République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université ABBAS LAGHROUR Khenchela

Faculté des Sciences et Technologies

Département Mathématique et Informatique

Mémoire de fin d'étude pour l'obtention de diplôme de master informatique

Option

Sécurité et Technologies Web

Thème

CLASSIFICATION DES PAGES WEB BASEES SUR UNE TECHNIQUE D'EXTRACTION DES MOTS CLES

Encadré par :

- **Dr.** HIOUAL Ouided

Réalisé par :

- Dridi Soumia
- Sid Amel

Année Universitaire 2020/2021

Dédicaces

Au nom de Dieu, le tout puissant, le Clément, le Miséricordieux…

Je dédie ce travail à ma famille, mes chers parents qui m'ont doté d'une éducation digne, qui m'ont soutenu durant toutes ces années pour mener ce projet à bien. Qu'ils trouvent ici le témoignage de ma profonde reconnaissance.

A mes sœurs, Naïma, Imen et Donia pour leur soutient inestimable et leur amour incommensurable lors de la rédaction de ce projet.

A mes frères Adel, Ali et Salah qui m'ont toujours encouragé. Certains seront absents aujourd'hui mais je les portes avec moi dans mon cœur.

A ma belle-sœur Amel pour avoir rendu notre vie plus joyeuse !

A mes neveux et nièce prunelles de mes yeux, Anis, Ishaq, Adam et Maria que Dieu vous permette d'aller aussi loin dans vos études Incha Allah.

A mes chers amis Sid Amel, Aib chaima.

A tous mes amis, pour leurs encouragements et leurs accompagnements et à qui je souhaite plus de succès …

A tous ceux que j'aime !!!

MERCI

Dridi Soumia

Dédicace

La dédicace primordiale va à ceux qui m'ont amené à l'existence et m'a fait la personne que je suis aujourd'hui, à l'affection, l'amour et le soutien que ma mère « **Saida** » incarne, et à la personne que je respecte et chéris, mon père et mon modèle « **Taher** »

À ceux qui sont toujours préoccupés par ma santé et mon bien-être, à ma tendre grand-mère « **Hafssia** »

Aux bougies avec lesquelles j'ai été guidé dans les ténèbres de la vie Aux perles dont l'éclat a orné ma vie Mes frères « **Ameur, Nafaa, Ibrahim, Okba** » et mes sœurs « **Nahed, Abla, Nour-el-yakinne**»

A qui mes espoirs et ses espoirs et rêves se mêlaient aux siens. A mon compagnon dans cet effort« **Dridi Soumia** ».

Sid Amel

Remerciements

Le prophète - que la paix et les bénédictions d'Allah soient sur lui- a dit : « L'acquisition de la science est une obligation incombant à chaque musulman. » (D'après Anas ibn Malik)

A tout début, il y a une fin…

Tout d'abord, merci à Allah le tout puissant de nous avoir permis de poursuivre nos études qui prendront fin avec ce mémoire …

La réalisation de ce mémoire a été possible grâce aux efforts de plusieurs personnes à qui nous souhaitons témoigner toutes nos gratitudes!

Nous souhaitons adresser toute nos reconnaissances à l'encadreur Dr Hioual Ouided, pour sa patience, sa disponibilité et surtout ses conseils judicieux, qui ont contribué à alimenter nos réflexions.

Nous désirerons aussi remercier les membres du jury qui nous fait l'honneur de bien vouloir évaluer notre travail, le Dr Hemam Mounine Soufiane ainsi que le Dr Chergui Outhaila, pour l'honneur qu'ils nous font, en acceptant la présidence de ce jury.

Enfin, nous tenons à témoigner toute notre gratitude , notre amour et notre profond respect à nos parents, après Dieu il y a eux ! Ces deux êtres chers qui nous ont permis d'arriver là où nous somme aujourd'hui ! Merci à eux de nous avoir éduqué et inculqué toutes leurs valeurs qui font de nous celles que nous somme aujourd'hui !

Une fin d'études certes mais ceux-là ne clôtureront pas nos envies d'apprendre et de savoir sur ce miracle qui est LA VIE !

SOMMAIRE

CHAPITRE 03. DESCRIPTION ET CONCEPTION DE LA METHODE PROPOSEE

CHAPITRE 04. MISE EN ŒUVRE DE LA METHODE PROPOSEE

LISTE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GENERALE

1. Contexte du travail

Nous assistons ces dernières années au développement accru du World Wide Web qui est devenu la principale source de données pour l'homme.

En effet, le web est devenu un outil essentiel pour un accès facile et rapide à l'information pour se renseigner, rechercher, apprendre et découvrir de nouvelles connaissances. Il permet de mieux répondre aux besoins toujours plus grandissants d'informations et de connaissances des internautes mais ces derniers sont de plus en plus submergés par ce volume mis à leur disposition.

La nécessité de disposer de nouvelles méthodes et d'outils avancés permettant de faciliter l'accès et répondre aux besoins des internautes, a incité les chercheurs à s'intéresser à un domaine appelé la classification

Depuis l'aube des temps, l'homme pratique la classification dans sa vie quotidienne, quand il essaie de répondre aux problèmes et questions sur la catégorie des objets, c'est-à-dire d'affectation d'objets à leur classe (en observant leurs formats, couleurs, tailles . . .etc.)

La classification joue un rôle essentiel dans de nombreuses tâches de gestion et de récupération de l'information sur le Web, la classification du contenu des pages est essentielle à l'exploration ciblée, au développement assisté d'annuaires Web, à l'analyse de liens Web spécifique à un sujet, à l'analyse contextuelle la publicité et à l'analyse de la structure thématique du Web. La classification des pages Web peut également contribuer à améliorer la qualité de la recherche sur le Web.

La classification des pages Web se fait généralement en extrayant le contenu textuel de la page et en ignorant les balises HTML et le code CSS et JS donc il faut une extraction des données clés dans la page web puis faire la classification. Il existe différentes approches et type de

classification des pages web(supervisé et non supervisé) et plusieurs techniques d'extraction automatique de mots clé (supervisé et non supervisé).

Les méthodes existantes pour la tâche d'extraction automatique de termes-clés sont soit supervisées, soit non-supervisées. Les méthodes non-supervisées sont des méthodes émergentes ayant la particularité de s'abstraire de la spécificité des données traitées. Cette abstraction s'explique par des approches basées sur des constatations à propos de ce qu'est un terme-clé au sens général : importance sémantique, degré d'information, structure syntaxique, etc. Contrairement aux méthodes non-supervisées, les méthodes supervisées n'utilisent pas de propriétés définies à partir des traits statistiques et linguistiques, mais elles utilisent des modèles de décision appris à partir de ces traits, calculés sur les termes-clés d'un corpus d'apprentissage. L'usage d'un corpus d'apprentissage implique que les modèles appris soient spécifiques au domaine disciplinaire et à la langue de celui-ci. Cette spécificité peut s'avérer avantageuse lorsque le domaine et la langue que représente le corpus sont les mêmes pour les documents qui sont ensuite analysés, mais si tel n'est pas le cas les résultats de l'extraction peuvent en pâtir.

2. Problématique et objectifs

De nos jours, les besoins de classification automatique des pages web en raison de l'augmentation constante du volume d'informations accessibles électroniquement, la conception et la mise en œuvre d'outils efficaces, permettant notamment à l'utilisateur de n'avoir accès qu'à l'information qu'il juge pertinente, deviennent une nécessité absolue. Comme la plupart de ces outils sont destinés à être utilisés dans un cadre professionnel, les exigences de fiabilité et de convivialité sont très importantes ; les problèmes à résoudre pour satisfaire ces exigences sont nombreux et difficiles.

Dans ce travail nous proposons une méthode de classification de page web fondé sur l'apprentissage automatique pour la classification et la technique TF-IDF pour l'extraction des mots clés des pages.

L'objectif principal de ce mémoire est l'extraction des mots-clés pour la classification des pages web est de rassembler les pages similaires selon une certaine catégorie, au sein d'une même classe ou catégorie à l'aide de la technique d'extraction des mots clés TF IDF et d'outil de classification les réseaux de neurones récurrents.

3. Organisation de mémoire

Le manuscrit est structuré en quatre chapitres et une conclusion générale.

Dans **le premier chapitre**, nous introduisons des notions générales sur les domaines : web, pages web, la classification en donnant quelques définitions, nous avons mentionné les approches de classification supervisé et non supervisé ensuite nous définissons la classification des pages web puis Les approches de classification des pages web et une étude comparative des algorithmes de classification les plus utilisées. Finalement les types de classification des pages.

Dans **le deuxième chapitre**, Nous commençons par les méthodes d'extraction automatique de termes-clés qui sont des méthodes supervisées et non supervisées en donnant des définitions de ces méthodes. Dans les méthodes non supervisées nous avons mentionné une approche statistique (TF IDF) et les méthodes à base de graphe (TextRank, SingleRank, TopicRank, Kcore) puis les approches par regroupement et pour finir nous avons discuté les différentes méthodes non supervisées.

Dans **le troisième chapitre**, nous présentons notre méthode proposée ainsi que sont fonctionnement.

Le **quatrième chapitre**, décrit les outils d'implémentation utilisés et expose un ensemble d'interfaces qui explique la description de notre méthode proposée

Enfin nous terminons ce manuscrit par conclusion générale accompagnée des futurs perspectives qui peuvent éventuellement améliorées la qualité des résultats obtenus par notre système, mais aussi ouvrir des portes vers d'autres idées ou approches à explorer, et une annexe montrant notre dataset utilisée dans ce travail et le code source des pages web utilisé.

Chapitre 01

La classification au profit des applications Web

1. Introduction

La classification joue un rôle essentiel dans de nombreuses tâches de gestion et de récupération de l'information. Sur le Web, la classification du contenu des pages est essentielle à l'exploration ciblée, au développement assisté d'annuaires Web, à l'analyse de liens Web spécifique à un sujet, à la publicité contextuelle et à l'analyse de la structure thématique du Web. La classification des pages Web peut également aider à améliorer la qualité de la recherche Web.

Dans ce chapitre, nous allons tenter de cerner les différentes approches et types de classification des pages web en commençant par quelques définitions de base, puis nous mentionnons les approches de classification supervisées et non supervisées dans un premier lieu, puis nous définissons la classification des pages web et ces approches, une étude comparative des algorithmes de classification les plus utilisées et finalement les types de classification des pages.

2. Web, Page web, Classification

2.1. Définitions

Le Web est un terme couramment utilisé pour désigner le World Wide Web ou WWW, traduit en français par World Spider Web. Il fait référence au système hypertexte s'exécutant sur le réseau informatique mondial Internet. En abusant du langage, le Web précise plus largement tous les contenus liés à ce monde Internet. Aujourd'hui, on ne fait plus toujours de distinction technique entre les contenus définis par le Web et les contenus définis par Internet [Selamat, A et all, 2004].

Page web : on entend tout document faisant partie d'un site Web et contenant des liens (également appelés hyperliens) pour faciliter la navigation entre les contenus.

Les pages Web sont développées à l'aide de langages de balisage tels que HTML et peuvent être interprétées par les navigateurs. Ainsi, les pages peuvent présenter des informations sous différents formats (texte, images, son, vidéo, animation), être associées à des données de style, ou contenir des applications interactives.

Il est possible de distinguer les pages web statiques (dont le contenu est prédéterminé) et les pages web dynamiques (contenu généré lorsque des informations sont demandées à un serveur web via un langage interprété comme JavaScript) [Selamat, A et all,2004].

Classification : la classification est un sujet lié de près ou de loin à de multiples domaines. Selon les objets qu'elle traite et les buts qu'elle vise à atteindre, elle est également connue sous diverses appellations (classification, clustering, segmentation...).

Pour attribuer la définition au terme « classification», il faut d'abord définir sa racine, qui vient du verbe « classer», qui signifie plus une action qu'un domaine, ou une série de méthodes plutôt qu'une théorie unifiée.

En mathématiques, la classification est la classification algorithmique des objets. Il s'agit d'attribuer une classe ou une catégorie à chaque objet (ou individu) à classer sur la base de données statistiques. Il utilise généralement des méthodes d'apprentissage et est largement utilisé pour la reconnaissance de formes. [Horri Mohamed et al., 2017].

D'une manière générale, selon ces définitions, la classification se définit alors comme une méthode mathématique d'analyse de données, pour faciliter l'étude d'une population d'effectif important, généralement des bases d'observations caractérisent un domaine particulier (animaux, plantes, malades, gènes, . . . etc.), où on les regroupe en plusieurs classes.

3. Les approches de classification

On distingue deux approches de classification, la classification supervisée : les classes sont connues à priori, par contre la classification non-supervisée (en anglais clustering) : les classes sont fondées sur la structure des objets.

3.1. Classification non supervisé (Clustering)

La classification non supervisée spécifie une série de méthodes visant à établir ou trouver une typologie existante qui caractérise un ensemble de n observations, de P caractéristiques mesurées sur chaque observation. Par typologie, nous entendons bien que les observations recueillies dans la même expérience ne proviennent pas toutes de la même Population homogène, mais plutôt de K populations [Oulai Siham et all ,2019].

La classification non supervisée consiste à trouver de manière automatique une organisation cohérente à un groupe de documents homogènes pour construire des regroupements cohérents(des classes ou clusters), elle correspond en statistiques au clustering, qui est également le terme utilisé en recherche d'informations.

Le clustering consiste donc, à diviser les objets en groupes sans connaître à priori leurs classes d'appartenance. Les techniques pour réaliser de tels regroupements constituent un domaine d'étude très riche.

Parmi les algorithmes de classifications non supervisées les plus connues, on peut citer par exemple [Quang, C. T, 2005]:

1. *K-means.*
2. *Single-pass.*
3. *Suffix tree clustering.*
4. *Hierarchical Agglomerative Clustering(HAC).*
5. *Les cartes auto organisatrices de Kohonen .*
6. *ART*

3.2. Classification supervisé (Catégorisation)

Contrairement à la classification non supervisé, nous commençons ici par un ensemble de classes connues et définies à l'avance. Nous disposons aussi d'une sélection initiale de données dont la classification est connue. Ces données sont supposées indépendantes et identiquement distribuées. Elles nous servent pour l'apprentissage de l'algorithme. La classification se fait par l'algorithme selon le modèle qu'il a appris [Ouali Siham et all ,2019].

Parmi les méthodes de classification supervisées les plus populaires, on peut citer par exemple [Quang, C. T, 2005]:

1. *Les k plus proches voisins.*
2. *les réseaux de neurones.*
3. *les arbres de décision.*
4. *les algorithmes génétiques.*
5. *Naïve Bayes.*
6. *Les machines à support de vecteurs.*

4. La classification des pages web

4.1. Définition

La classification (ou catégorisation) des pages web est un problème d'apprentissage automatique qui devient de plus en plus important jour après jour. Depuis le début d'Internet dans les années 90,le nombre d'utilisateurs d'Internet et le nombre de pages web destinées aux utilisateurs ont augmenté à un rythme rapide et continue de croître.

La classification de page Web est une application de recherche d'informations qui fournit des informations utiles qui peuvent être une base pour de nombreux domaines d'application différents. La catégorisation des pages Web fournit des informations utiles pour utilisation efficace d'Internet, filtrage des spams et de nombreux autres domaines d'application. Trouver

rapidement des résultats pertinents des millions de sites Web est un problème sérieux qui doit être résolu pour les moteurs de recherche. Pour cette raison, quelques moteurs de recherches nécessaires pour effectuer une classification thématique sur les pages Web afin que les résultats renvoyés aux utilisateurs puissent être retourné mieux. En plus de cela, les pages Web doivent être catégorisées afin que les politiques d'utilisation d'Internet puissent être déterminé pour les institutions ou les usages individuels [Ebubekir Buber et all ,2019]. La classification des pages Web peut également être utilisée par les applications cyber sécurité en bloquant les pages Web contenant du contenu malveillant avant qu'elles ne soient affichées par l'utilisateur. La classification automatique des pages Web nécessite le traitement de très grandes quantités de données. Manuel Page Web de la classification est un processus coûteux. Pour cette raison, les pages web classées manuellement constituent une très petite partie des pages Web actuelles. Le site web est le nom donné à la structure d'un groupe de pages web liés les uns aux autres de diverses manières. Un grand nombre de pages Web peuvent être trouvées sous un site Web. Les pages web sont liés les uns aux autres [Ebubekir Buber et all ,2019].

La classification de page web est le processus d'attribution d'une page web à l'une des catégories prédéterminées. Le problème de classification est un problème d'apprentissage automatique qui peut être résolu par des approches d'apprentissage supervisé. Les problèmes d'apprentissage supervisé se composent de 2 étapes de base. Ce sont la phase de formation et la phase de test. Un modèle peut être créés pour résoudre un problème de classification avec un certain nombre d'échantillons étiquetés. Ce modèle permet nous pour catégoriser les pages Web où nous ne connaissons pas les informations de catégorie [Ebubekir Buber et all ,2019].

Il existe également des études dans lesquelles les pages web sont catégorisées par des méthodes supervisées ainsi que des études classées par des méthodes non supervisées [Dural Burak, 2013]

Dans les approches non supervisées, les pages Web sont divisées en groupes selon des similitudes. Distance des calculs sont effectués pour déterminer les similitudes [Dural Burak, 2013] a optimisé les requêtes de recherche par clustering résultats des moteurs de recherche.

Afin de déterminer la catégorie d'une page web, il peut suffire d'analyser uniquement les informations sur la page web, cependant, pour pouvoir catégoriser un site web, il peut être nécessaire d'analyser toutes les pages web dans ce site web [Ebubekir Buber et all ,2019].

Sur la base de l'hypothèse que la page d'accueil d'un site Web fournit un bref résumé de ce site web, une partie des études n'ont catégorisé le site Web qu'en analysant la page principale du site

web. De cette façon, les sites web peuvent être catégorisés sans avoir à analyser de nombreuses pages web [Ebubekir Buber et all , 2019].

