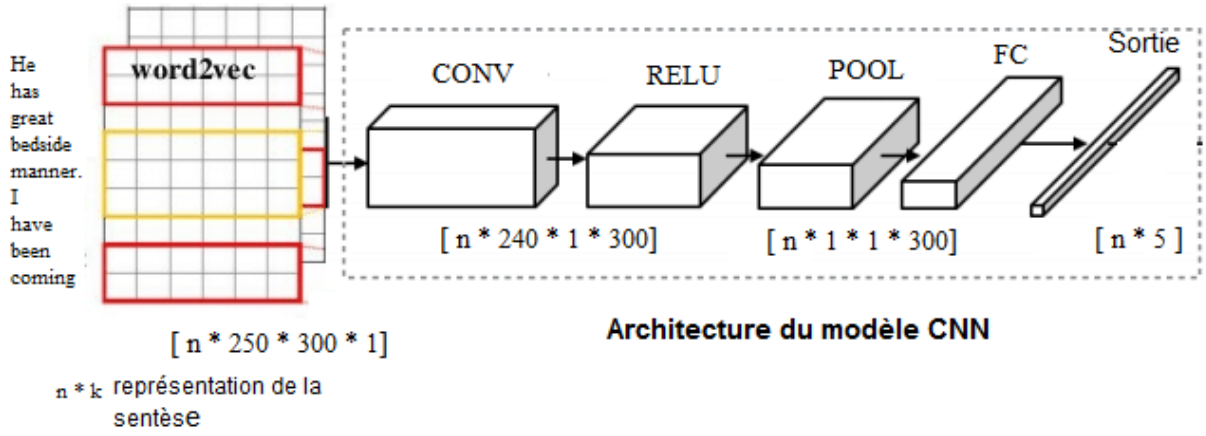


Fig36 : Modèle général de classification.

IV. Conclusion

Dans ce chapitre, On a appliqué la méthode du réseau de neurone à convolution pour la classification de contenu des pages web, On a défini le contenu principal et le contenu bruit de la page web et on a propose une conception générale et une conception détaillée du système. Dans le prochain chapitre, On va présenter l'implémentation de notre système et les résultats obtenus.

CHAPITRE

4

IMPLEMENTATION

I. Introduction

On a présenté dans le chapitre précédent la conception globale et la conception détaillée utilisée pour l'extraction de contenu des pages web en utilisant l'apprentissage profond. Et dans ce chapitre, On va expliquer les étapes qu'on a suivies pour implémenter notre system d'extraction.

Mais en premier temps, On va présenter le langage de programmation et l'environnement du développement avec les différents outils utilisés. Ensuite, On va exposer l'expérimentation qu'on a appliquée sur la méthode CNN et on va aussi expliquer les résultats obtenus.

II. Langage et l'environnement de la programmation

Python : Le langage de programmation utilisé dans ce projet est **Python**. Ce dernier est un langage de programmation inventé par **Guido van Rossum**. La première version de python est sortie en **1991**. Il est l'un des langages de programmation les plus intéressants du moment. Il est facile à apprendre et il est souvent utilisé en exemple lors de l'apprentissage de la programmation.

Python est un langage de programmation dynamique de haut niveau, interprété et populaire, qui met l'accent sur la lisibilité du code. La syntaxe dans Python aide les programmeurs à coder en moins d'étapes que Java ou C ++.

Anaconda : est un *bundle* Python actuellement très en vogue dans le monde scientifique. Développé par la société Anaconda Inc., non "libre" mais gratuit dans sa version de base, il a l'avantage d'être multiplateforme (Windows, mac OS et GNU/Linux) et d'intégrer une grande quantité d'outils et packages Python, notamment : IPython, Spyder, Jupyter, NumPy, SciPy, MatplotLib, Pandas, Sympy, PIP... [38].

Puisque on a utilisé une grande base de données pour l'apprentissage du CNN, donc on a besoin d'un processeur graphique très puissant et pour cela, on a utilisé l'outil Google Colaboratory.

Google Colaboratory : Colaboratory est un projet de recherche Google créé dans le but de diffuser la recherche en apprentissage automatique. Il s'agit d'un environnement pour ordinateur portable Jupyter qui ne nécessite aucune installation et fonctionne entièrement dans le cloud. Cela signifie que tant que vous avez un compte Google, vous pouvez librement former vos modèles sur un processeur graphique K80.

Lorsqu'on connecte au run time de Colaboratory, onobtiendra notre propre machine virtuelle avec un processeur graphique K80 et un environnement de d'édition Jupyter [39].

Pour implémenter notre code en python, On a besoin d'installer des packages, qui sont :

Tensorflow : c'est un frame work de Google pour faire du Deep Learning. Il est un outil open source d'apprentissage automatique développé par Google. Le code source a été ouvert le **9novembre2015** par Google et publié sous licence Apache. **TensorFlow** est l'un des outils les plus utilisés dans le domaine de l'apprentissage automatique. On va utiliser Keras, qui est un frame work s'appuyant sur Tensorflow (il simplifie les commandes et ajoute certaines fonctionnalités).

Keras : est une bibliothèque open source écrite en python qui peut fonctionner sur Theano ou TensorFlow, et permet d'interagir avec les algorithmes de réseaux de neurones profonds et de machine learning. Elle a été initialement écrite par **François Chollet.** Il a été développé pour rendre la mise en œuvre des modèles d'apprentissage en profondeur aussi rapide et facile que possible.

Gensim : est une bibliothèque Python pour la modélisation de sujets, l'indexation de documents et la recherche de similarité avec de grands corpus, inclut les implémentations de tf-idf, projections aléatoires, et algorithmesword2vec,...etc.

NumPy : est une extension du langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux. Plus précisément, cette bibliothèque logicielle libre et open source fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices et polynômes.

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

III. La création de la base

Dans ce travail, On va utiliser la base « **L3S-GN1 »**. Elle comprend 621 pages Web. Toutes les pages Web du L3S-GN1 sont en anglais et un examen des pages Web a révélé qu'elles appartenaient toutes à la catégorie articles.

La base **L3S-GN1** comprend deux répertoires : un répertoire contient les pages web ***origine***, et un répertoire contient les mêmes pages web mais elles sont ***annotées*** (l'annotation des pages web veut dire identifier le contenu principal comme montrer la figure 38).

Fig37 : Une page web Origine.