4.2. Les approches et les types de classification des pages web

4.2.1. Les approches de classifications

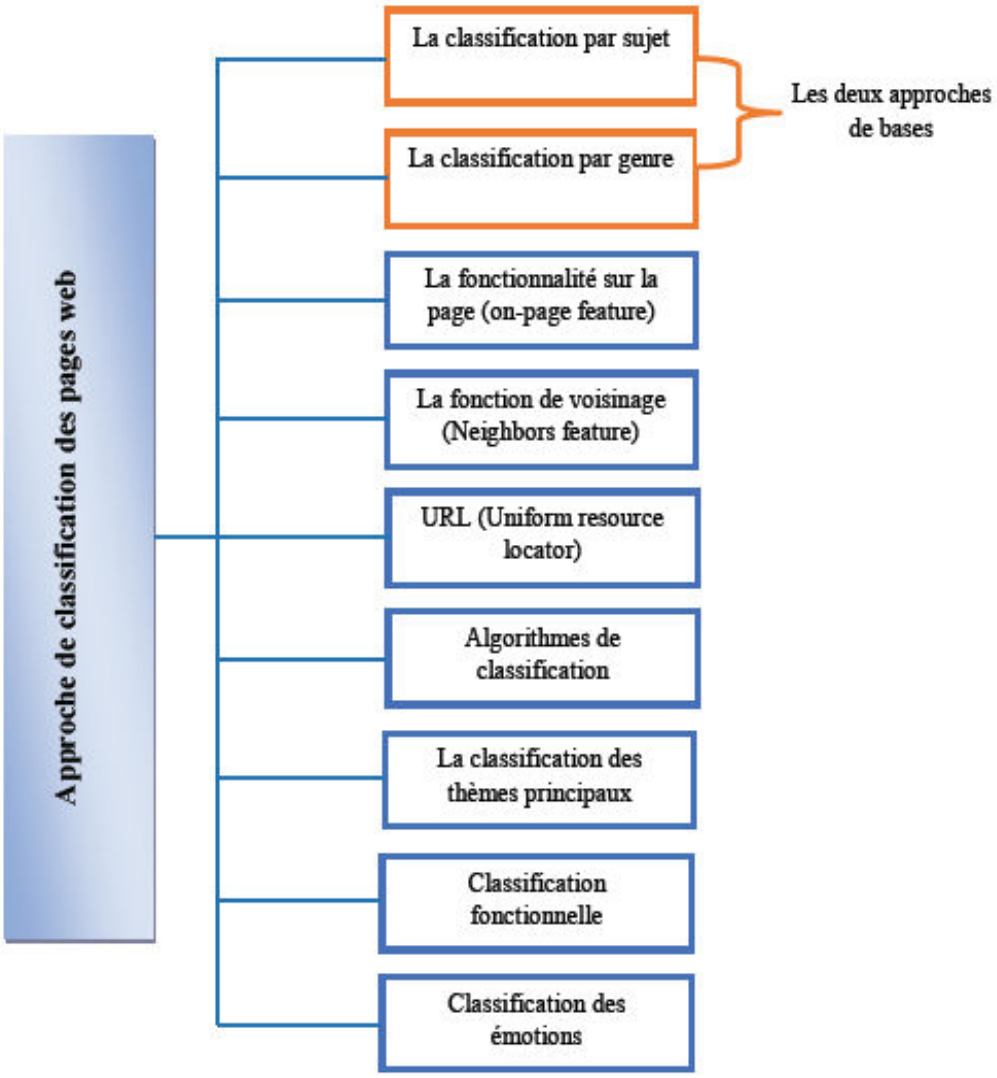

Figure 1.1 : Différentes approches de classification des pages web

La classification de page Web, également appelée catégorisation de page Web, est le processus d'attribution d'une page Web à une ou plusieurs étiquettes de classes (catégorie) prédéfinies. La classification est traditionnellement posée comme un problème d'apprentissage supervisé

[Mitchell ,1997] dans lequel un ensemble de données étiquetées est utilisé pour former un classificateur qui peut être appliqué pour étiqueter de futurs exemples (voir figure 1.1) .[Xiaoguang Qi and all, 2009].

La classification des pages Web est divisée en plusieurs sous-types en fonction des résultats de l'analyse. Une partie de ces types sont :

a. Les deux approches de base

La classification des pages Web est dans le domaine de l'apprentissage automatique, où l'apprentissage est sur les pages Web. L'utilisation de techniques d'apprentissage automatique sur des bases de données texte est appelé apprentissage de texte, qui a été bien étudié au cours des deux dernières décennies [Mladenic ,1998a]. L'apprentissage automatique sur les pages Web est similaire à apprentissage de texte puisque les pages Web peuvent être traitées comme des documents texte. Néanmoins, il est clair à l'avance que l'apprentissage sur les pages Web à des nouvelles caractéristiques. Premièrement, les pages Web sont des documents texte semi-structurés sont généralement écrits en HTML. Deuxièmement, les pages Web sont connectées à chaque d'autres formant des graphiques directs via des hyperliens. Troisièmement, les pages Web sont souvent courts et en utilisant uniquement du texte dans ces pages Web peut être insuffisant pour les analyser. Enfin, les sources des pages Web sont nombreuses, non homogènes, distribuées et en évolution dynamique. Afin de classer domaine Web si vaste et hétérogène, nous présentons dans ce chapitre deux approches de classification de base : classification par sujet et classification par genre et 7 différentes approches de classification des pages web les plus utilisé.

- ***La classification par sujet***

  Dans la classification par sujet (également appelée classification par sujet), les pages Web sont classées en fonction de leur contenu ou de leurs sujets. Cette approche définit de nombreuses catégories de sujets. Quelques exemples de catégories, sous les domaines « Science » utilisés dans Yahoo.com sont « Agriculture », « Astronomie », « Biologie », « Chimie », « Science cognitive », « Systèmes complexes » et "L'informatique". La classification par sujet peut être appliquée à créer des hiérarchies de sujets de pages Web, puis effectuer des recherches contextuelles de pages Web relatives à des sujets spécifiques **[**Choi B,2001].

- ***La classification par genre***

  Dans la classification basée sur le genre (également appelée classification basée sur le style), les pages Web sont classées en fonction de facteurs fonctionnels ou liés au genre. Au sens large, le mot « genre » est utilisé ici simplement comme une remplacer « une sorte de texte ». Quelques exemples de genres de pages Web sont « Catalogue de produits », « achats en ligne », « publicité », « appel à contribution », « Questions fréquemment posées », « page d'accueil » et « tableau d'affichage ». Ce l'approche peut aider les utilisateurs à trouver des intérêts immédiats. Bien que le genre de texte ait été étudié depuis longtemps dans la littérature linguistique, la classification automatique des genres de texte ne partage pas beaucoup de littérature et est moins sophistiquée des travaux de recherche ont été effectués sur la classification des genres de pages Web. Une raison importante est que, jusqu'à récemment, les collections numérisées ont été pour la plupart génériquement homogènes, comme les collections de résumés et articles de journaux. Ainsi le problème de l'identification des genres pourrait être mis de côté [Kessler et al, 1997]. Ce problème ne devient pas saillant jusqu'à ce que nous soyons confrontés à un domaine hétérogène comme le Web. Dans En fait, le Web est si diversifié qu'aucune taxonomie thématique ne peut espérer capturer tout sujets suffisamment détaillés. Il est à noter que le genre basé sur l'approche n'est pas de réduire l'importance de l'approche thématique, mais plutôt pour ajouter une autre dimension à la classification des pages Web dans son ensemble. **[**Choi B, 2001].

Dans la littérature il y a plusieurs approches mais nous avons choisi quelques autres approches qui sont en relation avec les approches de base que nous avons déjà parlé dans la section présidente.

b. La fonctionnalité sur la page (on-page feature)

La fonctionnalité sur la page (on-page feature) qui comprend les balises écrites et le contenu qui sont placés sur la page elle-même, l'une des fonctionnalités les plus directes que l'on puisse envisager d'utiliser est le contenu du texte. Une caractéristique notable qui ne semble pas les documents en texte brut mais les documents HTML sont des balises HTML [Xiaoguang Qi et all, 2009].

Pour augmenter les performances du classificateur, il a été confirmé que cela peut être fait en utilisant des informations dérivées de balises. Ainsi, en utilisant des balises, nous pouvons

profiter des informations structurelles entourées des fichiers HTML, ce qui est généralement négligé par les méthodes de texte brut.

Néanmoins, par la suite, les balises HTML maximales sont préoccupées par les symboles plutôt que par la sémantique, les page web auteurs peuvent créer différents mais correspondant théoriquement structure des balises. Ainsi, en utilisant les informations de balisage HTML dans la classification Web pourrait souffrir de la formation peu fiable des documents HTML [Pooja Vinod Nainwani et all,2018].

c. La fonction de voisinage (Neighbors feature)

Cette approche utilise la fonction de voisinage (Neighbors feature), afin d'identifier ce problème de classificateurs pour prendre des décisions sensées en fonction des fonctionnalités présentées sur la page Web, les fonctionnalités peuvent être retirer des pages voisines qui sont pertinentes d'une certaine manière à la page à classer pour fournir des informations pour la classification. Il existe de nombreuses façons de créer de telles connexions entre les pages, l'une des la connexion utilisée est le lien hypertexte [Xiaoguang Qi et all, 2009].

d. URL (Uniform Resource Locator)

Le localisateur de ressources uniforme (URL), cette approche à des montants plus rapide que la classification distincte des pages Web, car les pages Web elles-mêmes ne doivent pas être récupérées et analysé. Cette approche divise l'URL en expressive portions et ajoute un composant, séquentiel et fonctionnalités orthographiques pour modéliser des modèles pertinents. Les caractéristiques de résultant binaire sont utilisées dans l'entropie extrême supervisée démontrant. Il est examiné, l'utilité de l'approche dans classification binaire, multi-classes et hiérarchique. Une URL est d'abord divisée en jetons significatifs à l'aide de mesures théoriques. Ceci est important car peu de composants d'une URL ne sont pas entouré d'espaces (en particulier les noms de domaine). Ces jetons sont ensuite entrés dans un module d'analyse qui dérive caractéristiques composites importants pour la classification. Dans la seconde étape, l'apprentissage automatique est utilisé pour apporter une multi classe ou modèle de régression à partir d'URL d'entraînement catégorisées qui ont été traité par le module précédent [Kan MY et all, 2005].

e. Algorithmes de classification

Dans cette approche les arbres de décision sont principalement connus pour leur facilité et instinctivité. WEKA a été utilisé pour développer Arbres de décision [Witten IH et all, 2016], c'est-à-dire (Waikato Environment for Knowledge Analysis) qui est la collection de l'apprentissage automatique algorithmes utilisés pour les courses d'exploration de données [Witten IH et all,]. Différent algorithmes de classification ont également été envisagés, parmi lesquels Arbre du modèle logistique(Logistic Model Tree (LMT)), Meilleur premier arbre de décision (Best First Decision Tree (BFT)), Arbre élagué J48(J48 Pruned Tree (J48PT)) et arbre à greffer J48 (J48 Graft Tree (J48GT)).L'arbre de décision est développé en générant l'attribut d'objet Tableau après avoir rassemblé toutes les informations qui comprennent Liens externes (External links (EL)), longueur du texte de la page (Page's text length (TL)), image (Im),Liens internes (Internal links (IL)), Blog Word (Word Blog (WB)), Images externes (External Images (EI)), Objets multimédias (Multimedia Objects (MO)) Word Vidéo (Word Video (WV)), Word Flash (WF), Image Word (WI), Word News (WN), Images internes (Internal Images (II))[Pooja Vinod Nainwani et all,2018].

Le tableau 1.1. montre une étude comparative des algorithmes de classification les plus utilisées.

Numéro	Algorithmes	Caractéristiques	Limites
1	**Algorithme de plus proche K-Voisin**	Il n'est pas obligatoire que les cours soient séparables linéairement. Parfois c'est robuste à cause de certaines données d'entraînement bruyantes et peuvent être efficacement utilisé pour les classes multimodales	Lors de la recherche du plus proche voisin en grand ensemble de données de formation, cela prend trop de temps. Il est délicat à non pertinent ou bruyant les attributs. Le nombre de dimensions utilisées détermine les performances de l'algorithme.

2	**Algorithme de Bayes naïf**	Il a une efficacité de calcul décente et taux de classification et la mise en œuvre est simple. Il attend un résultat précis pour le problème de prédiction et classification.	La précision de l'algorithme diminue si le volume de l'ensemble de données est inférieur. Pour des résultats précis et bons en grande quantité de l'ensemble de données est requis.
3	**Machine à vecteur de soutien**	La précision est élevée. Fonctionne efficacement même lorsque les données sont linéairement ou non linéairement séparables.	Pour la formation et le test du l'exigence de vitesse et de taille est plus. Difficulté élevée et mémoire répandue exigences de classification.
4	**Réseau neuronal artificiel**	Il peut être facilement utilisé en ajustant uniquement quelques paramètres. Un réseau de neurones apprend et la reprogrammation n'est pas nécessaire. La mise en œuvre est simple et applicable à large éventail de problèmes réels.	Nécessite un temps de traitement élevé est-il grand réseau de neurones. Difficile de prévoir le nombre de couches et neurones nécessaires. Le processus d'apprentissage prend du temps.

Tableau 1.1 : Etude comparative des algorithmes de classification.[Pooja Vinod Nainwani et all,2018].

f. La classification des thèmes principaux

Classez la page Web en fonction du sujet. Par exemple : art, affaires, des sports.

j. Classification fonctionnelle

Classification de page Web selon son mécanisme fonctionnel. Par exemple : page d'accueil, page de connexion, page d'administration.

h. Classification des émotions

Classification faite pour comprendre l'opinion de l'auteur d'une certaine manière [Ebubekir Buber et all ,2019].

4.2.2. Les types de classification

Il existe plusieurs types de classifications qui sont :

a. La classification binaire

Catégorise instances dans exactement l'une des deux classes (voir la figure.1.2 (a) ci-dessous).

b. La classification multi-classe

Traite plus de deux classes. Basé sur le nombre de classes pouvant être attribuées pour une instance, la classification multi-classe peut être divisée en classification à étiquette unique et classification à étiquette multiple.

- ***La classification multi-classe à étiquette unique***

Dans la classification à étiquette unique, une et une seule étiquette de classe doit être assignée à chaque instance,

- ***La classification multi-classe multi-étiquettes***

Dans cette classification, plusieurs classes peuvent être assignées à une instance. Si un problème est multi-classe, par exemple, une classification à quatre classes, cela signifie que quatre classes sont impliquées, par exemple, les arts, les affaires, l'informatique et les sports. Il peut s'agir d'une seule étiquette, où exactement une étiquette de classe peut être attribuée à une instance (voir la figure.1.2 (b) ci-dessous), ou multi-étiquette, où une instance peut appartenir à un, deux ou tous des classes (voir la figure.1.2 (c) ci-dessous). Selon le type d'affectation de classe. Même ces deux classifications peuvent être divisées en classification dure et classification douce :

- *la classification dure* : est une instance qui peut être ou non dans une classe particulière, sans état intermédiaire (voir la figure.1.2 (b), (c) ci-dessous).

- *la classification douce* : est une instance qui peut être prédite comme étant dans une classe avec vraisemblance (souvent une distribution de probabilité dans toutes les classes, (voir la figure.1.2 (d) ci-dessous) [Xiaoguang Qi et all,2009].

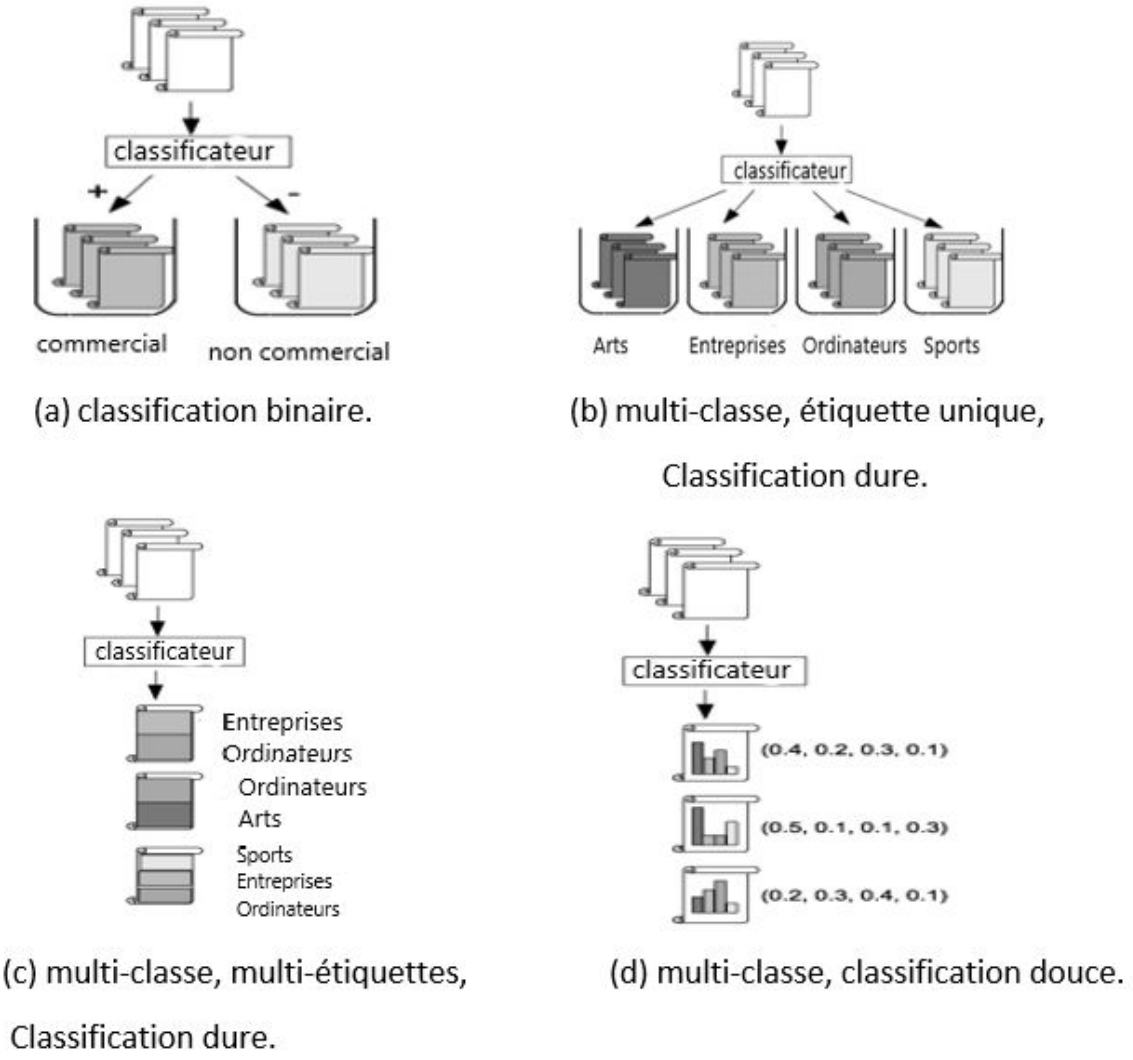

Figure 1.2 : Types de classification [Xiaoguang Qi et all, 2009]

On se basant sur l'organisation des classes, il existe d'autres types de classification qui sont : la classification plate et hiérarchique.

c. La classification plate

Dans cette classification les classes sont considérées comme parallèles, c'est-à-dire qu'une classe n'en remplace pas une autre (voir la figure.1.3 ci-dessous).

d. La classification hiérarchique

Les classes sont organisées dans une structure arborescente hiérarchique, dans laquelle chaque classe peut avoir un certain nombre de sous-classe (voir la figure.1.3 ci-dessous) [Xiaoguang Qi et all, 2009].

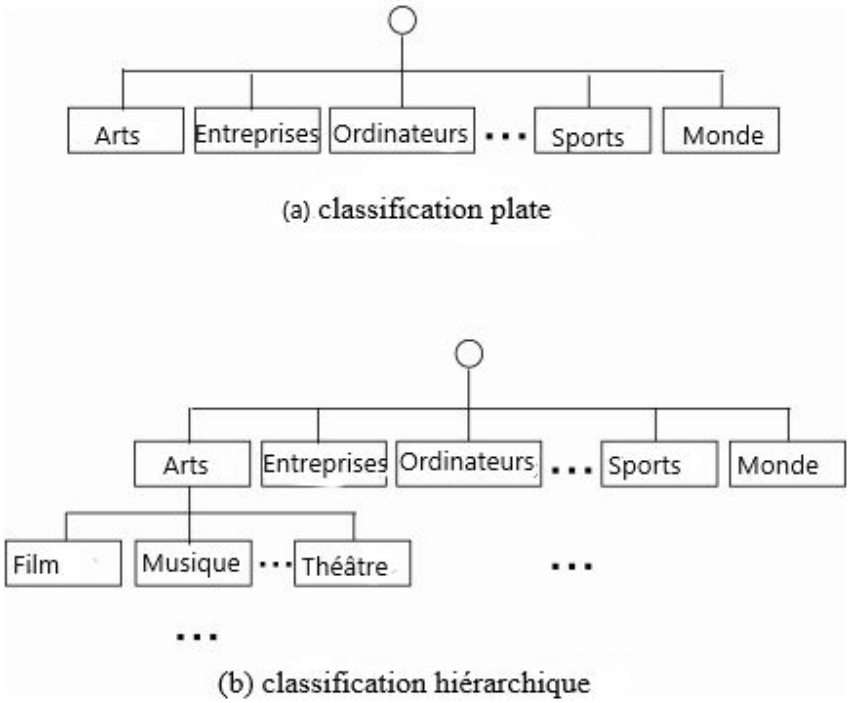

Figure 1.3 : Classification plate et classification hiérarchique [Xiaoguang Qi et all,2009]

5. Conclusion

Les approches de classification des pages Web dans la littérature sont explorées sous les trois catégories suivantes : (a) classification basée sur le texte, (b) basée sur l'image et (c) la classification combinée basée sur le texte et l'image.

Dans notre travail en s'intéressent à la classification des pages web basées sur le texte, cependant, la classification du texte se concentre principalement sur l'idée de compter les fréquences avec lesquelles les termes d'un lexique apparaissent dans le texte pour former un vecteur de caractéristiques et d'appliquer ces vecteurs de caractéristiques pour former un classificateur. Dans le chapitre qui suit nous présentant les différentes techniques d'extractions des mots clés.

Chapitre 02

Techniques d’extraction des mots clés

1. Introduction

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 ...).

L'extraction des informations ou bien des données à partir des sources web suscite un intérêt particulier ces dernières années. Alors il n'existe aucun standard, car les sources d'information Web restent très hétérogènes.

Dans ce chapitre nous discutant les différentes méthodes d'extraction automatique des termes clés.

2. Les méthodes d'extraction automatique de termes-clés

Les méthodes d'extraction de termes-clés se concentrent beaucoup plus sur les unités textuelles qui composent ces phrases. Un ensemble de termes -clés peut donc être perçu comme un résumé dont les points clés sont exprimés sans liaisons entre eux. Les termes candidats sont les unités textuelles sur lesquelles travaillent les systèmes d'extraction automatique de termes-clés. Ces derniers sont des mots ou des multi-mots, L'extraction de termes candidats est une étape préliminaire de l'extraction de termes-clés, que ce soit pour les méthodes non-supervisées ou supervisées [Wan et al., 2007].

2.1. Méthodes non-supervisées

Les méthodes non-supervisées d'extraction de termes-clés ont la particularité de s'abstraire du domaine et de la langue des documents à analyser. Cette abstraction est due au fait que les termes candidats sont analysés avec des règles simples déduites à partir de traits statistiques issus seulement du texte analysé, ou bien d'un corpus de référence non annoté.

De nombreuses approches sont proposées. Certaines se fondent uniquement sur des statistiques alors que d'autres les combinent avec des représentations plus complexes des documents. Ces représentations peuvent aller de groupes de mots sémantiquement similaires à des graphes dont les nœuds sont des unités textuelles (mots, expressions, phrases, etc.) liées par des relations de recommandation .

2.1.1. Approche statistiques

Dans la littérature, il ya plusieurs approches cherchent à définir ce qu'est un terme-clé en s'appuyant sur certains traits statistiques et en étudiant leur rapport avec la notion d'importance d'un terme candidat. Plus un terme candidat est jugé important vis-à-vis du document analysé, plus celui-ci est pertinent en tant que terme-clé.

a. *TF-IDF*

TF-IDF (ou Term Frequency – Inverse Document Frequency) [Paukkeri et Honkela ,2010].mesure le pouvoir discriminant d'un mot ou d'un groupe de mots dans un document donné. Essentiellement, cette technique mesure l'importance d'un certain terme dans un document par rapport aux autres documents de la même collection.

$$TF - IDF(terme) = TF(terme) \times \log(\frac{N}{DF(terme)})$$

Avec TF nombre d'occurrence du terme dans le document et DF nombre de documents dans lequel le terme est présent et N nombre total de documents.

Cette mesure est utilisée pour pondérer les termes-candidats : plus la valeur TF-IDF d'un terme-candidat est élevée, plus celui-ci est important dans le document analysé. En prenant compte de tous les documents dans le corpus, cette méthode présente généralement de meilleurs résultats [Ding et al. 2011].

2.1.2. Méthodes à base de graphe

Les méthodes qui vont suivre sont des méthodes à base de graphe noté G (N, A) où N est l'ensemble des noeuds et A l'ensemble de ses arcs sortants et entrants. Chaque sommet du graphe représente un terme-candidat et la constitution des arêtes est propre à chaque méthode.

a. *TextRank*

TextRank est basée sur le calcul du score d'importance des sommets en utilisant le principe de vote ou de recommandation entre deux sommets [Mihalcea *et al,* 2004]. TextRank utilise une représentation efficace d'un document, elle peut aussi être utilisée pour faire des résumés automatiques d'un document. Malgré cela, elle possède un inconvénient : au lieu d'ordonner des termes-candidats, elle n'ordonne que des mots. Cette méthode repose sur les étapes suivantes : *construction du graphe, calcul des scores des sommets et extractions des termes clés.*

- *Construction du graphe* : les termes-candidats initiaux sont des mots simples , ils sont utilisés comme sommets du graphe. Deux sommets sont reliés par une arête s'ils co-occurrent dans une fenêtre de N mots.
- *Calcul des scores des sommets* : Au départ, les scores de tous les sommets du graphe sont initialisés aléatoirement. Un algorithme de classement calcule les scores de chaque sommet à chaque itération et s'arrête lorsque le seuil donné est atteint.
- *Extraction des termes-clés* : C'est à partir des sommets les plus importants (selon leur score par ordre décroissant) que sont choisis les mots-clés. Les séquences des mots (des sommets importants) adjacents dans le document constituent les termes clés composés de plusieurs mots, les autres mots non adjacents dans les documents qui obtiennent les meilleurs scores sont également retenus comme mots-clés.

b. *SingleRank*

SingleRank est une modification de la méthode TextRank, la différence se trouve dans la pondération des arêtes du graphe de mots et dans l'extraction des mots-clés à partir des mots-candidats, c'est-à-dire le calcul des scores des termes [Wan et al, 2008]. Dans la majorité des cas, cette méthode fournit de meilleurs résultats que TextRank. L'inconvénient de cette méthode est qu'elle favorise les termes-candidats les plus longs (c'est-à-dire les termes formés par de nombreux mots) tout en faisant monter les candidats redondants dans le classement. Cette méthode, comme TextRank, repose sur trois étapes : *construction du graphe, calcul des scores des sommets et extractions des termes clés.*

- *Construction du graphe* : les terme-candidats peuvent être composés et chaque mot composant chaque terme-candidat est considéré comme sommet du graphe. Deux sommets sont reliés par une arête s'ils co-occurrent dans une fenêtre de N mots et le poids de cette arête est le nombre de cooccurrence de ces deux sommets.
- *Calcul des scores des sommets* : SingleRank utilise un algorithme de classement pour calculer les scores des sommets.
- *Extraction des termes-clés* : Ce sont les termes-candidats ayant les scores les plus importants qui sont retenus comme termes-clés. Il n'y a donc pas de génération de termes-clés comme cela est le cas dans TextRank.

c. *TopicRank*

TopicRank, fondée sur TextRank, est différente par rapport aux autres méthodes à base de graphe, parce qu'au lieu de faire une recherche des unités textuelles importantes du document, elle cherche ses sujets importants. Par rapport à TextRank et SingleRank, elle présente les avantages suivants : suppression des problèmes de redondance dans les termes-clés extraits, construction d'un graphe plus compacte, renforcement des poids des arêtes dans le graphe, amélioration de la qualité d'ordonnancement et suppression du paramètre de la fenêtre de co-occurrences. Cette méthode repose sur trois étapes : *identification des sujets, ordonnancement des sujets, sélection des mots-clés* [Boudin *et al,* 2013].

- *Construction du graphe* : ce sont les termes-candidats, composés de plusieurs mots ou non, qui représentent les sommets du graphe et tous les sommets sont reliés entre eux, nous avons un graphe complet.
- *Identification des sujets* : un sujet est une information spécifique (le plus souvent) ou générale transportée au minimum par une unité textuelle. Deux termes-candidats $C1$ et $C2$ sont groupés à partir de la similarité de Jaccard :

  $$sim(C_1, C_2) = \frac{\|C_1 \cap C_2\|}{\|C_1 \cup C_2\|}$$

  Avec : $C1 \cup C2$: nombre de mots commun à $C1$ et $C2$, $C1 \cap C2$: nombre de mots composant $C1$ et $C2$.

  Dès que la similarité entre toutes les paires de termes-candidats est connue, l'algorithme de classification ascendante hiérarchique est appliqué. Au début, chaque terme-candidat est considéré comme un groupe et puis les deux groupes présentant la plus forte similarité sont réunis en un seul. Ce regroupement est répété jusqu'à ce que le nombre (prédéfini) de groupes soit atteint. La similarité entre deux groupes est obtenue en calculant la similarité entre les termes-candidats composant chaque groupe.
- *Ordonnancement des sujets* : la pondération des arêtes est très importante durant cette étape. C'est la force du lien sémantique [Wan and Xiao, 2008] entre les nœuds du graphe (ou plutôt les sujets) qui est considérée comme poids d'une arête. Pour représenter cette force sémantique, la distance entre les termes-candidats des sujets est utilisée.
- *Sélection des termes-clés* : chaque sujet important ne va fournir qu'un seul terme clé. Pour le choix d'un terme-clé représentant le mieux un sujet, trois méthodes ont été proposées :

- *Première position* : le terme-candidat d'un sujet apparaissant le premier dans le document est sélectionné,
- *Fréquence* : le terme-candidat d'un sujet le plus fréquent dans le document analysé est sélectionné.

d. Kcore

Kcore [Rousseau *et al,* 2015] est aussi une méthode d'extraction de termes-clés à base de graphe. La construction de son graphe de mots est semblable à celle de TextRank ou SingleRank. Contrairement aux méthodes à base de graphe que nous avons présentées ci-dessus, elle n'utilise pas le principe de vote ou de recommandation pour calculer les scores d'importance des sommets mais utilise l'algorithme de Batagelj et Zaveršnik [Batagelj *et al,* 2011]. Le problème est qu'elle dépend de la fenêtre de co-occurrence de mots.

- *Construction du graphe* : ce sont les termes-candidats, constitués de plusieurs mots ou non, qui représentent les sommets du graphe et deux sommets sont reliés si les termes-candidats représentant ces sommets co-occurrent dans une fenêtre de N mots. Ces liens peuvent être pondérés par le nombre de co-occurrences des mots qu'ils relient dans le document, on parle de WKcore , sinon (dans le cas d'un graphe non pondéré) ils seront tous pondérés par 1, on parle de Kcore .

2.1.3. Approches par regroupement

Le but des approches par regroupement est de définir des groupes dont les unités textuelles partagent une ou plusieurs caractéristiques communes. Ainsi, lorsque des termes-clés sont extraits à partir de chaque groupe, cela permet de mieux couvrir le document analysé selon les caractéristiques utilisées.

- Dans la méthode de [Matsuo et Ishizuka ,2004], ce sont les termes (phrasèmes) qui sont regroupés. Parmi ceux-ci, seuls les plus fréquents sont concernés par le regroupement. Celui-ci s'effectue en fonction du lien sémantique entre les termes. Après le regroupement, la méthode consiste à comparer les termes candidats du document analysé avec les groupes de termes fréquents, en faisant l'hypothèse qu'un terme candidat qui co-occurre plus que selon toute probabilité avec les termes fréquents d'un ou plusieurs groupes est plus vraisemblablement un terme-clé.
- Dans l'algorithme Key Cluster, [Liu *et al* ,2009] utilisent aussi un regroupement sémantique, mais dans leur cas ils considèrent les mots du document analysé et ils

excluent les mots outils. Dans chaque groupe sémantique, le mot qui est le plus proche du centroïde est sélectionné comme mot de référence. L'ensemble des mots de référence est ensuite utilisé pour filtrer les termes candidats en ne considérant comme termes-clés que ceux qui contiennent au moins un mot de référence (tous les mots de référence devant être utilisés dans au moins un terme-clé).

2.2. Méthodes supervisées

Les méthodes supervisées sont des méthodes capables d'apprendre à réaliser une tâche particulière, soit ici l'extraction de termes-clés. L'apprentissage se fait grâce à un corpus dont les documents sont annotés en termes-clés. L'annotation permet d'extraire les exemples et les contres exemples dont les traits statistiques et/ou linguistiques servent à apprendre une classification binaire.

La classification binaire consiste à indiquer si un terme candidat est un terme-clé ou non. De nombreux algorithmes d'apprentissage sont utilisés dans divers domaines. Ils peuvent potentiellement s'adapter à n'importe quelle tâche, dont celle de l'extraction automatique de termes-clés. Les algorithmes utilisés pour celle-ci construisent des modèles probabilistes, des arbres de décision, des Séparateurs à Large Marge (SVM) ou encore des réseaux de neurones .

KEA [Witten *et al*, 1999] est une méthode qui utilise une classification naïve bayésienne pour attribuer un score de vraisemblance à chaque terme candidat, le but étant d'indiquer s'ils sont des termes-clés ou non, Il est important de noter que le score de vraisemblance pour chaque terme candidat permet aussi de les ordonner entre eux. . [Witten *et al* ,1999] utilisent trois distributions conditionnelles apprises à partir du corpus d'apprentissage. La première correspond à la probabilité pour que chaque terme candidat soit étiqueté *oui* (terme-clé) ou *non* (non terme-clé). Les deux autres correspondent à deux différents traits qui sont le poids TF-IDF du terme candidat et sa première position dans le document.

L'un des avantages de la classification naïve bayésienne est que chaque distribution est supposée indépendante. L'ajout de nouveaux traits dans la méthode KEA est donc très aisé. Parmi les variantes de KEA proposées, [Frank *et a,* 1999] ajoutent un troisième trait : le nombre de fois que le terme candidat est un terme-clé dans le corpus d'apprentissage. L'ajout de ce trait permet d'améliorer les performances de la version originale de KEA, mais uniquement lorsque la quantité de données d'apprentissage est très importante. Une autre amélioration de KEA, proposée par [Turney,2003], tente d'augmenter la cohérence entre les termes candidats les mieux

classés. Pour ce faire, une première étape de classification est effectuée avec la méthode originale. Cette première étape permet d'obtenir un premier classement des termes candidats selon leur score de vraisemblance. Ensuite, de nouveaux traits sont ajoutés et une nouvelle étape de classification est lancée. Les nouveaux traits ont pour but d'augmenter le score de vraisemblance des termes candidats ayant un fort lien sémantique avec certains des termes les mieux classés après la première étape. Enfin, [Nguyen et Kan, 2007] proposent l'ajout des informations concernant la structure des documents. En effet, certaines sections telles que l'introduction et la conclusion dans les articles scientifiques sont plus susceptibles de contenir des termes-clés qu'une section présentant des résultats expérimentaux, par exemple.