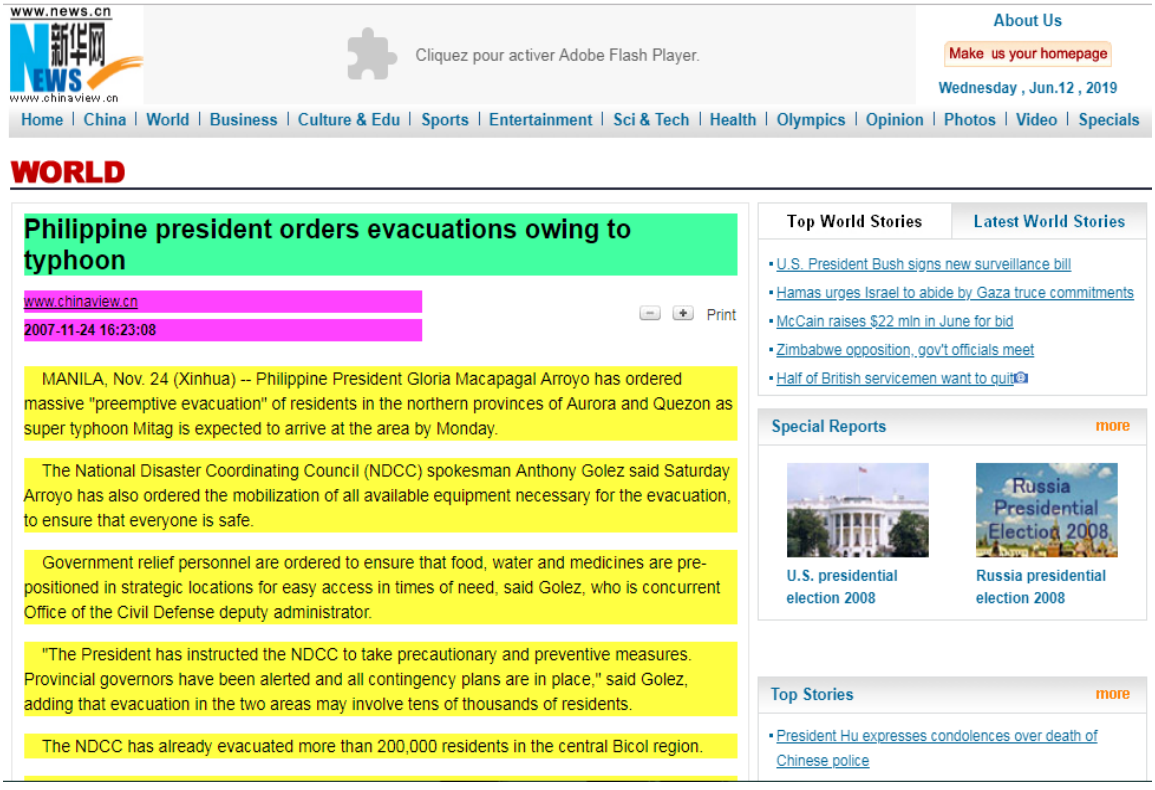


Fig38 : Une page web Annoté.

Les éléments HTML en couleur représentent le contenu principal de la page, et les autres éléments sont le contenu bruyant.

Pour créer la base d'apprentissage et de test on va prendre la base **L3S-GN1 annoté**, puis on va appliquer le ***Web scraping*** pour séparer le contenu principal et le contenu bruyant de la page web (c'est-à-dire on va faire une classification de notre base en utilisant le web scraping pour créer les deux classes à savoir ; classe principale pour le contenu principal et classe bruyante pour le contenu bruyant.

1. Web scraping

Pour faire l'extraction du contenu de la page web, On va utiliser les bibliothèques de **python« Beautiful Soup » et « request ».**

Il faut d'abord télécharger la page web avec la bibliothèque de python « **request** ».Cette bibliothèque va faire une requête **url lib** au serveur Web, qu'il va télécharger le contenu HTML d'une page Web donnée.

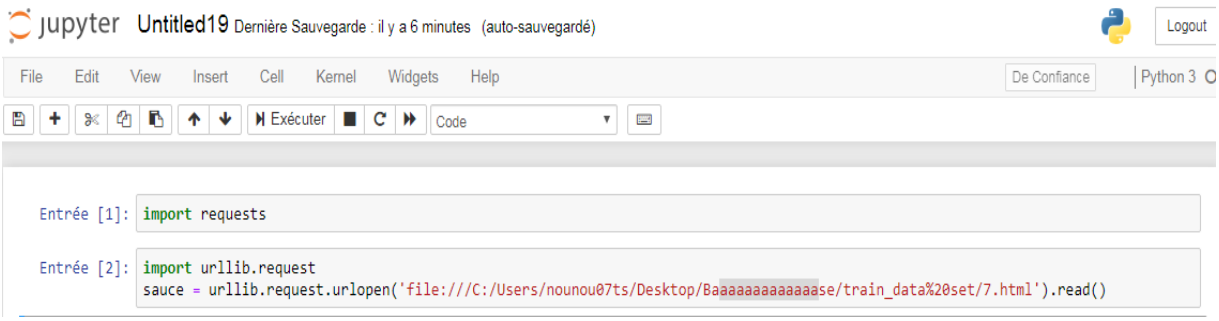


Fig39 : Télécharger le contenu HTML d'une page Web.

Pour afficher le contenu HTML de la page, On a utilisé la bibliothèque « **Beautiful Soup** ».

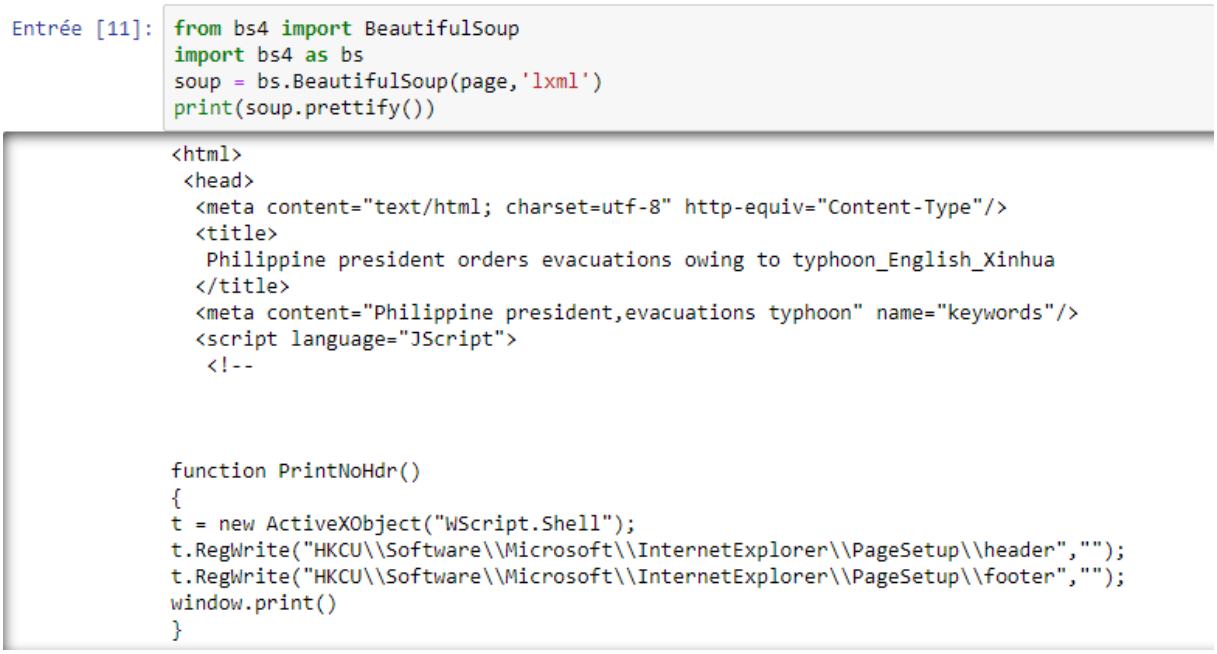


Fig40 : Le contenu HTML de la page web.