En même temps que KEA [Witten *et al*, 1999], [Turney,1999] met au point l'algorithme génétique GenEx. GenEx est constitué de deux composants. Le premier composant, le géniteur, sert à apprendre des paramètres lors de la phase d'apprentissage. Ces paramètres sont utilisés par le second composant, l'extracteur, pour donner un score d'importance à chaque terme candidat.

Plus les paramètres sont optimaux, meilleure est la classification des termes. Pour ce faire, les paramètres sont représentés sous la forme de bits qui constituent une population d'individus que le géniteur fait évoluer jusqu'à obtenir un état stable correspondant aux paramètres optimaux.

Dans son article présentant GenEx, [Turney, 1999] discute une autre méthode pour l'extraction de termes-clés. Cette méthode utilise de nombreux traits qui servent à entraîner 50 arbres de décision (technique de *Random Forest*). Dans un arbre de décision, chaque branche représente un test sur l'un des traits d'un terme candidat. Les tests permettent un routage du terme candidat vers la feuille de l'arbre qui détermine sa classe. Grâce à la technique de *Random Forest*, soit l'usage de plusieurs arbres entraînés sur un échantillon différent du corpus d'apprentissage, l'extraction automatique de termes-clés est réduite à un vote de chaque arbre pour chaque terme candidat. Cela permet un classement des termes candidats en fonction de leur nombre de votes positifs. Les termes-clés extraits correspondent aux termes candidats les mieux classés.

La même année que les travaux de [Hulth, 2003] sur le bien fondé d'utiliser des traits linguistiques pour l'extraction automatique de termes-clés, [Sujian *et ,* 2003] proposent une méthode utilisant un modèle d'entropie maximale dont l'un des traits repose sur les parties du discours des mots qui composent les termes candidats. Un modèle de maximum d'entropie consiste à trouver parmi plusieurs distributions, une pour chaque trait, laquelle a la plus forte

entropie. La distribution ayant la plus forte entropie est par définition celle qui contient le moins d'informations, ce qui la rend de ce fait moins arbitraire pour l'extraction des termes-clés.

Les Séparateurs à Large Marge sont aussi des classifieurs utilisés par les méthodes d'extraction automatique de termes-clés. Ils exploitent divers traits afin de projeter des exemples et des contres-exemples sur un plan, puis ils cherchent l'hyperplan qui les sépare. Cet hyperplan sert ensuite dans l'analyse de nouvelles données. Dans le contexte de l'extraction de termes-clés, les exemples sont les termes-clés et les contres-exemples sont les termes candidats qui ne sont pas des termes-clés. Ce mode de fonctionnement des SVM est utilisé par [Zhang *et al,* 2006], mais un autre type de SVM est plus largement utilisé dans les méthodes supervisées d'extraction de termes-clés. Il s'agit de SVM qui utilisent de multiples marges représentant des rangs. Ces classifieurs permettent donc d'ordonnancer les termes-clés lors de leur extraction [Herbrich *et al* , 1999] [Joachims, 2006] [Jiang *et al*, 2009]. La méthode KeyWE de [Eichler et Neumann ,2010] utilise ce type de SVM avec le trait TF-IDF ainsi qu'un trait booléen ayant la valeur vraie si le terme candidat apparaît dans un titre d'un article Wikipedia (un terme candidat apparaissant dans le titre d'un article de Wikipedia a une plus forte probabilité d'être un terme-clé). L'ordonnancement des termes candidats par le SVM permet ensuite de contrôler le nombre de termes-clés à extraire (choix des *k* termes candidats les mieux classés).

Tout comme [Turney ,1999], [Ercan et Cicekli ,2007] utilisent eux aussi une forêt d'arbres dans leur méthode d'extraction de termes-clés. Ils utilisent des traits classiques et leur contribution se situe au niveau de l'utilisation d'un trait calculé à partir de chaînes lexicales. Une chaîne lexicale lie les mots d'un document selon certaines relations telles que la synonymie, l'hyponymie ou la méronymie.

Ces relations permettent de calculer un score qui sert de trait. Cette approche est intéressante, mais du fait de limitations des chaînes lexicales actuellement disponibles elle présente l'inconvénient de ne retourner que des mots (aucun multi-mot). Cependant, l'usage d'une forêt d'arbre permet un classement des mots à partir de leur nombre de votes positifs. Il est donc envisageable de déduire les termes-clés à partir de la liste ordonnée et pondérée des mots clés (voir les méthodes non-supervisées à bases de graphe – section 2.1.2).

Une autre méthode pour l'extraction automatique de termes-clés consiste à utiliser un perceptron multi-couches [Sarkar *et al*, 2010]. Un perceptron multi-couches est un réseau de neurones constitué d'au moins trois couches, chaque couche étant composée de neurones. Dans les deux

couches extrêmes les neurones représentent respectivement les entrées et les sorties. Les couches centrales sont des couches cachées qui permettent d'acheminer les valeurs des entrées vers les sorties, où de nouvelles valeurs sont obtenues grâce à la pondération des transitions d'un neurone d'une couche vers un neurone de la couche suivante. Les entrées correspondent aux traits d'un terme candidat (ici TF-IDF, la position, la taille, etc.) et les sorties représentent les classes qu'il peut prendre (terme-clé ou non terme-clé). La valeur obtenue pour chaque sortie (classe) permet d'obtenir une probabilité pour que le terme candidat analysé soit un terme-clé ou non.

Dans leur méthode, [Sarkar *et al,* 2010] utilisent cette probabilité pour ordonner les termes candidats afin de mieux contrôler le nombre de termes-clés à extraire.

Dans leurs travaux, [Liu *et al,* 2011] proposent une méthode d'extraction de termes-clés basée sur un modèle génératif. Leur méthode est très différente de celle de [Witten *et al,* 1999] puisqu'ils décident d'utiliser une approche de traduction automatique. L'usage original de cette approche est justifié par le fait qu'un ensemble de termes-clés doit décrire de manière synthétique le document. Leur hypothèse est donc qu'un ensemble de termes-clés est une traduction d'un document dans un autre langage. Le modèle est appris à partir de paires de traductions dont l'un des termes est issu des titres ou des résumés des documents du corpus d'apprentissage et dont l'autre terme est issu des corps de ces mêmes documents. Les titres et les résumés sont utilisés comme langage synthétique et les corps des documents comme le langage naturel de ceux-ci.

3. Conclusion

L'extraction automatique de termes-clés est une tâche importante qui permet la valorisation d'un document (représentation synthétique, mise en évidence des points clés dans le document, etc.) et qui facilite l'accès aux documents pertinents pour une requête utilisateur (indexation pour la recherche d'information).

Les méthodes existantes pour la tâche d'extraction automatique de termes-clés sont soit supervisées, soit non-supervisées.

Dans le chapitre qui suit nous présentons en détaille notre contribution dans le domaine de classification des pages web.

Chapitre 03

Description et conception de la méthode proposée

1. Introduction

Avec l'augmentation du nombre d'internautes, la croissance des sites Web est proportionnelle. En conséquence, le classement des pages Web est devenu un énorme sujet de recherche ces dernières années. Cela a fait une demande toujours croissante pour des techniques de classification automatisées avec une précision de classification élevée. Pour catégoriser et manipuler automatiquement les pages Web, les systèmes actuels utilisent un contenu de page visuel, qui comprend le contenu affiché. Cependant, jusqu'à présent, peu de travaux ont été réalisés sur l'utilisation de contenu textuel et de code HTML.

Dans ce chapitre , nous proposons une méthode de classification de pages Web, basée sur leur contenu textuel. Les pages Web présentent en général des informations de différentes classes variées en fonction de leurs sujets spécifique. Cette méthode est basée sur une technique non supervisés statistique (TF-IDF) d'extraction des mots clé combinés avec une approche supervisée d'apprentissage automatique à savoir les réseaux de neurones récurrents.

2. Problématique et objectifs

De nos jours, les besoins de classification automatique des pages web en raison de l'augmentation constante du volume d'informations accessibles électroniquement, la conception et la mise en œuvre d'outils efficaces, permettant notamment à l'utilisateur de n'avoir accès qu'à l'information qu'il juge pertinente, devient une nécessité absolue. Comme la plupart de ces outils sont destinés à être utilisés dans un cadre professionnel, les exigences de fiabilité et de convivialité sont très importantes ; les problèmes à résoudre pour satisfaire ces exigences sont nombreux et difficiles.

Notre objectif est de proposer une méthode de classification de page web fondé sur l'apprentissage automatique pour la classification et la technique TF-IDF pour l'extraction des mots clés des pages.

3. Méthode proposée

L'internet se développe à un rythme exponentiel et peut couvrir à peu près toutes les données requises. Néanmoins, l'immense quantité de pages Web rend plus difficile la découverte efficace des données cibles par un utilisateur. Par conséquent, une méthode efficace pour classer cette énorme quantité de données est essentielle si les pages Web doivent être exploitées à leur plein

potentiel. C'est pour tous ces raisons que nous avons développé une méthode de classification automatique de page web basé sur l'extraction des mots clé.

3.1. Aperçu du fonctionnement de la méthode proposée

La figure 3.1 illustre l'architecture générale de notre système qui comprend deux composants essentiels qui sont :

1-Partie extraction : qui consiste à extraire les mots clé à partir d'une page web.

2-Partie classification : dans cette partie en a utilisé une approche supervisée d'apprentissage automatique à savoir les réseaux de neurones récurrents.

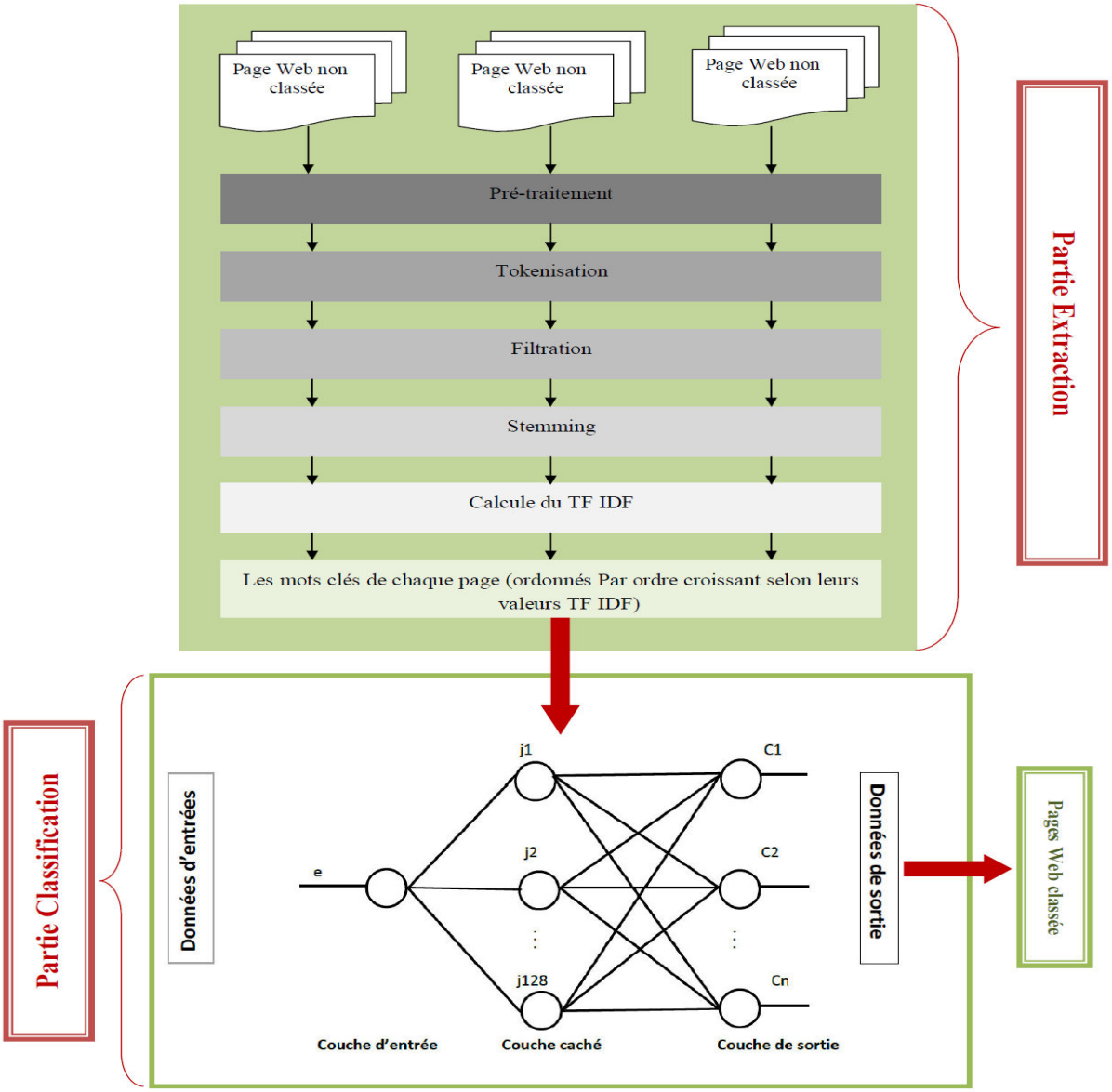

Figure 3.1 : L'architecture proposée de notre méthode

3.2. Fonctionnement de la méthode proposée

Dans cette section, nous présentons comment notre méthode fonctionne et répond aux objectifs fixé au début de ce travail. Ceci est fait à travers un digramme de séquence illustré dans la figure 3.2, et comme indiqué ci-dessus, nous utilisons une technique d'extraction de mot clé TF-IDF et une approche d'apprentissage supervisé qui est Les réseaux de neurones récurrents.

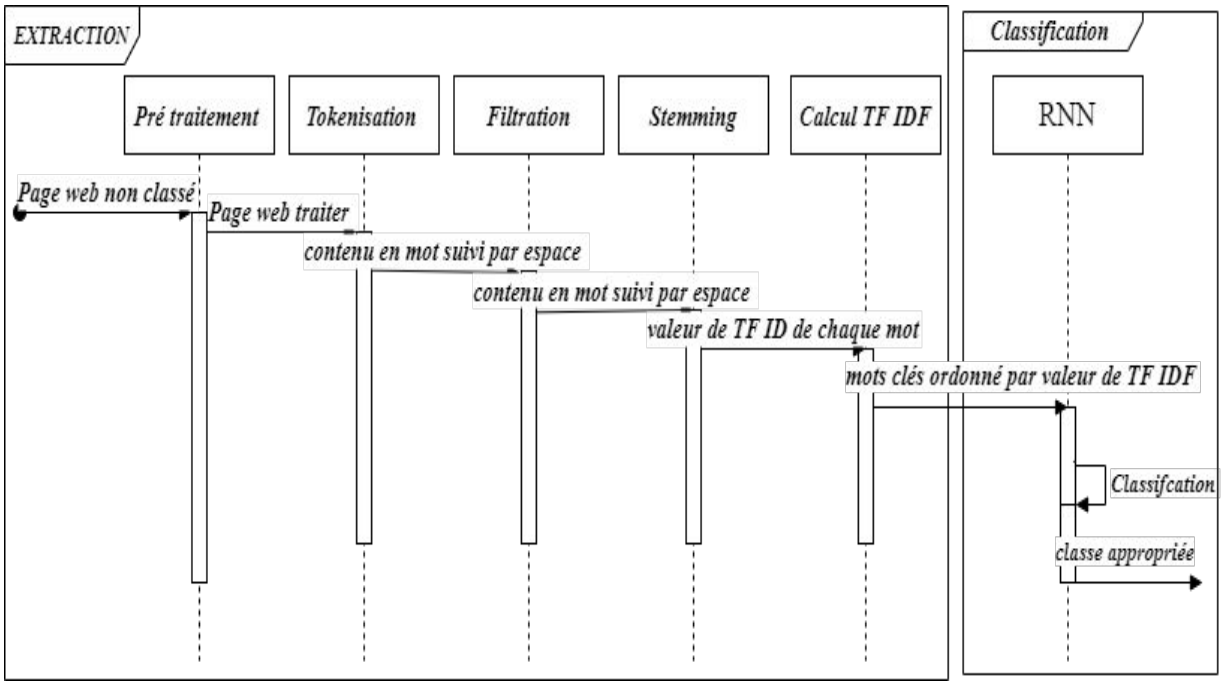

Figure 3.2 : Diagramme de séquences de la méthode proposée

3.2.1. Partie Extraction

Dans la littérature il existe plusieurs méthodes d'extraction automatique de mot clés que nous avons citées dans le chapitre 2, dans notre travail en va utiliser une méthode non supervisés statistique qui est TF-IDF, puisqu'elle fournit de meilleurs résultats que les autres méthodes.]T.Derdra Amel et all,2011].

a- Méthode TF-IDF

TF-IDF (Terme Fréquence - Fréquence Inverse du Document) : est une mesure statistique qui évalue la pertinence d'un mot pour un document dans une collection de documents

- **TF (Terme Fréquence) :** La fréquence d'un terme est simplement le nombre d'occurrences de ce terme dans le document considéré ;

- **IDF (Fréquence Inverse du Document) :** La fréquence inverse de document est une mesure de l'importance du terme dans l'ensemble du corpus ;

- **TF*IDF(Terme Fréquence - Fréquence Inverse du Document):**

  Le poids d'un terme *T* dans un document *D* est calculé comme suit :

$$\textbf{TF-IDF } (T_i, D_j) = \textbf{TF } (T_i, D_j) * \log (N/ DF(T))$$

Avec :

- **TF** (T_i, D_j): la fréquence du terme dans le document ;
- **N** : le nombre total de documents de la base documentaire ;
- **DF**(T_i) : le nombre de documents contenant le terme.

b- Les étapes de TF IDF

Le prétraitement : 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]T.Derdra Amel et all,2011].

L'entrer est une page web sou forme de fichier.html, donc il faut un pré traitement qui est la suppression de tous les balises HTML et le code CSS et java script ensuite la suppression des chaines de caractère des caractères spéciaux (exemple @h12-D*65), puis le transfert de tous le contenus en minuscule (puisque dans python INFORMATIQUE =/= informatique), finalement la suppression des ponctuations (:,;?!.), ensuite la suppression des chiffres.(voir figure 3.3)

Tokenisation : découpé le contenu en mot suivi par espace (mot _mot_...) Manning, c.det all, 2008].