Dans ce projet, On va extraire le **titre** et les **paragraphes** de la page web comme contenu principal de la page web et le reste comme contenu bruyant.

- Extraction des titres (la balise <titre>) et des paragraphes (la balise <P>):

```
Entrée [15]: titl=soup.title.text
             print(titl)
             for paragraph in soup.find_all('p'):
                 par=paragraph.text
                 print(par)
```

```
Philippine president orders evacuations owing to typhoon_English_Xinhua

    MANILA, Nov. 24 (Xinhua) -- Philippine President
Gloria Macapagal Arroyo has ordered massive "preemptive evacuation" of residents
in the northern provinces of Aurora and Quezon as super typhoon Mitag is
expected to arrive at the area by Monday.

    The National Disaster Coordinating Council (NDCC)
spokesman Anthony Golez said Saturday Arroyo has also ordered the mobilization
of all available equipment necessary for the evacuation, to ensure that everyone
is safe.

    Government relief personnel are ordered to ensure
that food, water and medicines are pre-positioned in strategic locations for
easy access in times of need, said Golez, who is concurrent Office of the Civil
Defense deputy administrator.

    "The President has instructed the NDCC to take
precautionary and preventive measures. Provincial governors have been alerted
and all contingency plans are in place," said Golez, adding that evacuation in
```

Fig41 : L'extraction de contenu principal.

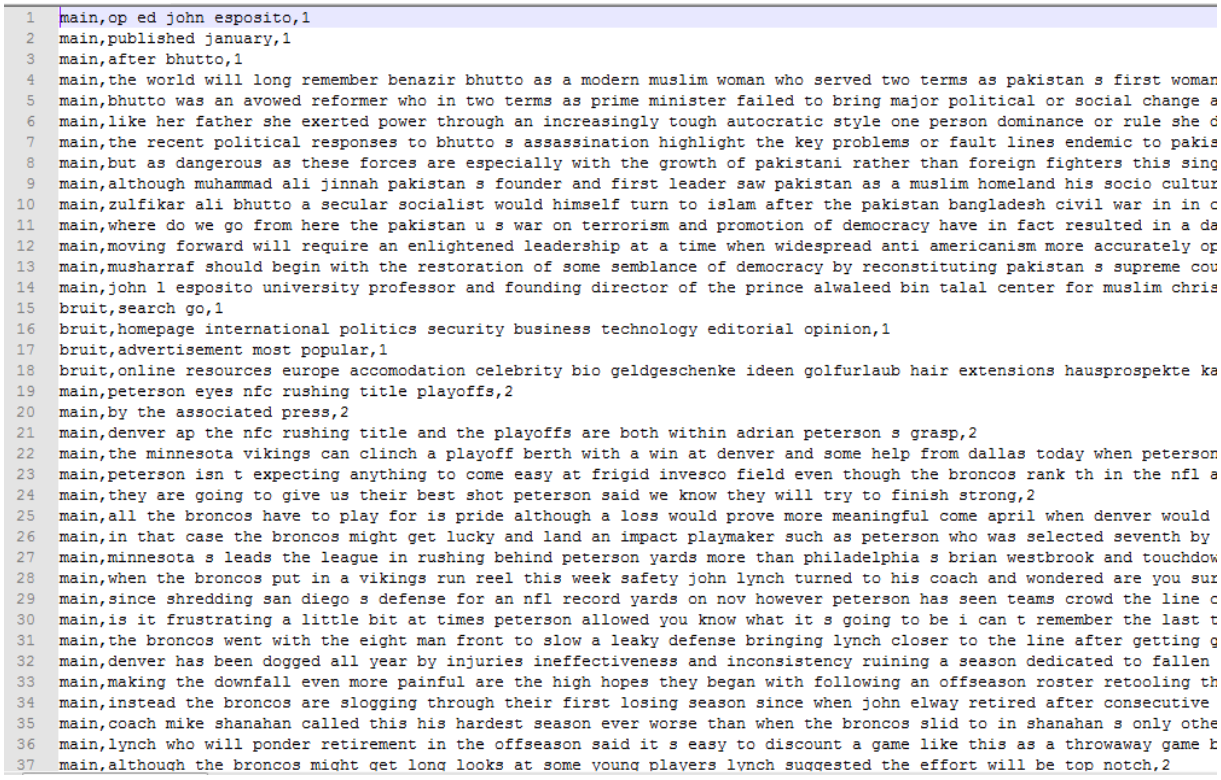
Puis on sauvegarde le contenu extrait dans un **fichier (csv)** et le note comme une classe **'main'** comme il est montré dans la figure 42.

```
149  main,Philippine president orders evacuations owing to typhoon,7
150  main,MANILA Nov Xinhua Philippine President Gloria Macapagal Arroyo has or
151  main,The National Disaster Coordinating Council NDCC spokesman Anthony Gol
152  main,Government relief personnel are ordered to ensure that food water and
153  main,The President has instructed the NDCC to take precautionary and preve
154  main,The NDCC has already evacuated more than residents in the central Bic
155  main,The government Saturday also declared a unilateral truce with the lef
156  main,Armed Forces public information office chief Bartolome Bacarro said t
157  main,The SOMO was declared to support the governments disaster preparednes
158  main,Super typhoon Mitag locally called Mina has changed direction and is
159  main,PAGASA warned earlier that Mitag could arrive in central Luzon and Me
```

Fig42 : Classe main extraite à partir de la page numéro 7.

On fait la même chose pour extraire le contenu bruit de la page web, et le sauvegarder dans le même fichier **csv**. On obtient dans ce cas un fichier contient le contenu principal et le contenu bruit d'une page web.