La filtration : les suppressions des mots vide sémantiquement. 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 , 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.

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. Donc le résultat de cette étape sera sous forme de stemmer.

Le calcule de TF IDF : le calcule de TF de chaque mot puis DF ensuit IDF qui est l'inverse de DF selon leurs formules mathématiques finalement le produit entre TF et IDF qui sera la valeur de TF ID de chaque mot.

L'affichage des résultats : les résultats des mots clés seront ordonné par ordre croissant selon leurs valeurs de TF IDF (voir figure 3.3).

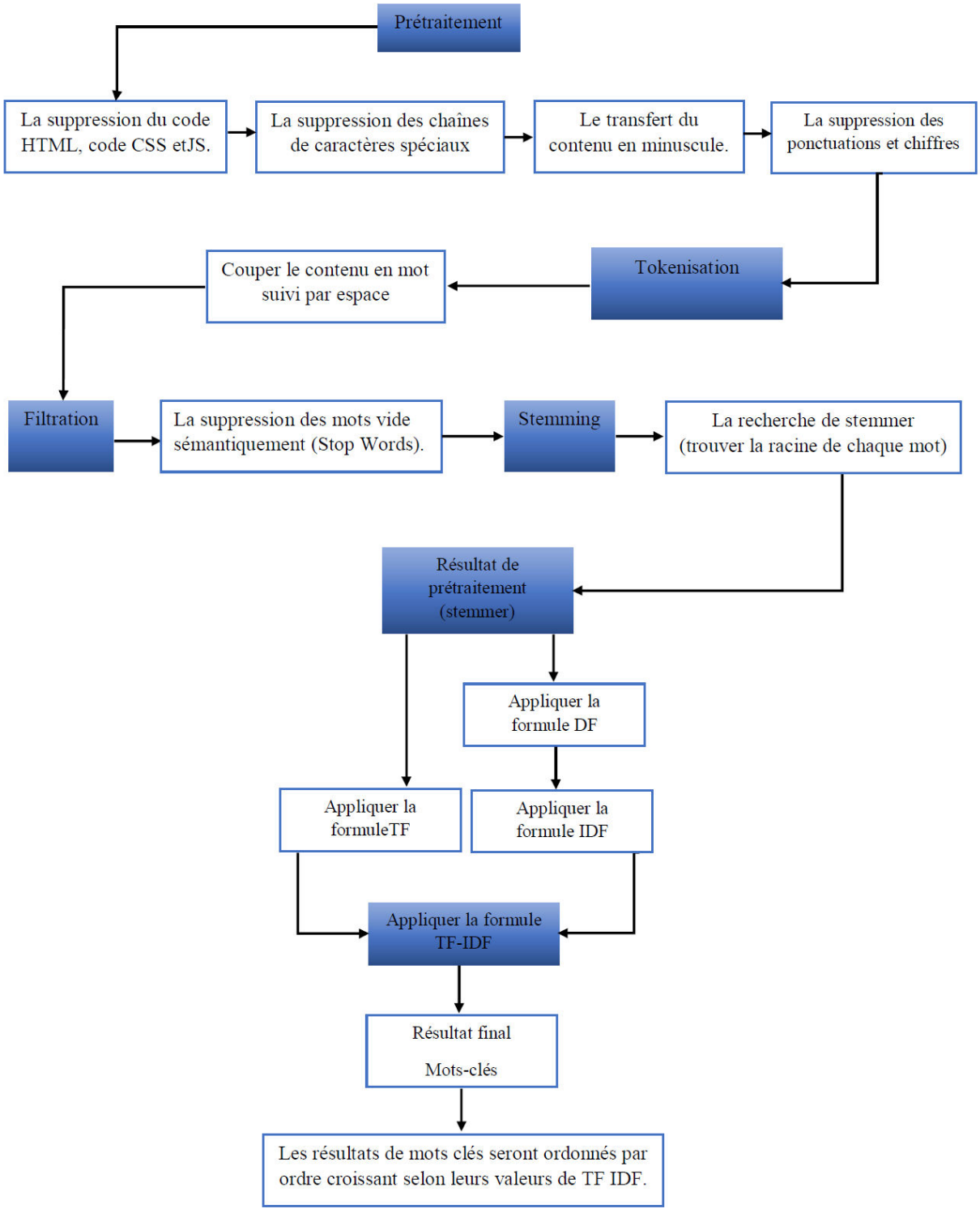

Figure 3.3 : Etape de TF-IDF

3.2.2. Partie classification

Les réseaux de neurones sont généralement optimisés par des méthodes d'apprentissage de type statistique grâce à leur capacité de paradigmes permettant de générer de vastes espaces fonctionnels, souples et partiellement structurés .Ils appartient d'autre part à la famille des méthodes de l' intelligence artificielle qu'ils enrichissent en permettant de prendre des décisions s'appuyant davantage sur la perception que sur le raisonnement logique forme. réseau de neurones (ou *Artificial Neural Network* en anglais) est un modèle de calcul dont la conception est très schématiquement inspiré du fonctionnement de vrais classification et de généralisation**,** tels que la classification automatique de codes postaux ou la prise de décision concernant un achat boursier en fonction de l'évolution des cours [S. Prabhu et all,2007]

Un réseau de neurone est en général composé d'une succession de couches dont chacune prend ses entrées sur les sorties de la précédente. Chaque couche (i) est composée de Ni neurones, prenant leurs entrées sur les Ni-1 neurones de la couche précédente. À chaque synapse est associée un poids synaptique, de sorte que les Ni-1 sont multipliés par ce poids, puis additionnés par les neurones de niveau i, ce qui est équivalent à multiplier le vecteur d'entrée par une matrice de transformation. Mettre l'une derrière l'autre, les différentes couches d'un réseau de neurones reviendrait à mettre en cascade plusieurs matrices de transformation et pourrait se ramener à une seule matrice, produit des autres, s'il n'y avait à chaque couche, la fonction de sortie qui introduit un non linéarité à chaque étape. Ceci montre l'importance du choix judicieux d'une bonne fonction de sortie : un réseau de neurones dont les sorties seraient linéaires n'aurait aucun intérêt.

Il existe plusieurs type de réseau de neurone parmi lesquels, Les réseaux de neurones récurrents (bouclés) qui sont utilisé dans notre travail.

a- Les réseaux de neurones récurrents (bouclés)

Un réseau bouclé (récurrent), régi par une ou plusieurs équations différentielles, résulte de la composition des fonctions réalisées par chacun des neurones et des retards associés à chacune des connexions [SRINIVASA.V et al, 1996].

Qu'elle soit **détaillée** ou **simplifiée**, la représentation d'un réseau récurrent **n'est pas aisée**, car il est difficile de faire apparaître **la dimension temporelle** sur le schéma. C'est notamment le cas pour les connexions récurrentes, qui utilisent l'information du temps précédent. Pour solutionner

ce problème, on utilise souvent une représentation du réseau "**déplié dans le temps**", afin de faire apparaître explicitement celui-ci. La figure suivante montre un exemple de réseau déplié :

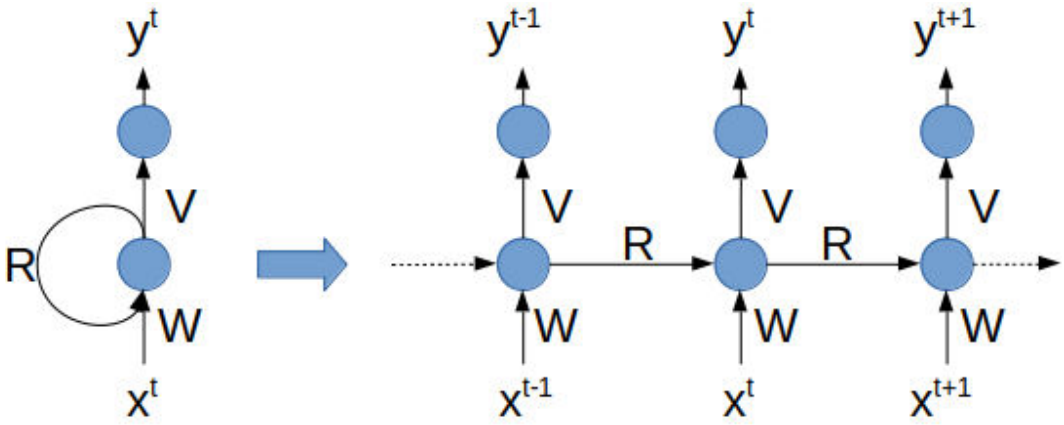

Figure 3.4 : RNN : version récurrente et version dépliée

Cette version dépliée dans le temps fait désormais apparaître clairement les variables d'entrée au cours du temps : x_{t-1} , x_t , x_{t+1}, etc. (idem pour la sortie), et l'impact des sorties précédentes sur la sortie courante du réseau. Dans sa version dépliée, les matrices WW, RR et VV sont dupliquées et apparaissent ainsi sur le schéma autant de fois que le nombre de dépliements du réseau dans le temps .

En faisant explicitement apparaître la dimension temporelle, la version dépliée suggère trois utilisations possibles d'un réseau récurrent : **étiquetage** de séquences, **classification** de séquences ou **génération** de séquences. Dans notre travail, on s'intéresse à la classification des séquences

- **Classification de séquence**

Dans ce mode de fonctionnement, le réseau parcourt la séquence d'entrée de taille T selon le sens de lecture, et ne produit une sortie qu'une fois la séquence d'entrée terminée, comme illustré sur la figure suivante :

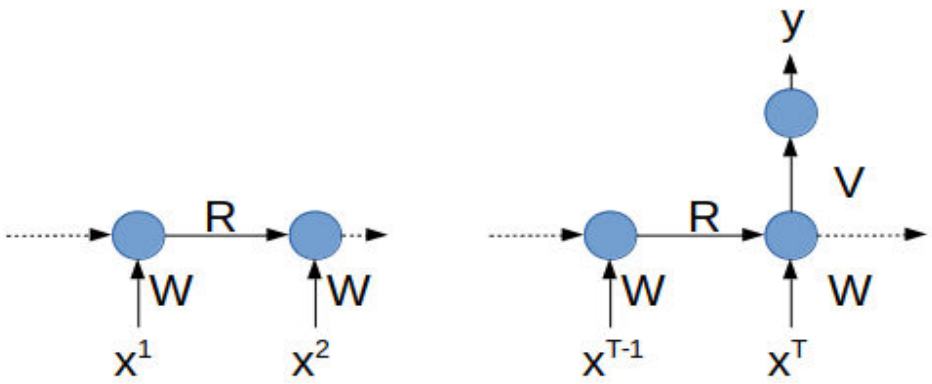

Figure 3.5 : Classification de séquences : le réseau "lit" la séquence dans son intégralité, et produit sa sortie au dernier pas de temps.

Dans ce cas, la sortie n'est pas une séquence, mais seulement une étiquette. Cette approche fonctionne aussi en **régression** ; dans ce cas, la sortie est une valeur ou un vecteur de valeurs.

b- Fonctionnement de notre réseau récurrent

Nous allons construire et former un RNN de base au pour classer les pages web selon les mots clés. Ce didacticiel, ainsi que les deux suivants, montrent comment prétraiter les données pour la modélisation PNL « à partir de zéro », en particulier en n'utilisant pas la plupart des fonctions pratiques de torchtext, afin que nous puissions voir comment le prétraitement pour la modélisation PNL fonctionne à bas niveau.

Un RNN au niveau des caractères lit les mots comme une série de caractères - produisant une prédiction et un "état caché" à chaque étape, alimentant son état caché précédent dans chaque étape suivante. Nous considérons que la prédiction finale est la sortie, c'est-à-dire à quelle classe appartient le mot.

Plus précisément, nous allons nous entraîner sur quelques milliers de noms de 10 domaines et prédire de quel domaine provient un nom en fonction de l'orthographe

Préparation des données

Le répertoire data/names comprend 10 fichiers texte nommés « [Domaie].txt ». Chaque fichier contient un tas de noms, un nom par ligne,

On aboutira à un dictionnaire de listes de noms par domaine, {domaine: [names ...]}. Les variables génériques « catégorie » et « ligne » (pour les domaine et le nom dans notre cas) sont utilisées pour une extensibilité ultérieure.

Nous avons maintenant category_lines, un dictionnaire mappant chaque catégorie (domaine) à une liste de lignes (noms). Nous avons également gardé une trace de all_catégories (juste une liste de domaines) et n_categories pour référence ultérieure.

Transformer les noms en tenseurs

Maintenant que nous avons organisé tous les noms, nous devons les transformer en tenseurs pour pouvoir les utiliser.

Pour représenter une seule lettre, nous utilisons un « vecteur unique » de taille <1 x n_lettres>. Un vecteur one-hot est rempli de 0 à l'exception d'un 1 à l'index de la lettre actuelle, par ex. "b" = <0 1 0 0 0 ...>.

Pour faire un mot, nous en joignons un tas dans une matrice 2D <line_length x 1 x n_letters>.

Cette dimension supplémentaire est due au fait que PyTorch suppose que tout est en lots - nous utilisons simplement une taille de lot de 1 ici.

Établir des tenseurs

Le RNN prend un tenseur d'entrée de longueur fixe et un tenseur de sortie de longueur fixe (vous pouvez les considérer comme des tableaux ici).

La longueur de sortie correspond au nombre de catégories qu'un titre pourrait être.

Pour représenter un mot ou un morceau de texte, nous combinons les tenseurs dans une matrice 2D. Cependant, plus tard, nous alimenterons les tenseurs de lettres un par un dans le réseau.

Création du réseau

Notre RNN aura la structure suivante :(voir figure 3.6)

Nous pouvons diviser cette structure en trois couches : la couche d'entrée, la couche de calcul et la couche de sortie.

La couche d'entrée crée une entrée pour la couche de calcul. Ici, nous voulons que chaque exécution prenne un mot (en tant que vecteur unique) et un état caché. L'état caché est un tenseur de taille <1 x n>. Vous pouvez jouer avec n, mais nous le définirons sur 128. Comme les réseaux de neurones prennent en entrée un tenseur, nous concaténons simplement le vecteur one-hot et l'état caché pour former le tenseur combiné.

La couche de calcul exécute deux transformations linéaires (Entrée-Sortie & Entrée-Cachée) sur le tenseur combiné pour créer la couche de sortie, constituée d'une prédiction et de l'état caché suivant. Nous exécutons un softmax sur la prédiction pour écraser les valeurs de Entrée-Sortie entre [0,1] pour obtenir les probabilités multi-classes. Par exemple, il peut afficher 0,01 indiquant

que le réseau pense qu'il y a une probabilité de 1 % que le mot appartient à la classe Informatique. Cette sortie est directement comparable à notre tenseur cible, qui peut ressembler à [1, 0] . Cela nous permet de calculer la perte pour chaque prédiction.

La couche de sortie contient notre prédiction et le prochain état caché.

Dans notre code, nous héritons du nn.Module qui est la classe de base pour tous les réseaux de neurones dans PyTorch. Nous initialisons une instance en spécifiant les tailles d'entrée/sortie/état caché qui aideront à créer les couches linéaires et softmax. La fonction forward() exécute une étape dans le réseau de neurones.

Les étapes ci-dessous montre comment nous gérons une étape du réseau :

- Initialisation du RNN
- Création du tenseur d'entrée
- Création de l'état caché initial
- Passer ces deux tenseurs dans le RNN

Entrainement

Dans cette section, nous allons :

Convertissez les paires d'entraînement en tenseurs et alimentez-les dans le RNN

Former le RNN à l'aide d'un optimiseur, d'une fonction de perte et d'un taux d'apprentissage. Nous avons notre réseau de neurones, nos tenseurs d'entrée et nos tenseurs cibles pour comparer les prédictions du réseau

Pour mettre à jour les poids du réseau à chaque étape (c'est-à-dire la rétropropagation), nous aurons besoin d'une fonction de perte pour calculer les gradients en fonction de la différence entre une prédiction et la vraie valeur. Ensuite, nous devrons spécifier un optimiseur et un taux d'apprentissage pour réellement mettre à jour le réseau. Ici, nous avons choisi :

Fonction de perte : NLLLoss(…) (perte de vraisemblance négative du log)

Optimiseur : torch.optim.SGD(…) (descente de gradient stochastique)

Taux d'apprentissage : 0,0002

Une étape d'apprentissage utilise un mot en entrée et son étiquette correspondante. Chaque étape :

- Définissez le modèle en mode d'entraînement
- Créez le tenseur d'entrée à partir du titre et le tenseur cible à partir de l'étiquette
- Créer un état caché initial (plein de zéros)
- Introduisez le mot à travers le réseau u, en passant les états cachés à l'exécution suivante

- Calculer la perte en la comparant à la vraie valeur à l'aide de la fonction de perte
- Mettre à jour les paramètres du réseau avec l'optimiseur
- Renvoie la sortie et la perte pour montrer comment le réseau apprend

Évaluer le modèle

Dans cette section, nous allons :

Calculez l'exactitude, le rappel et la précision de la prédiction du modèle sur l'ensemble de test. Ici, nous calculons les trois mesures en faisant prédire au modèle pour le mot si la classe correspondant est fiable ou non. La fonction d'évaluation () fonctionne de manière similaire à l'étape d'apprentissage, en ce sens qu'elle exécute le mot dans le RNN et prend la sortie. Nous utilisons la fonction category_from_output() définie dans notre code (voir annexe) pour prendre la catégorie avec la probabilité la plus élevée et l'utiliser comme estimation du modèle.

Dans mon cas, nous avons obtenus les résultats suivants :

-Rappel : 0,706

-Précision : 0,697

Remarque :

La fonction **Softmax :** transforme les logits (sortie numérique de la dernière couche linéaire d'un réseau de neurones de classification multi-classes) en probabilités en prenant les exposants de chaque sortie, puis en normalisant chaque nombre par la somme de ces exposants afin que le vecteur de sortie entier s'ajoute à un - toutes les probabilités doivent totaliser un.

C'est-à-dire que Softmax attribue des probabilités décimales à chaque classe d'un problème à plusieurs classes. ... La somme de ces probabilités décimales doit être égale à 1.

Nous pouvons utiliser la fonction argmax pour obtenir l'indice les indices de la valeur maximale de tous les éléments du tenseur d'entrée.

La fonction **forward** calcule les tenseurs de sortie à partir des tenseurs d'entrée. La fonction forward reçoit le gradient des Tenseurs de sortie par rapport à une valeur scalaire et calcule le gradient des Tenseurs d'entrée par rapport à cette même valeur scalaire.

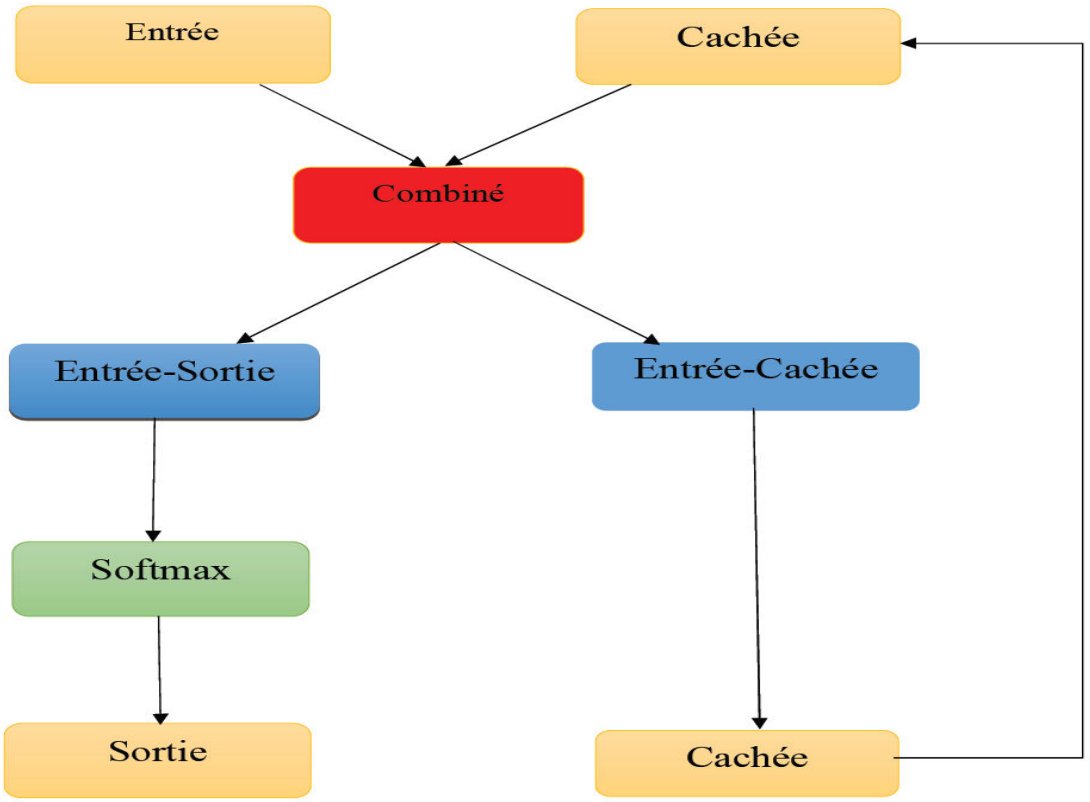

Figure 3.6 : Fonctionnement de RNN

4. Conclusion

Dans ce chapitre, nous avons présenté la description et la conception de ma méthode proposée ; cette dernière est basée sur une technique non supervisés statistique (TF-IDF) d'extraction des mots clé combinés avec une approche supervisée d'apprentissage automatique à savoir les réseaux de neurones. Le prochain chapitre sera consacré à la partie implémentation et évaluation de notre méthode.

Chapitre 04

Mise en œuvre de la méthode proposée

1. Introduction

Dans le chapitre précédent de ce mémoire, nous avons proposé notre méthode basée sur la classification des pages web à l'aide d'une technique d'extraction des mots clés.

Le présent chapitre est consacré à la réalisation et la concrétisation de notre modèle proposé.Dans un premier temps, nous présentons l'environnement de notre travail, on va donner le langage de programmation ainsi que l'environnement matériel et logiciel. Ensuite, l'implémentation de notre modèle et nous terminerons ce chapitre par une conclusion.

2. La Configuration du Matériel Utilisé

Nous avons réalisé ce travail sur une machine avec un processeur Intel® Core™ i7-8700k CPU @ 3.70GHZ (12 CPUs), 3.7GHZ doté d'une capacitémémoire de 32768 MB, sous Windows 10de 64 bits avec une carte graphique NVIDIA avec processeur GeForce GTX 1080 et un convertisseur Integrated RAMDAC et une mémoire 24425 MB.

3. Langage et environnement de développement

3.1. Langage de programmation python

Avec la complexification du web et l'accumulation des données, Python a pris une placemajeure dans le développement informatique car il peut être utilisé dans des situationsvariées, dans tous les secteurs d'activité.

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi , développer du code avec Python est plus rapide qu'avec d'autres langages.

Il reste aussi accessible pour les débutants, à condition de lui consacrer un peu de temps pour la prise en main. De nombreux tutoriels sont d'ailleurs disponibles pour l'étudier sur des sites Internet spécialisés ou sur des comptes You Tube. Sur les forums d'informatique, il est toujours possible de trouver des réponses à ses questions, puisque beaucoup de professionnels l'utilisent.

Les principales utilisations de Python par les développeurs sont :

- La programmation d'applications.
- La création de services web.

- La génération de code.
- La métaprogrammation.

On différencie deux versions : Python 2 et Python 3. Python 2, l'ancienne version propose des mises à jour jusqu'en 2020. Python 3 est la version actuelle. Son interpréteur est plus efficace, ainsi que son contrôle de concurrence. Dans ce travail, nous utilisons la version Python 3.

3.2. Navigateur utilisé : Anaconda

Qu'est-ce qu'Anaconda Navigator ?

Anaconda Navigator est une interface utilisateur graphique (GUI) de bureau inclus dans la distribution Anaconda® qui vous permet de lancer des applications et de gérer facilement les packages, les environnements et les canaux conda sans utiliser de commandes de ligne de commande. Navigator peut rechercher des packages sur Anaconda.org ou dans un référentiel Anaconda local. Il est disponible pour Windows, MacOs et Linux (Voir figure 4.1).

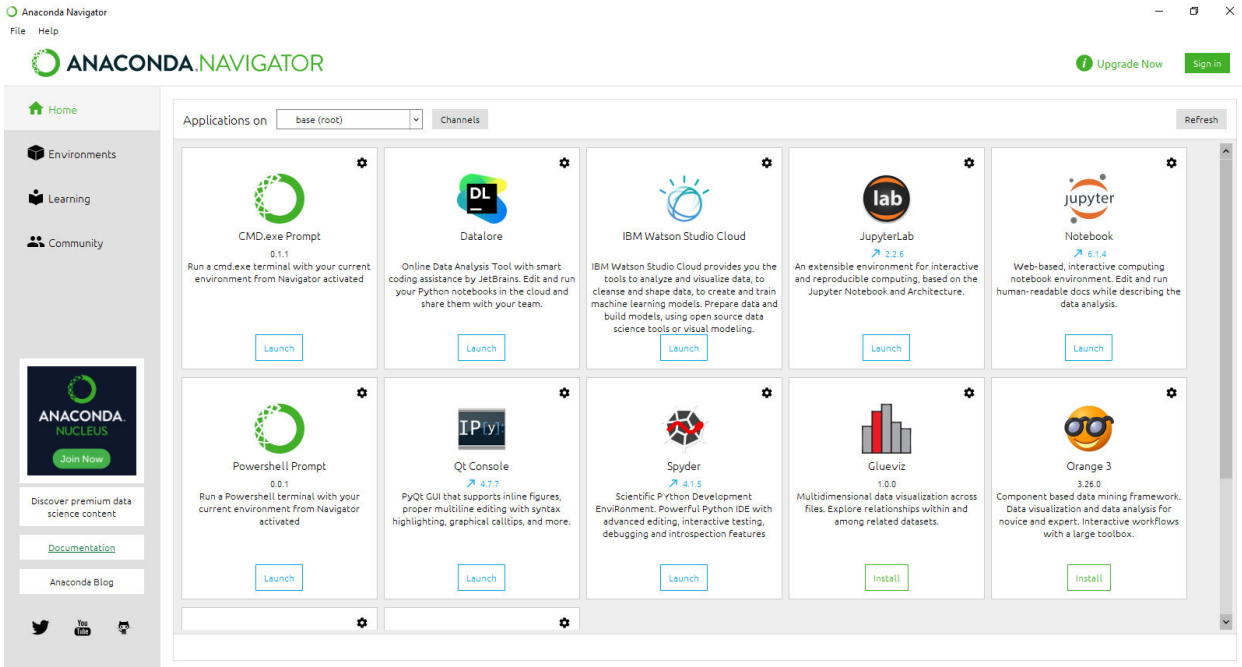

Figure 4.1 : Le navigateur Anaconda

Pourquoi utiliser Navigateur ?

- Pour fonctionner, de nombreux packages scientifiques dépendent de versions spécifiques d'autres packages. Les scientifiques des données utilisent souvent plusieurs versions de

nombreux packages et utilisent plusieurs environnements pour séparer ces différentes versions.

- Le programme en ligne de commande conda est à la fois un gestionnaire de paquets et un gestionnaire d'environnement. Cela aide les data scientistes à s'assurer que chaque version de chaque package possède toutes les dépendances dont elle a besoin et fonctionne correctement.
- Navigator est un moyen simple, pointer-cliquer, de travailler avec des packages et des environnements sans avoir besoin de taper des commandes conda dans une fenêtre de terminal. Vous pouvez l'utiliser pour trouver les packages que vous souhaitez, les installer dans un environnement, exécuter les packages et les mettre à jour, le tout dans Navigator.

À quels environnements puis-je accéder à l'aide de Navigator ?

Les environnements suivants sont disponibles par défaut dans Navigateur :

- JupyterLab.
- Jupyter Notebook.
- Spyder.
- PyCharm professional.
- VSCode.
- Glueviz.
- Orange 3 App.
- RStudio.
- Anaconda Prompt (Windows only).
- Anaconda PowerShell (Windows only).

Nous avons utilisé comme environnement de développement Spyder, qui sera détaillé dans la section qui suit.

3.3. Environnement de développement : Spyder

Spyder est un environnement scientifique gratuit et open source écrit en Python, pour Python, et conçu par et pour des scientifiques, des ingénieurs et des analystes de données. Il présente une combinaison unique des fonctionnalités avancées d'édition, d'analyse, de débogage et de profilage d'un outil de développement complet avec l'exploration de données, l'exécution interactive,

l'inspection approfondie et les belles capacités de visualisation d'un package scientifique.**[Https://www.spyder-ide.org/]**(Voir figure 4.2, 4.3).

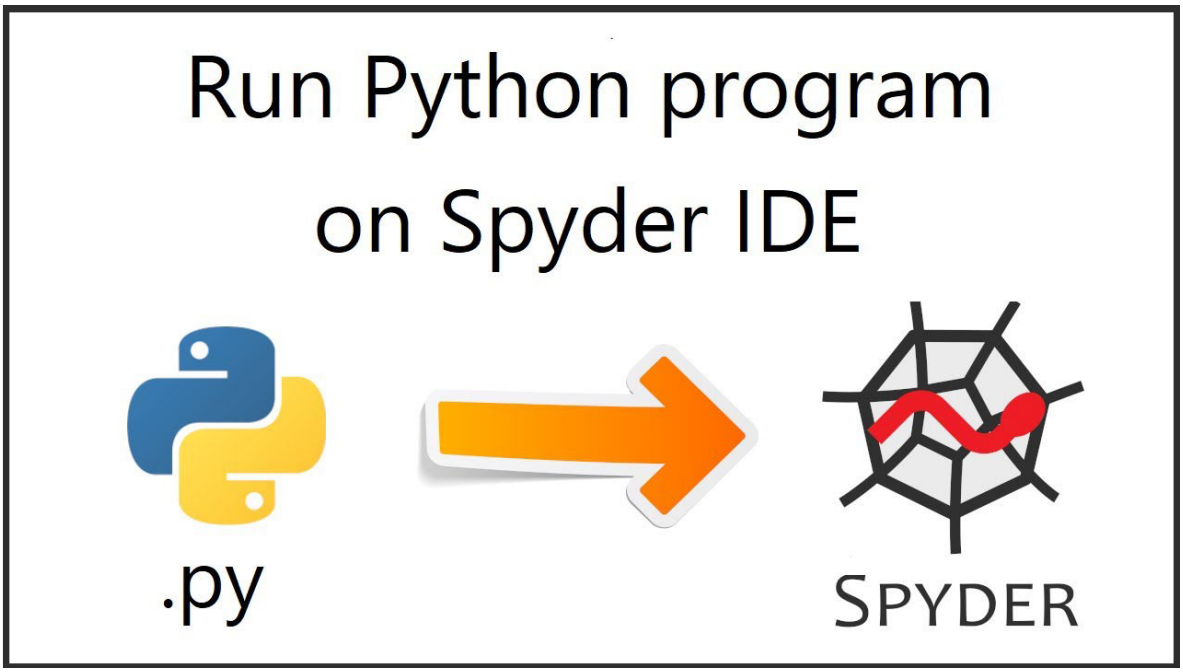

Figure 4.2 : Python, Spyder

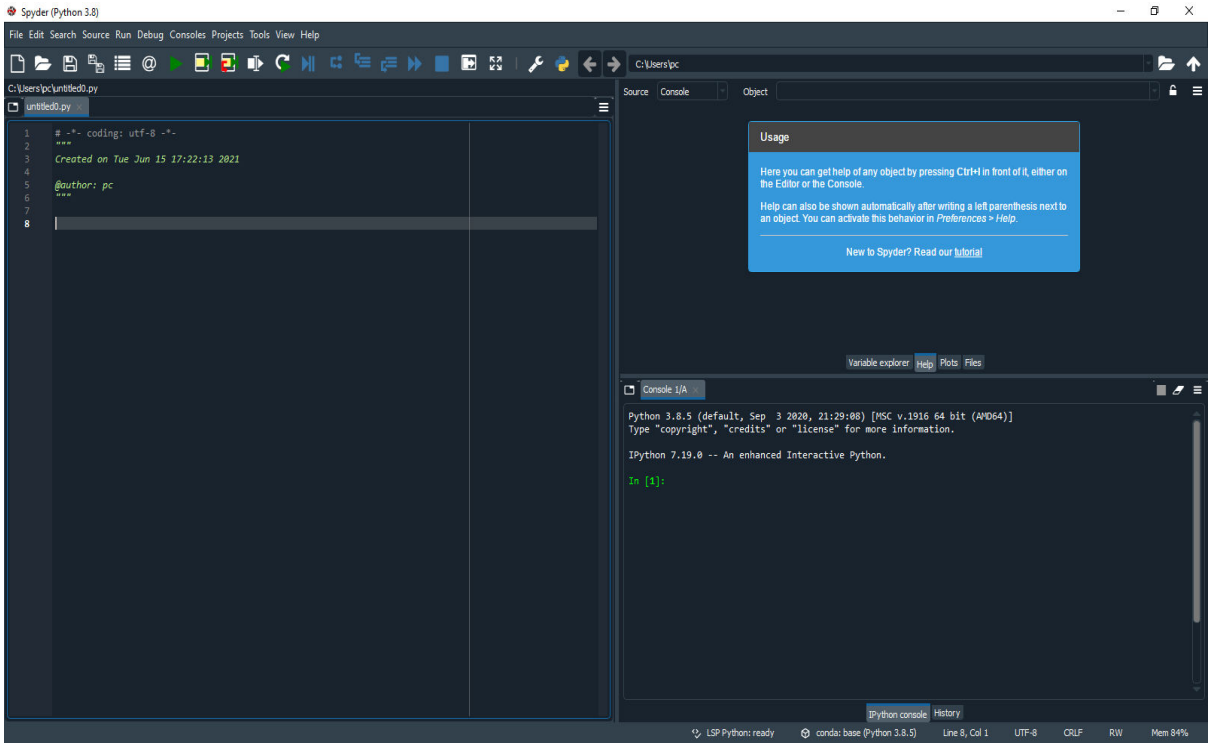

Figure 4.3 : L'interface de l'environnement Spyder.

4. Résultats

Notre modèle est divisé en deux parties. La première partie est l'extraction des mots clé et la deuxième partie la classification des pages web ce basant sur les résultats de la première partie.

4.1. L'extraction des mots clés

4.1.1. TF-IDF : le principe de la première partie consiste a :

-Lire le nombre de pages web à classées ainsi que leurs liens

- Suppression des balise HTML, CSS et JS.
- Le pré traitement qui est la séparation des mots avec des espaces
- Supprimons des stop-words (les mots vide).
- Trouver la racine des mots (Stemmer)
- Finalement le calcule de TF IDF.

Les captures ci-dessous montre les étapes de l'exécution en détail du TF-IDF (Figure 4.4,4.5, 4.6, 4.7,4.8,4.9).

- **1ère étape** : Nous avons entré le nombre de pages. (Voir figure 4.4, 4.5).