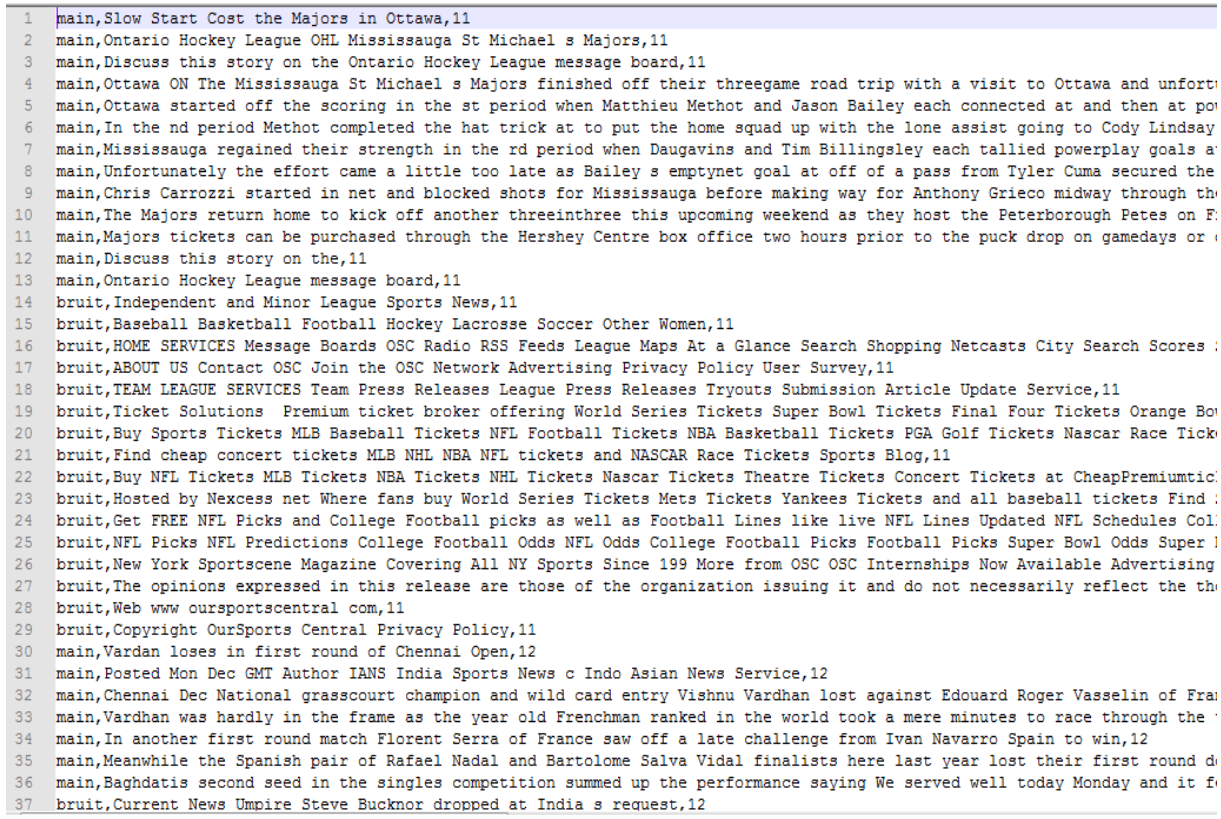
Après l'extraction de contenu à partir de toutes les pages web de la base d'apprentissage, On fait la **concaténation** des fichiers obtenus pour construire un fichier globale résumant toute la base d'apprentissage. Comme il est illustré dans la figure suivante.



```
main,op ed john esposito,1
main,published january,1
main,after bhutto,1
main,the world will long remember benazir bhutto as a modern muslim woman who served two terms as pakistan s first woman
main,bhutto was an avowed reformer who in two terms as prime minister failed to bring major political or social change a
main,like her father she exerted power through an increasingly tough autocratic style one person dominance or rule she d
main,the recent political responses to bhutto s assassination highlight the key problems or fault lines endemic to pakis
main,but as dangerous as these forces are especially with the growth of pakistani rather than foreign fighters this sing
main,although muhammad ali jinnah pakistan s founder and first leader saw pakistan as a muslim homeland his socio cultur
main,zulfikar ali bhutto a secular socialist would himself turn to islam after the pakistan bangladesh civil war in in c
main,where do we go from here the pakistan u s war on terrorism and promotion of democracy have in fact resulted in a da
main,moving forward will require an enlightened leadership at a time when widespread anti americanism more accurately op
main,musharraf should begin with the restoration of some semblance of democracy by reconstituting pakistan s supreme cou
main,john l esposito university professor and founding director of the prince alwaleed bin talal center for muslim chris
bruit,search go,1
bruit,homepage international politics security business technology editorial opinion,1
bruit,advertisement most popular,1
bruit,online resources europe accomodation celebrity bio geldgeschenke ideen golfurlaub hair extensions hausprospekte ka
main,peterson eyes nfc rushing title playoffs,2
main,by the associated press,2
main,denver ap the nfc rushing title and the playoffs are both within adrian peterson s grasp,2
main,the minnesota vikings can clinch a playoff berth with a win at denver and some help from dallas today when peterson
main,peterson isn t expecting anything to come easy at frigid invesco field even though the broncos rank th in the nfl a
main,they are going to give us their best shot peterson said we know they will try to finish strong,2
main,all the broncos have to play for is pride although a loss would prove more meaningful come april when denver would
main,in that case the broncos might get lucky and land an impact playmaker such as peterson who was selected seventh by
main,minnesota s leads the league in rushing behind peterson yards more than philadelphia s brian westbrook and touchdow
main,when the broncos put in a vikings run reel this week safety john lynch turned to his coach and wondered are you sur
main,since shredding san diego s defense for an nfl record yards on nov however peterson has seen teams crowd the line c
main,is it frustrating a little bit at times peterson allowed you know what it s going to be i can t remember the last t
main,the broncos went with the eight man front to slow a leaky defense bringing lynch closer to the line after getting g
main,denver has been dogged all year by injuries ineffectiveness and inconsistency ruining a season dedicated to fallen
main,making the downfall even more painful are the high hopes they began with following an offseason roster retooling th
main,instead the broncos are slogging through their first losing season since when john elway retired after consecutive
main,coach mike shanahan called this his hardest season ever worse than when the broncos slid to in shanahan s only othe
main,lynch who will ponder retirement in the offseason said it s easy to discount a game like this as a throwaway game b
main,although the broncos might get long looks at some young players lynch suggested the effort will be top notch,2
```

Fig43 : La base d'apprentissage extraite.