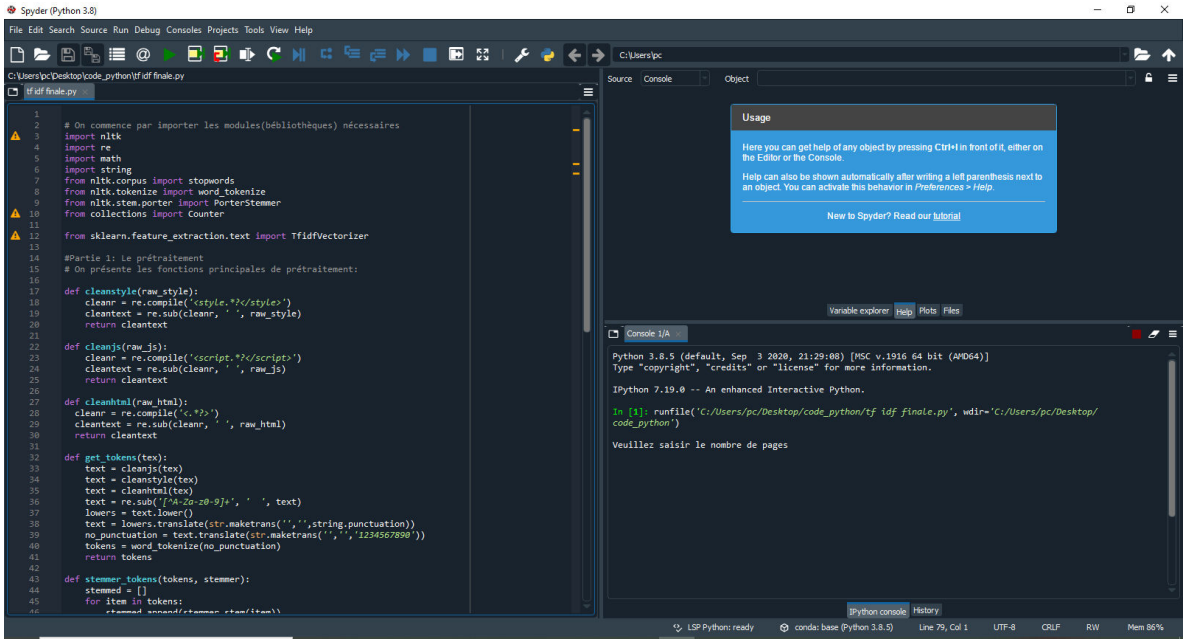

Figure 4.4 : La lecture de nombre de pages (a)

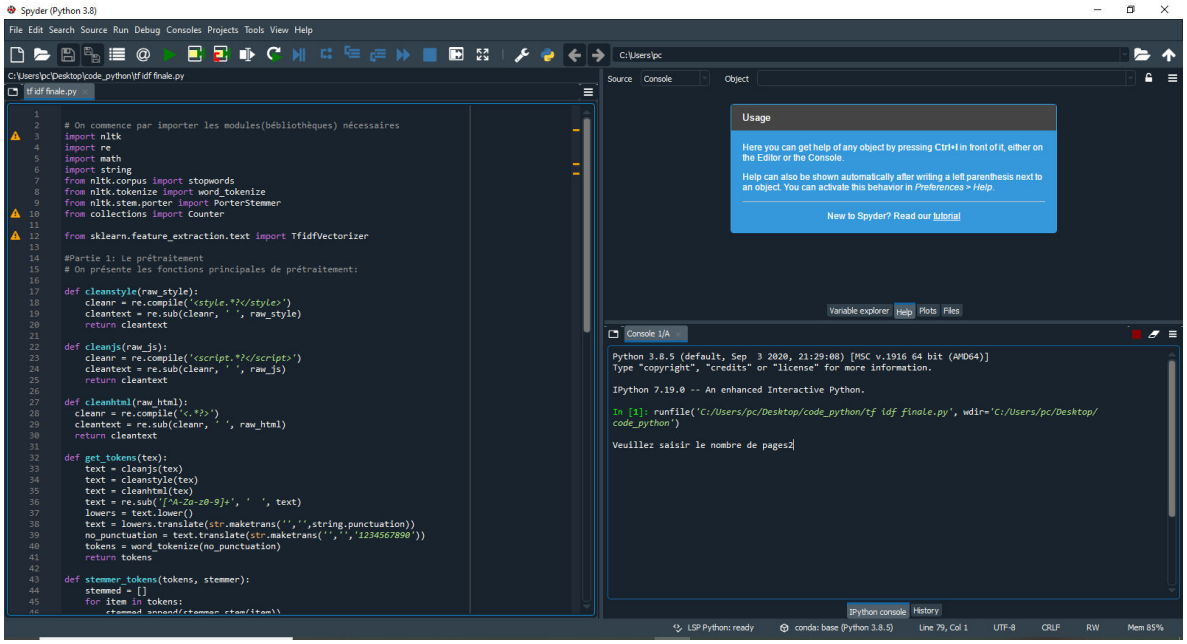

Figure 4.5 : La lecture de nombre de pages (b)

Remarque: le nombre de pages doit être supérieur ou égal à 2.

Dans notre exemple, Nous avons choisi deux pages.

- **2ème étape :** Nous avons entrer le lien des pages que nous voulons classer. (Voir figure 4.6, 4.7).

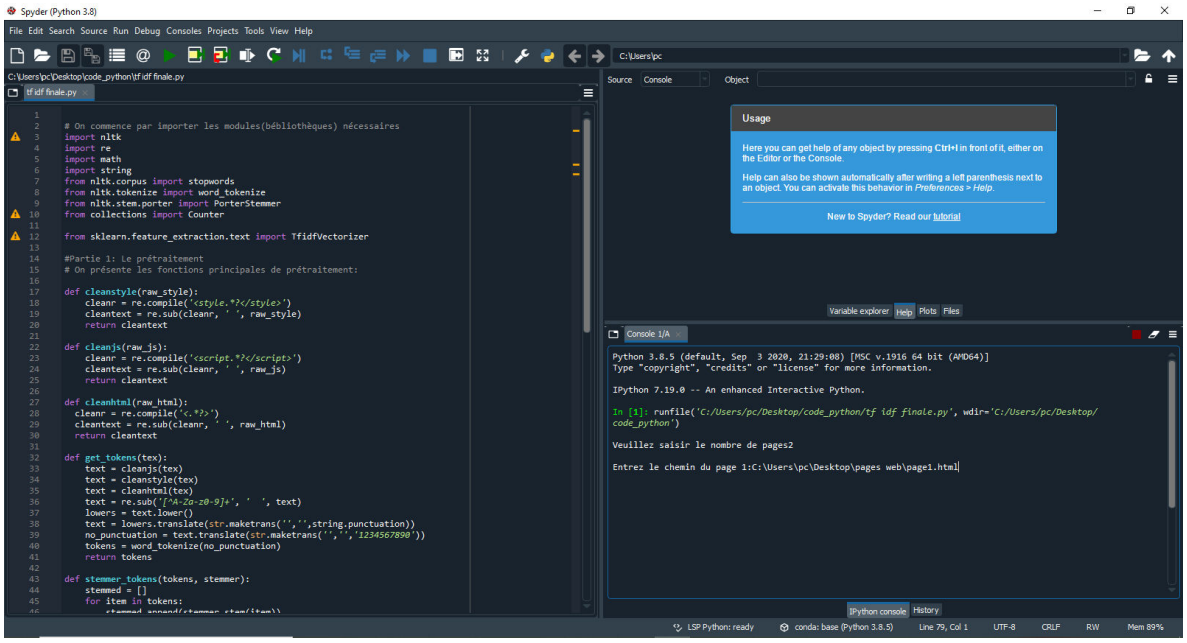

Figure 4.6 : La lecture de lien de la première page

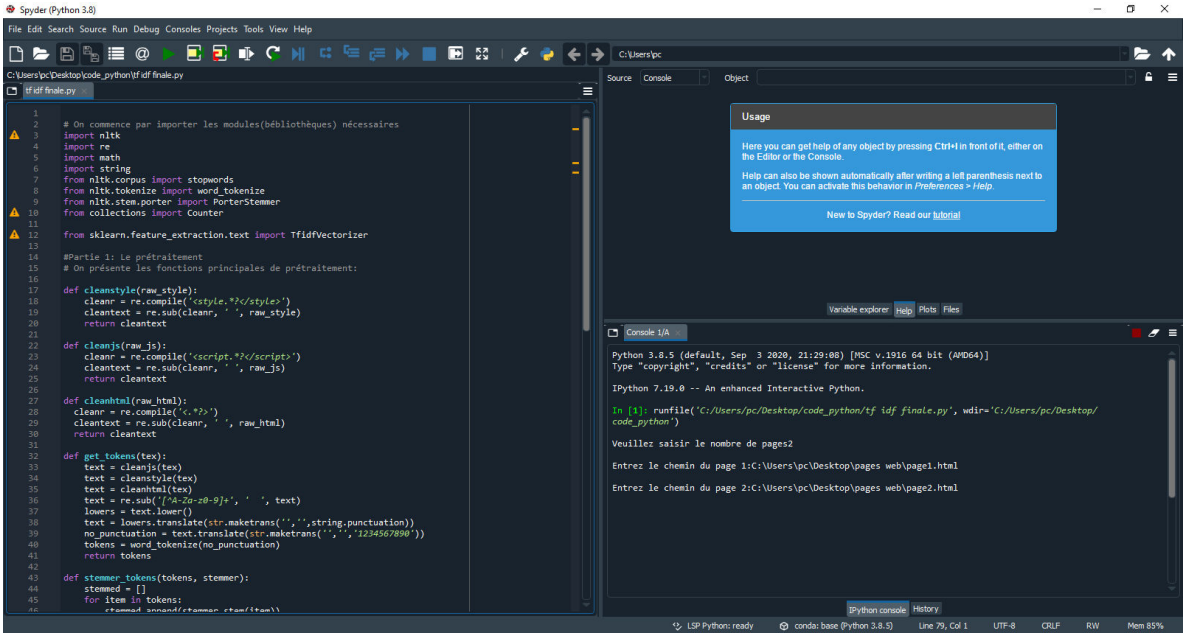

Figure 4.7 : La lecture de lien de la deuxième page

Pour le code source des pages voire la partie annexe.

Après quelque seconde le résultat de TF-IDF sera affiché. Voir les captures suivantes (figure 4.8, 4.9). Les mots sont affichés par ordre croissant selon leurs fréquences d'apparition dans la page.

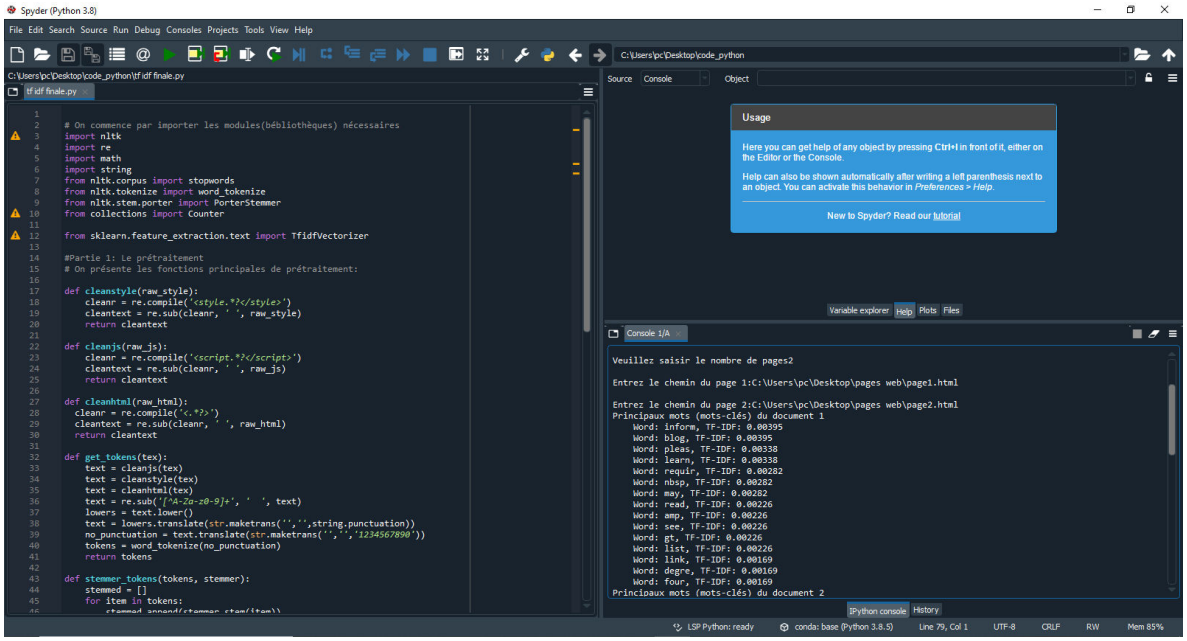

Figure 4.8 : Le résultat de TF IDF de la première page

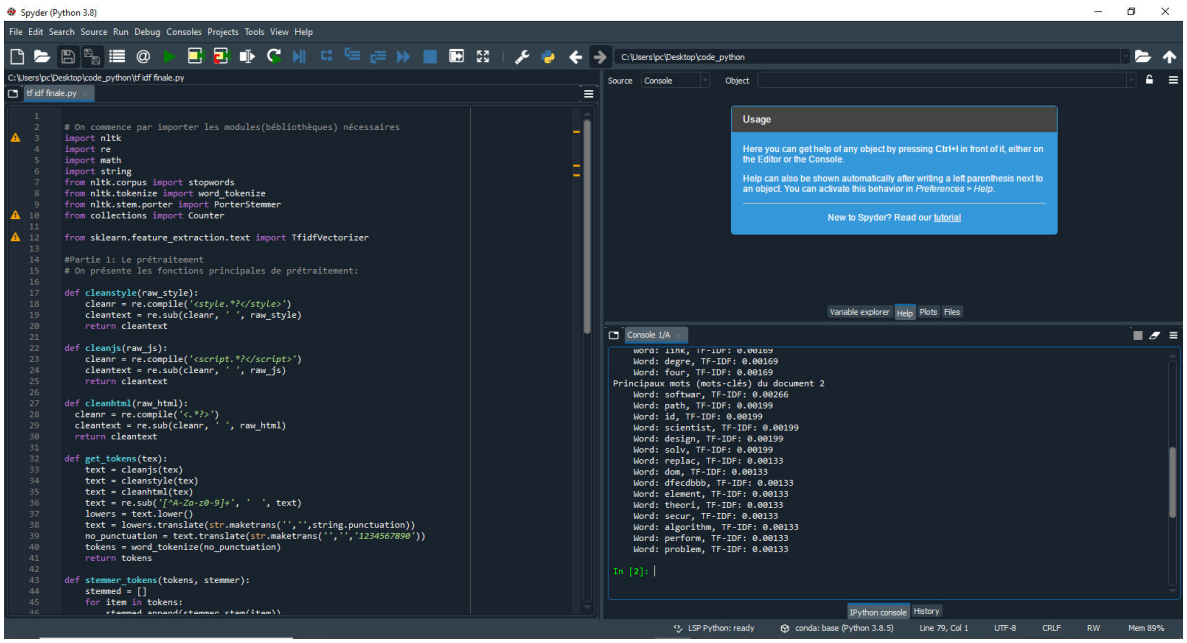

Figure 4.9 : Le résultat de TF IDF de la deuxième page.

4.2. La classification

Dans cette section en va décrire en détail la partie de classification. Commençons d'abord par notre dataset.

4.2.1. La préparation de notre dataset

Pour la construction du dataset, nous avons commencés par trouver tous les mots de chaque domaine qu'on a choisi, on a utilisé un générateur de mots en ligne nommé Related Words. Puisque les mots résultats du TF-IDF sera sous forme de stemmer, la forme des mots du dataset serons les mêmes,

La partie ci-dessous montre les différentes étapes pour construire notre dataset.

- **1ère étape :** Générer les mots de chaque domaine à l'aide de générateur de mots mentionné précédemment (Voir figure 4.10).

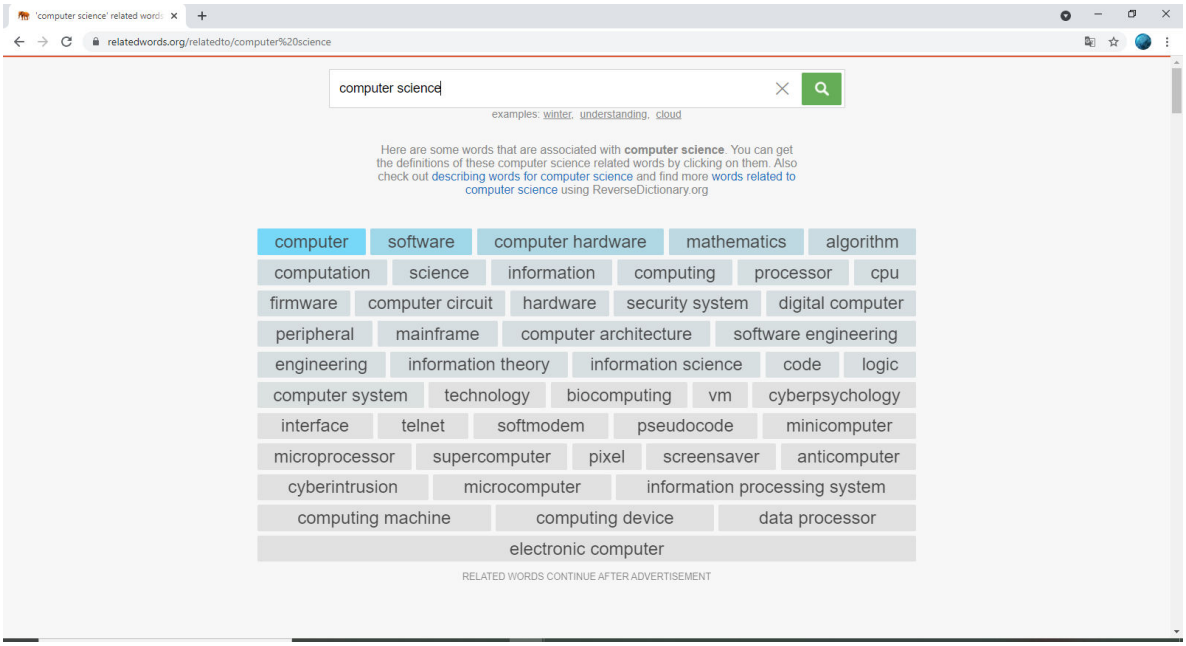

Figure 4.10 : Les résultats des mots du domaine informatique trouvé par le générateur de mots Related Words

- **2ème étape :** Nous sauvegardons les résultats du générateur dans un fichier texte (Voir figure 4.11).