Après la construction de la base d'apprentissage, On va utiliser le même processus pour construire la base de teste qu'on va utiliser dans ce projet.



```
main,Slow Start Cost the Majors in Ottawa,11
main,Ontario Hockey League OHL Mississauga St Michael s Majors,11
main,Discuss this story on the Ontario Hockey League message board,11
main,Ottawa ON The Mississauga St Michael s Majors finished off their threegame road trip with a visit to Ottawa and unfort
main,Ottawa started off the scoring in the st period when Matthieu Methot and Jason Bailey each connected at and then at po
main,In the nd period Methot completed the hat trick at to put the home squad up with the lone assist going to Cody Lindsay
main,Mississauga regained their strength in the rd period when Daugavins and Tim Billingsley each tallied powerplay goals a
main,Unfortunately the effort came a little too late as Bailey s emptynet goal at off of a pass from Tyler Cuma secured the
main,Chris Carrozzi started in net and blocked shots for Mississauga before making way for Anthony Grieco midway through th
main,The Majors return home to kick off another threeinthree this upcoming weekend as they host the Peterborough Petes on F
main,Majors tickets can be purchased through the Hershey Centre box office two hours prior to the puck drop on gamedays or
main,Discuss this story on the,11
main,Ontario Hockey League message board,11
bruit,Independent and Minor League Sports News,11
bruit,Baseball Basketball Football Hockey Lacrosse Soccer Other Women,11
bruit,HOME SERVICES Message Boards OSC Radio RSS Feeds League Maps At a Glance Search Shopping Netcasts City Search Scores
bruit,ABOUT US Contact OSC Join the OSC Network Advertising Privacy Policy User Survey,11
bruit,TEAM LEAGUE SERVICES Team Press Releases League Press Releases Tryouts Submission Article Update Service,11
bruit,Ticket Solutions  Premium ticket broker offering World Series Tickets Super Bowl Tickets Final Four Tickets Orange Bo
bruit,Buy Sports Tickets MLB Baseball Tickets NFL Football Tickets NBA Basketball Tickets PGA Golf Tickets Nascar Race Tick
bruit,Find cheap concert tickets MLB NHL NBA NFL tickets and NASCAR Race Tickets Sports Blog,11
bruit,Buy NFL Tickets MLB Tickets NBA Tickets NHL Tickets Nascar Tickets Theatre Tickets Concert Tickets at CheapPremiumtic
bruit,Hosted by Nexcess net Where fans buy World Series Tickets Mets Tickets Yankees Tickets and all baseball tickets Find
bruit,Get FREE NFL Picks and College Football picks as well as Football Lines like live NFL Lines Updated NFL Schedules Col
bruit,NFL Picks NFL Predictions College Football Odds NFL Odds College Football Picks Football Picks Super Bowl Odds Super
bruit,New York Sportscene Magazine Covering All NY Sports Since 199 More from OSC OSC Internships Now Available Advertising
bruit,The opinions expressed in this release are those of the organization issuing it and do not necessarily reflect the th
bruit,Web www oursportscentral com,11
bruit,Copyright OurSports Central Privacy Policy,11
main,Vardan loses in first round of Chennai Open,12
main,Posted Mon Dec GMT Author IANS India Sports News c Indo Asian News Service,12
main,Chennai Dec National grasscourt champion and wild card entry Vishnu Vardhan lost against Edouard Roger Vasselin of Fra
main,Vardhan was hardly in the frame as the year old Frenchman ranked in the world took a mere minutes to race through the
main,In another first round match Florent Serra of France saw off a late challenge from Ivan Navarro Spain to win,12
main,Meanwhile the Spanish pair of Rafael Nadal and Bartolome Salva Vidal finalists here last year lost their first round d
main,Baghdatis second seed in the singles competition summed up the performance saying We served well today Monday and it f
bruit,Current News Umpire Steve Bucknor dropped at India s request,12
```

Fig44 : La base de teste extraite.

Et on crée un troisième fichier **csv** qui contient les noms de classes pour l'apprentissage :

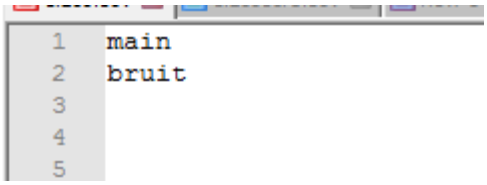


Fig45 : Les classes.

Donc comme résultat final, on obtient trois fichiers : **training_data_set.csv**, **test.csv**, et **class.csv**.

2. Préparation de données pour l'apprentissage

Pour préparer les données à l'apprentissage, On suit les étapes suivantes

2.1. Prétraitement de données

2.1.1. Convertir les lettres en minuscules

Pour unifier la forme de la base d'apprentissage, On transforme les lettres de la base en minuscules.

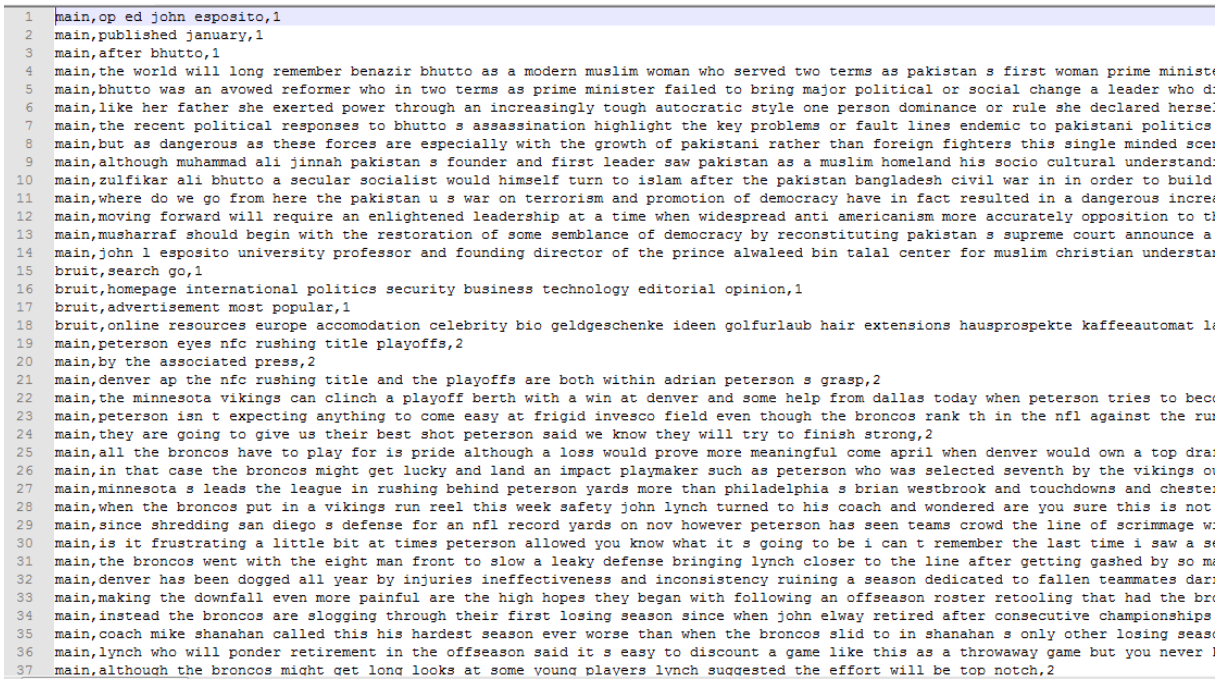


Fig46 : La base d'apprentissage en minuscule.

2.1.2.Suppression des mots vides

Avant la suppression des mots vides, On va appliquer tout d'abord la tokenization sur les données d'apprentissage (texte), où On va utiliser l'outil nltk.

```
from keras.preprocessing.text import Tokenizer
from nltk.tokenize import sent_tokenize, word_tokenize
```

```
  y_train.append(all[i][0])
  X.append(nltk.word_tokenize(all[i][1]))
```

Fig47 : Tokenization des données.

Après la tokenization des données, On va supprimer les mots vides contenus dans la base d'apprentissage.

```
from nltk.corpus import stopwords
# ---------------------remove stop word -------------------#
for w in doc:
    for word in w:
        if not word in stopset:
            stop_word.append(word)
```

Fig48 : Suppression des mots vides.

Où **stopset** contient la liste des tous les mots vides existe dans la langue anglais puis nous avons parcouru notre base et supprimer les mots vides.

2.1.3.Lemmatisation

Après l'élimination des mots vides, On applique la lemmatisation sur notre base.

```
from nltk.stem.wordnet import WordNetLemmatizer
##--------------------lemmatisation --------------------##
for lem in stop_word:
    lemmatisation.append(lemmatiser.lemmatize(lem))
```

Fig49 : La lemmatisation.

2.1.4.Stemming

Comme dernière étape du prétraitement, On applique la phase Stemming.

```
from nltk.stem import PorterStemmer
##--------------------stemming --------------------##
for stem in lemmatisation:
    stemming.append(stemmer.stem(stem))
```

Fig50 : Stemming.

Après la phase du prétraitement, On obtient un contenu nettoyé et prêt pour la représentation vectorielle.

2.2. Word2Vector

Notre base d'apprentissage est un ensemble de mots, elle représente l'entré de notre CNN. Mais le CNN accepte comme entré sauf les valeurs numériques. Et pour cela, On va utiliser la méthode Word2Vectqui va convertir le contenu textuel dans la base d'apprentissage vers des valeurs numériques.

- Dans la base d'apprentissage : On va représenter chaque mot avec un vecteur comprend 5 numéros de type float.

Exemple : le mot 'op' et le mot 'world' dans la base d'apprentissage vont être codé comme suit :

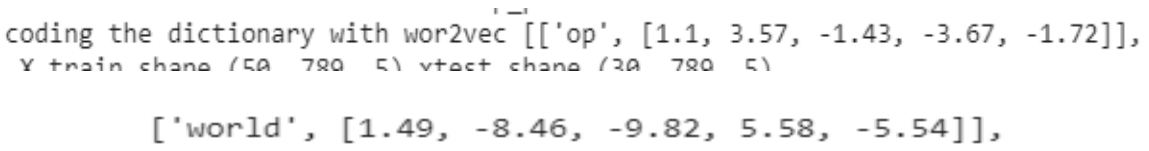


Fig51 : Codage de la base d'apprentissage.

Les paramètres principaux dumodèleword2vector, sont:

- La liste d'entrée : textes d'entrées (la liste 'stm' contient la base d'apprentissage).
- Taille : Spécifie la taille du vecteur représentant du chaque mot, par défaut=100.
- Min-count : Nombre minimal d'apparitions d'un mot à prendre en compte pour l'apprentissage du modèle. On donne 1 pour mettre tous les mots pris en compte.

Fig52 : Le modèle Word2Vector.

On a aussi donné une représentation vectorielle pour nos classes comme suit :

```
class to vector [['main', 1], ['bruit', 2]]
```

Fig53 : Codage de classes.

Avec la méthode suivante :

```
u=1
for m in classes:
   if not u in tr:
      tr.append(u)
      class_train_vec.append([m,u])
   u+=1
print("class to vector",class_train_vec)

return class_train_vec
```

Fig54 : Méthode de codage de classe.

3. Apprentissage de CNN

Dans ce projet, On a construit un modèle de CNN. Ce dernier est composé de trois couches convolutives et une couche entièrement connectée.

- La première couche convolutive est composée de 100 filtres de taille 3*3 chacune de couches de convolution est suivie d'une fonction d'activation ReLU cette fonction force les neurones à retourner des valeurs positives, ensuite on applique Max pooling sur les features maps afin de réduire la taille de texte ainsi la quantité.

À la sortie de cette couche, On a 100 features maps de taille moitié de la taille d'origine. On répète cette procédure avec les couches de convolutions deux et trois.

La fonction d'activation ReLU est appliquée toujours sur chaque convolution. Un Global Max pooling est appliquée après la troisième couche de convolution.

À la sortie de cette couche, On a un vecteur de caractéristiques de dimension égale à 100.

Après ces trois couches de convolution, On utilise un réseau de neurone composé de deux couches entièrement connectées, la première couche est composée de 280 neurones où la fonction d'activation utilisée est le ReLU et la deuxième couche est une soft max qui permet de calculer la distribution de probabilité des 2 classes (principal ou bruit).

```
'''****************** the model training ******************'''
def CNN(features,y_train_c,len_train):
    max_features =len_train
    batch_size = 10
    hidden_dims =280
    epoches = 15
    model = Sequential()
    model.add(Conv1D(filters=100, kernel_size=3, padding='same',
        input_shape=(features.shape[1], features.shape[2]), activation='relu'))
    model.add(MaxPooling1D(pool_size=2))
    model.add(keras.layers.Dropout(0.2))
    model.add(Conv1D(filters=100,kernel_size=4,padding='same',strides=1))
    model.add(MaxPooling1D(pool_size=2))
    model.add(keras.layers.Dropout(0.2))
    model.add(Conv1D(filters=100,kernel_size=5,padding='same',strides=1))
    model.add(GlobalMaxPooling1D())
    model.add(Dense(hidden_dims))
    model.add(Dropout(0.2))
    model.add(Activation('relu'))
    model.add(Dense(y_train_c.shape[1]))
    model.add(Activation('sigmoid'))
 # We add a vanilla hidden layer
    model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
    print(model.summary())
    model.fit(features, y_train_c, epochs=epoches, batch_size=batch_size, verbose=1)
    return model
```

Fig55 : Le modèle de CNN.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_1 (Conv1D)            (None, 789, 100)          1600
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 394, 100)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 394, 100)          0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 394, 100)          40100
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 197, 100)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 197, 100)          0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 197, 100)          50100
_________________________________________________________________
global_max_pooling1d_1 (Glob (None, 100)               0
_________________________________________________________________
dense_1 (Dense)              (None, 280)               28280
_________________________________________________________________
dropout_3 (Dropout)          (None, 280)               0
_________________________________________________________________
activation_1 (Activation)    (None, 280)               0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 562
_________________________________________________________________
activation_2 (Activation)    (None, 2)                 0
=================================================================
Total params: 120,642
Trainable params: 120,642
Non-trainable params: 0
```

Fig56 : Le résultat du modèle CNN utilisé.

4. Résultats Expérimentaux

Après la création de notre modèle CNN, On va l'entraîner. Pour entraîner un modèle, il suffit d'appeler la méthode **fit** qui prend en arguments :

- Les données d'apprentissage **'features'**.
- Les prédictions attendues d'apprentissage, ici **'y_train_c'**
- Le nombre d'itérations de l'entraînement **'epochs'**.
- Et enfin le paramètre **batch_size**, qui est le nombre de données d'entrées à analyser, qui vaudra 10 (valeur standard).

```
model.fit(features, y_train_c, epochs=epoches, batch_size=batch_size, verbose=1)
```

Fig57 : La méthode fit pour entrainer le modèle CNN.

On a testé notre modèle sur la base de test, On a testé la méthode fit avec différent nombre d'itérations d'apprentissage '**epochs'** pour obtenir le meilleur résultat d'apprentissage.

Pour epochs = 10, On obtient :

- le nombre d'itération epochs = 10, On obtient :

```
Epoch 1/10
50/50 [==============================] - 1s 17ms/step - loss: 0.7673 - acc: 0.6800
Epoch 2/10
50/50 [==============================] - 0s 7ms/step - loss: 0.6772 - acc: 0.7000
Epoch 3/10
50/50 [==============================] - 0s 7ms/step - loss: 0.5819 - acc: 0.7500
Epoch 4/10
50/50 [==============================] - 0s 7ms/step - loss: 0.5367 - acc: 0.7700
Epoch 5/10
50/50 [==============================] - 0s 7ms/step - loss: 0.4904 - acc: 0.7500
Epoch 6/10
50/50 [==============================] - 0s 7ms/step - loss: 0.4877 - acc: 0.7500
Epoch 7/10
50/50 [==============================] - 0s 7ms/step - loss: 0.4353 - acc: 0.8000
Epoch 8/10
50/50 [==============================] - 0s 7ms/step - loss: 0.3945 - acc: 0.8200
Epoch 9/10
50/50 [==============================] - 0s 7ms/step - loss: 0.3892 - acc: 0.7900
Epoch 10/10
50/50 [==============================] - 0s 7ms/step - loss: 0.3423 - acc: 0.8900
30/30 [==============================] - 0s 5ms/step
Test score: 0.6980658769607544
```

Fig58 : L'entrainement du CNN avec epochs = 10.

Après 10 itérations d'apprentissage, le **taux de reconnaissance de notre CNN égale à69.8%**

- le nombre d'itération epochs = 15, On obtient :

```
Epoch 1/15
50/50 [==============================] - 1s 20ms/step - loss: 0.7281 - acc: 0.7300
Epoch 2/15
50/50 [==============================] - 0s 8ms/step - loss: 0.6267 - acc: 0.6500
Epoch 3/15
50/50 [==============================] - 0s 8ms/step - loss: 0.5032 - acc: 0.7700
Epoch 4/15
50/50 [==============================] - 0s 8ms/step - loss: 0.5163 - acc: 0.7500
Epoch 5/15
50/50 [==============================] - 0s 8ms/step - loss: 0.4278 - acc: 0.8300
Epoch 6/15
50/50 [==============================] - 0s 8ms/step - loss: 0.4522 - acc: 0.7800
Epoch 7/15
50/50 [==============================] - 0s 8ms/step - loss: 0.4578 - acc: 0.8100
Epoch 8/15
50/50 [==============================] - 0s 8ms/step - loss: 0.3957 - acc: 0.7800
Epoch 9/15
50/50 [==============================] - 0s 8ms/step - loss: 0.3389 - acc: 0.8300
Epoch 10/15
50/50 [==============================] - 0s 8ms/step - loss: 0.3103 - acc: 0.8900
Epoch 11/15
50/50 [==============================] - 0s 8ms/step - loss: 0.2220 - acc: 0.9100
Epoch 12/15
50/50 [==============================] - 0s 8ms/step - loss: 0.2179 - acc: 0.9400
Epoch 13/15
50/50 [==============================] - 0s 8ms/step - loss: 0.1464 - acc: 0.9800
Epoch 14/15
50/50 [==============================] - 0s 8ms/step - loss: 0.1194 - acc: 0.9600
Epoch 15/15
50/50 [==============================] - 0s 8ms/step - loss: 0.1130 - acc: 0.9700
30/30 [==============================] - 0s 4ms/step
Test score: 0.9203873872756958
```

Fig59 : L'entrainement du CNN avec epochs = 15.

Après 10 itérations d'apprentissage, le **taux de reconnaissance de notre CNN égale à 92.03%**

- le nombre d'itération epochs = 20, On obtient :

```
Epoch 5/20
50/50 [==============================] - 0s 7ms/step - loss: 0.5511 - acc: 0.7400
Epoch 6/20
50/50 [==============================] - 0s 7ms/step - loss: 0.4980 - acc: 0.7400
Epoch 7/20
50/50 [==============================] - 0s 7ms/step - loss: 0.4348 - acc: 0.7900
Epoch 8/20
50/50 [==============================] - 0s 7ms/step - loss: 0.4014 - acc: 0.7900
Epoch 9/20
50/50 [==============================] - 0s 7ms/step - loss: 0.3937 - acc: 0.8200
Epoch 10/20
50/50 [==============================] - 0s 7ms/step - loss: 0.4667 - acc: 0.7900
Epoch 11/20
50/50 [==============================] - 0s 7ms/step - loss: 0.2261 - acc: 0.9600
Epoch 12/20
50/50 [==============================] - 0s 7ms/step - loss: 0.2337 - acc: 0.9500
Epoch 13/20
50/50 [==============================] - 0s 7ms/step - loss: 0.2066 - acc: 0.9000
Epoch 14/20
50/50 [==============================] - 0s 7ms/step - loss: 0.1558 - acc: 0.9800
Epoch 15/20
50/50 [==============================] - 0s 7ms/step - loss: 0.1139 - acc: 0.9900
Epoch 16/20
50/50 [==============================] - 0s 7ms/step - loss: 0.0812 - acc: 0.9900
Epoch 17/20
50/50 [==============================] - 0s 7ms/step - loss: 0.0642 - acc: 0.9900
Epoch 18/20
50/50 [==============================] - 0s 7ms/step - loss: 0.0552 - acc: 1.0000
Epoch 19/20
50/50 [==============================] - 0s 7ms/step - loss: 0.0401 - acc: 1.0000
Epoch 20/20
50/50 [==============================] - 0s 7ms/step - loss: 0.0545 - acc: 0.9900
30/30 [==============================] - 0s 6ms/step
Test score: 0.8187849521636963
```

Fig60 : L'entrainement du CNN avec epochs = 20.

Après 10 itérations d'apprentissage, le **taux de reconnaissance de notre CNN égale à 81.8%**

Discussion des résultats

Donc, après plusieurs essayes, On a conclu que notre modèle CNN est plus performons de connaitre le contenu principal et le contenu bruit d'un page web après 15 itérations d'apprentissage avec un taux de reconnaissance= **92.03%.**

IV. Conclusion

Dans ce chapitre, On a présenté l'implémentation de notre système : L'environnement et les outils de développement, ensuite on a présenté quelques expérimentations appliquées sur notre base des pages Web puis on exposé les résultats obtenus en terme de taux de reconnaissance.

CONCLUSION GÉNÉRALE

Conclusion générale

Le processus de l'extraction de contenu de la page web est devenu de plus en plus importants de jour en jour, à cause de la grande quantité de données gérées par Internet.

Dans ce mémoire, On a présenté une méthode d'extraction de données à partir du web en utilisant un algorithme d'apprentissage profond. On a considéré la tâche d'extraction de contenu en tant que problème de classification.

On a prendre en premier temps une vue générale sur l'extraction de contenu et les techniques utilisés pour l'extraction. Puis on a présenté l'apprentissage profond et le réseau de neurone à convolution car Pour présenter ce système, On a besoin d'une méthode de classification, la méthode la plus efficace et la plus utilisée est la méthode CNN qu'on a déjà expliqué dans le deuxième chapitre.

Puis on a proposé une conception détaillée et on a situé les étapes et les outils pour implémenter notre système, On a utilisé le Web scraping pour extraire les données à partir de la page web et les classer manuellement comme contenu principal ou bruyant, dans la tâche de préparation de donnée pour l'apprentissage. Après la préparation de données, On a utilisé la méthode Word2Vector pour les présenter sous forme d'un vecteur pour entraîner notre modèle d'apprentissage profond.

Après l'apprentissage de CNN, On a évalué notre modèle à l'aide d'une base de test (comme la base d'apprentissage, On a aussi extrait le contenu de la base de test manuellement avec le web scraping à partir des différentes pages web). On a trouvée que notre système est capable de séparer entre le contenu principal et le contenu bruit par 92%.

BIBLIOGRAPHIE

[1] METOMO JOSEPH BERTRAND RAPHAËL, Machine Learning : Introduction à l'apprentissage automatique, Article Publié le 10/10/2017 à 13:17:20.

[2] IDIOU GHNIA, Régression et modélisation par les réseaux de neurones, mémoire pour l'obtention du diplôme de magistère, 30 juin 2009.

[3] MARC PARIZEAU, Réseaux de neurones, 2006.

[4] HAOHAN WANG, BHIKSHA RAJ, On the origin of deep learning, 3 Mars 2017.

[5] MOUALEK DJALOUL YOUCEF, Deep Learning pour la classification des images, Mémoire de master, 2016-2017.

[6] CHRISTIAN GAGN, Apprentissage profond, Apprentissage et reconnaissance, 30 novembre 2016.

[7] G. DREYFUS, Les réseaux de neurones, septembre 1998

[8] CLAUDE TOUZET, Les réseaux de neurones artificiels introduction au connexionnisme cours, exercices et travaux pratique, Juillet 1992

[9] ALP MESTAN, Introduction aux Réseaux de Neurones Artificiels FeedForward, 2008

[10] KADOUS DJAMILA, Utilisation des réseaux de neurones comme outil du datamining : Génération de modèle comportemental d'un processus physique à partir de données, MEMOIRE DE MASTER, 28/06/2012

[11] REMY SUN, Apprentissage profond et acquisition de représentations latentes de séquences peptidiques, 28 février 2017

[12] CHARLES CROUSPEYRE, Comment les Réseaux de neurones à convolution fonctionnent, 17 juillet 2017.

[13] FLORENT SIMON, Deep Learning, le réseau à convolution, Publié le 05/10/2018 à 22:19:20

[14] PASCAL VINCENT, Modèles à noyaux à structure locale, Thèse présentée à la Faculté des études supérieures en vue de l'obtention du grade de Philosophioe Doctor (Ph.D.) en informatique, 2003

[15] MOKHTAR TAFFAR, Initiation à l'apprentissage automatique.

[16] OLIVIER CHAPELLE, BERNHARD SCHOLKOPT, et ALEXANDER ZIEN, Semi-Supervised Learning, 2006

[17] MARC BOULLE, Apprentissage supervisé, Séminaire TSI – 28/09/2006

[18] FRANÇOISE FESSANT, Apprentissage non supervisé, 28/09/2006

[19] Mme AURORE JAUMARD-HAKOUN, Modélisation et synthèse de voix chantée à partir de descripteurs visuels extraits d'images échographiques et optiques des articulateurs, Thèse de doctorat de l'université PIERRE ET MARIE CURIE.

[20] EL MAHDI BRAKNI, Ré réseaux de neurones artificiels appliqués à la méthode électromaghétique transitoire infinitem, mémoire présenté l'université du QUÉBEC à CHICOUTIMI comme exigence partielle de la maitrise ingénierie, Mai 2011.

[21] Dr. ABDELHAMID DJAFFEL, Cours Réseaux de Neurones, 2017-2018

[22] BOUGHABA MOHAMMED et BOUKHRIS BRAHIM, L`apprentissage profond (Deep Learning) pour la classification et la recherche d'images par le contenu, Mémoire Master Professionnel, 2016/2017

[23] JOHN A. BULLINARIA, Recurrent Neural Networks, 2015.

[24] BHARATI M. RAMAGERI / Indian Journal of Computer Science and Engineering, Vol. 1 No. 4 301-305, Data mining techniques and application.

[25] MEADI MOHAMED NADJIB, Technique basée HITS/SVM pour la réduction et la pondération des caractéristiques des pages Web, THÈSE pour obtenir le diplôme de Docteur en Sciences

[26] LOUKAM MOURAD, Technologies et services Web.

[27] LESZEK BORZEMSKI, The use of data mining to predict web performance, 23 Feb 2007.

[28] HARI MOHAN PANDEY et SHIPRA SAINI, Review on Web Content Mining Techniques, May 2015.

[29] BRIJENDRA SINGH et HEMANT SINGH, Web Data Mining research: A survey, January 2011

[30] S.VIDYA, K.BANUMATHY, Web Mining- Concepts and Application, 2015.

[31] J.SHARMILA, A.SUBRAMANI, A method for extracting information from the web using Deep Learning algorithm, 20th October 2014.

[32] HAMZA YUNIS, Content Extraction from Web pages Using Machine Learning, Master's Thesis, December 16, 2016.

[33] BO ZHAO, Web Scraping, May 2017.

[34] ABBASSI MEFTAH, Les systèmes de recherche d'information.

[35] Mme DALILA BEBBOUCHI BENELKEZADRI, Recherche d'Information.

[36] DAVID MEYER, How exactly does word2vec work?, 2016.

[37] BELKHIRI KHADIDJA, Analyse des textes en utilisant le deep learning, Mémoire présenté pour obtenir le diplôme de master académique en Informatique, 2018.

[38] Jean-Daniel Bonjour, Python et outils associés - Installation et utilisation.

[39] TIAGO CARNEIRO, RAUL MEDEIROS , THIAGO NEPOMUCENO, GUI-BIN BIAN, VICTOR HUGO C. DE ALBUQUERQUE, AND PEDRO P. REBOUÇAS FILHO, Python Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications, .