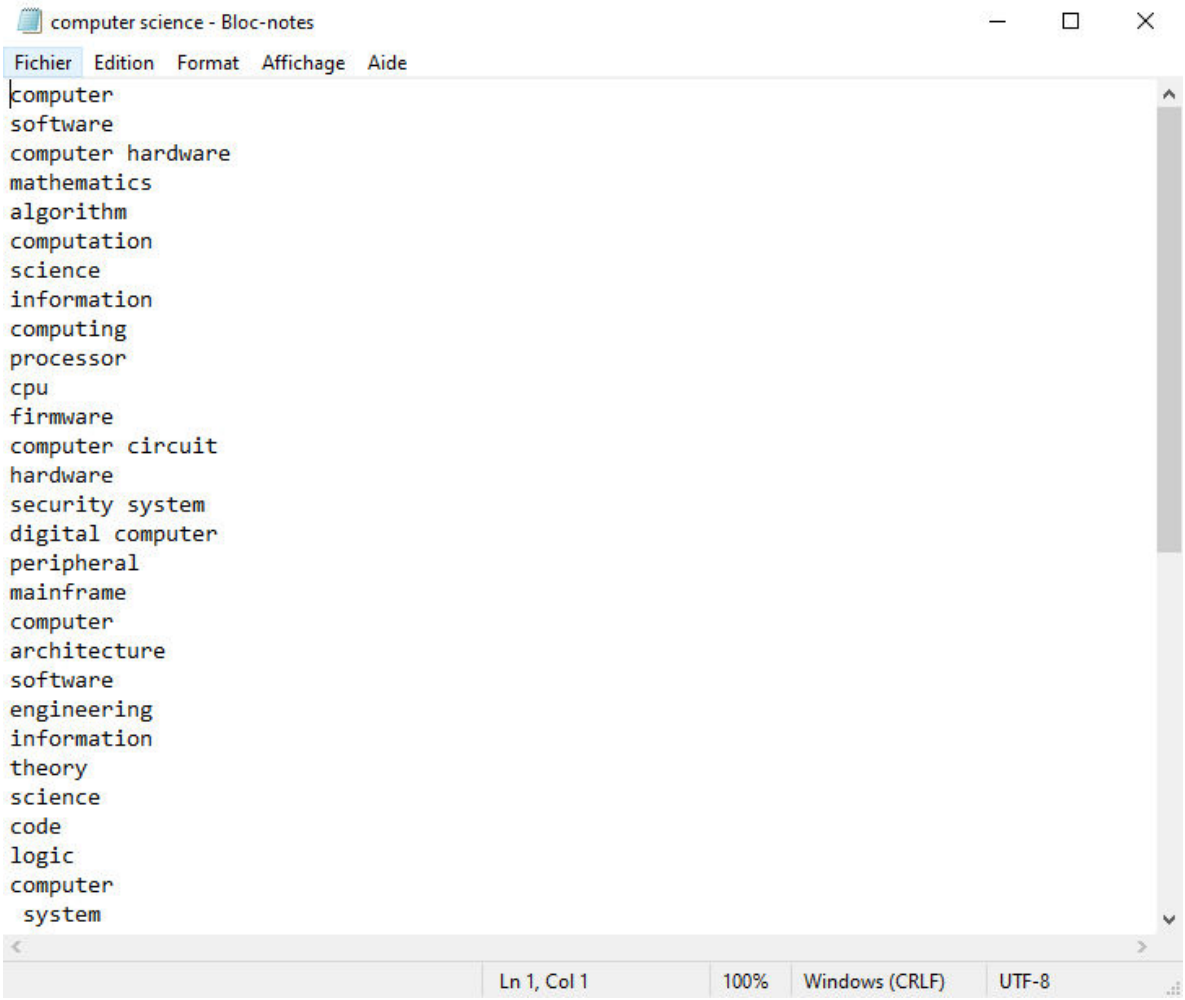

Figure 4.11 : Le résultat de générateur sous forme de fichier texte

- **3ème étape :** Nous avons appliqué un algorithme pour supprimer les mots vide et trouver la racine (Stemmer) du chaque mot (Voir figure 4.12).

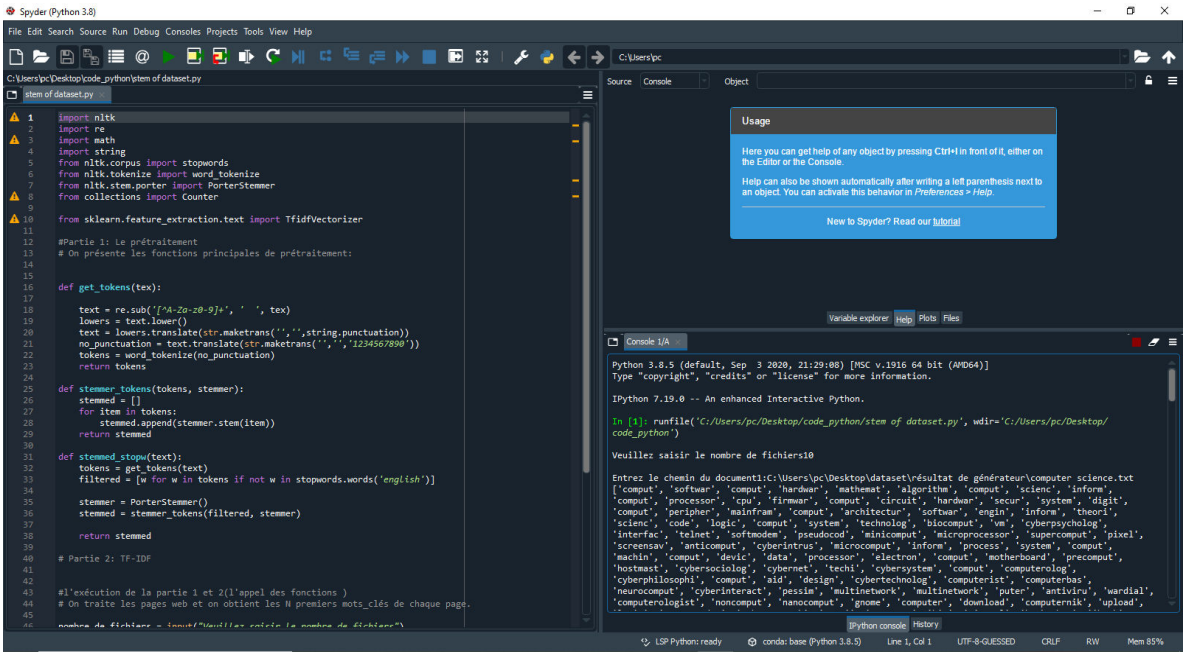

Figure 4.12 : Le résultat d'algorithme de Stemmer appliqué sur le résultat de générateur de mots

4ème étape : Ensuite nous avons sauvegardé le résultat d'algorithme de stemmer dans un autre fichier texte comme montre la figure ci-dessous.

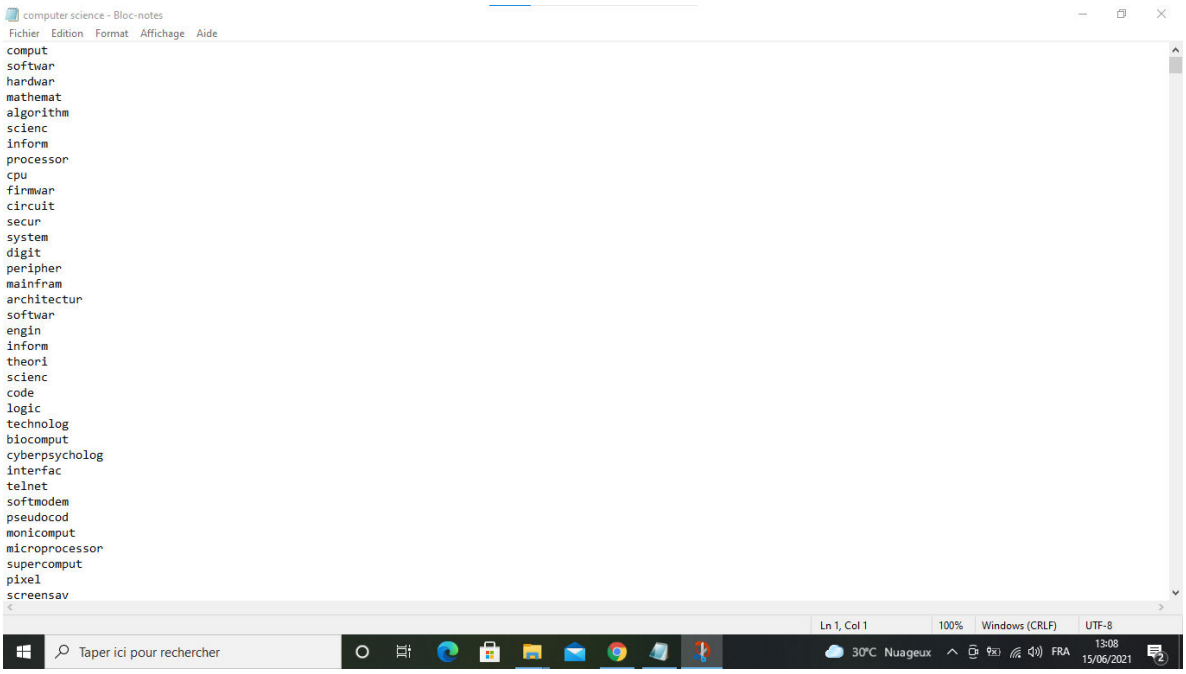

Figure 4.13. Le résultat de l'algorithme Stemmer sous forme d'un fichier texte

- **5ème étape :** nous avons utilisé l'outils en ligne TEXT FIXERFR[https://www.textfixerfr.com/] pour réorganiser les mots par ordre alphabétique du fichier de l'étape précédente (voir figure 4.14, 4.15, 4.16).

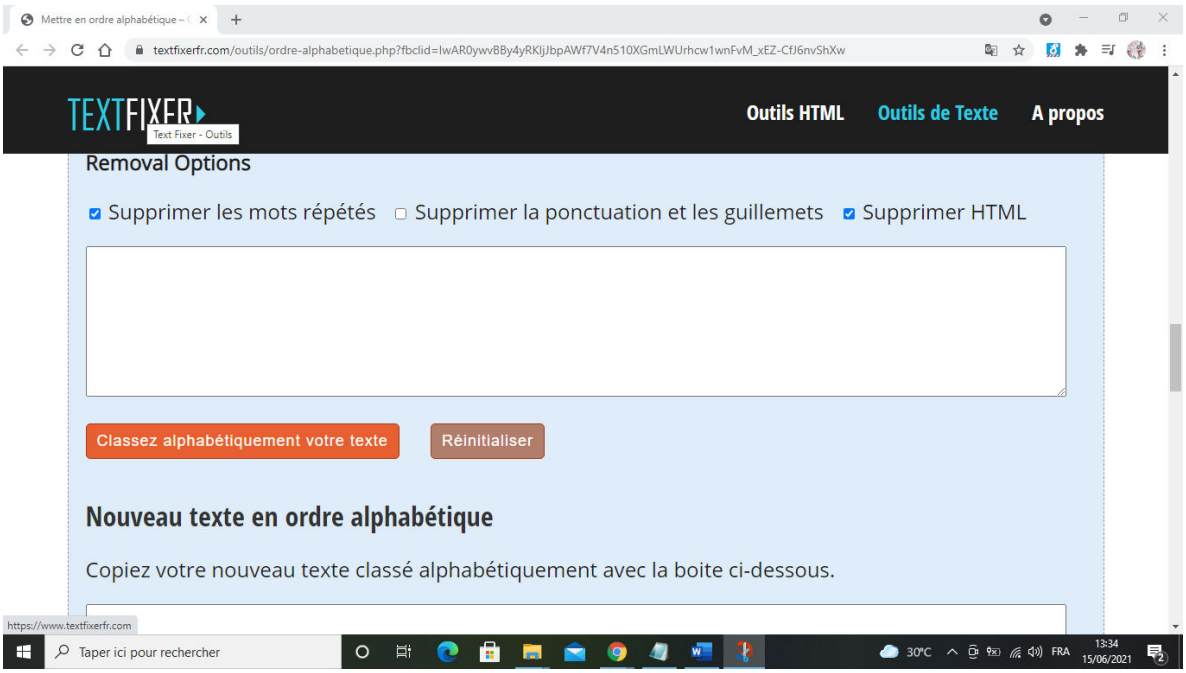

Figure 4.14 : L'interface graphique de l'outils TEXT FIXERFR

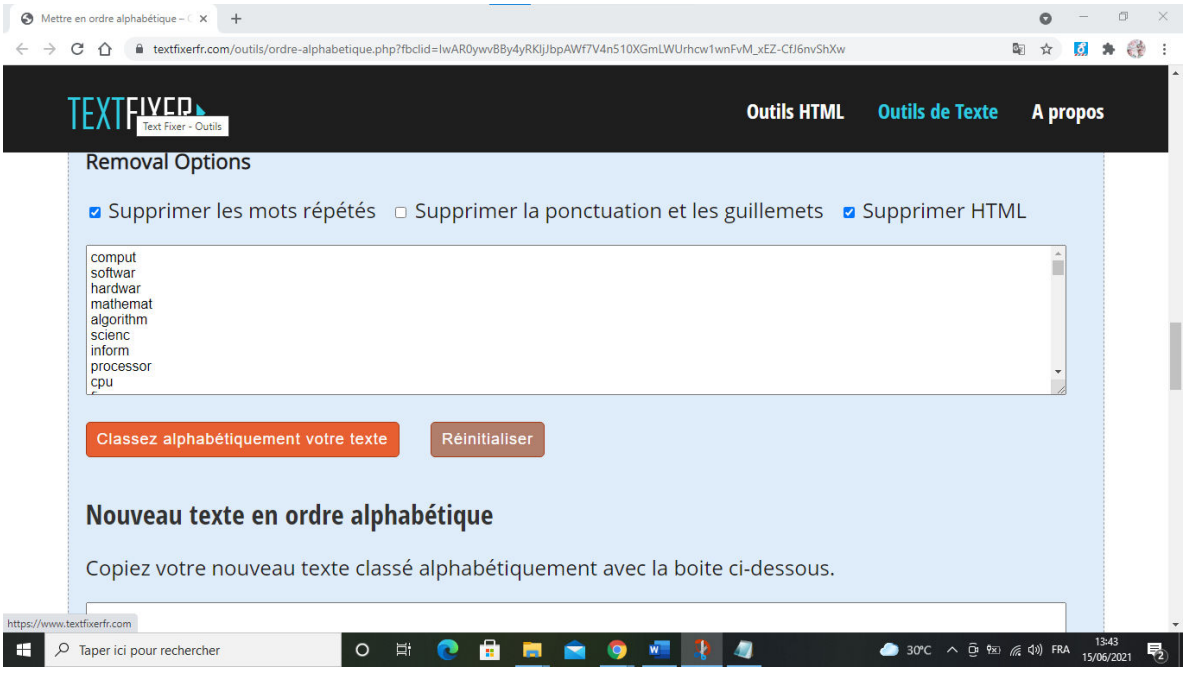

Figure 4.15 : Les mots qu'on a ajoutés dans le TEXT FIXERFR.

Figure 4.16 : Le résultat de l'ordre alphabétique des mots.

- **6ème étape :** ensuite nous avons enregistré le résultat après les avoir mis en ordre alphabétique dans un fichier Word afin de mettre la première lettre de chaque mot en majuscule (voir figure 4.17, 4.18).

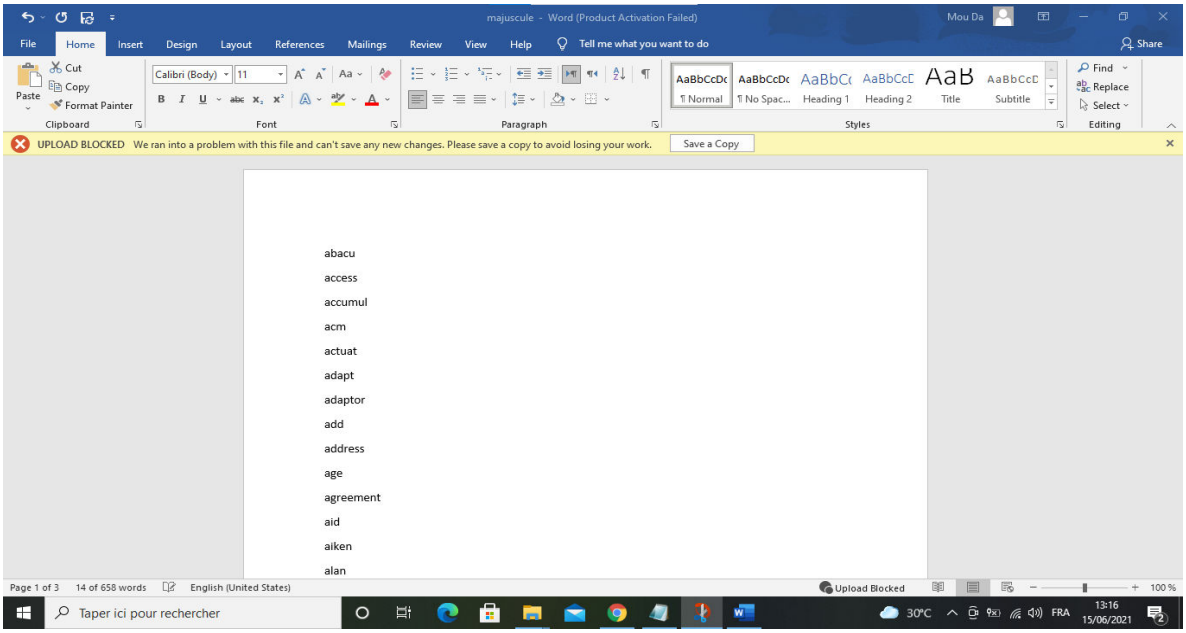

Figure 4.17 : Fichier Word qui contient les mots après le résultat de l'ordre alphabétique

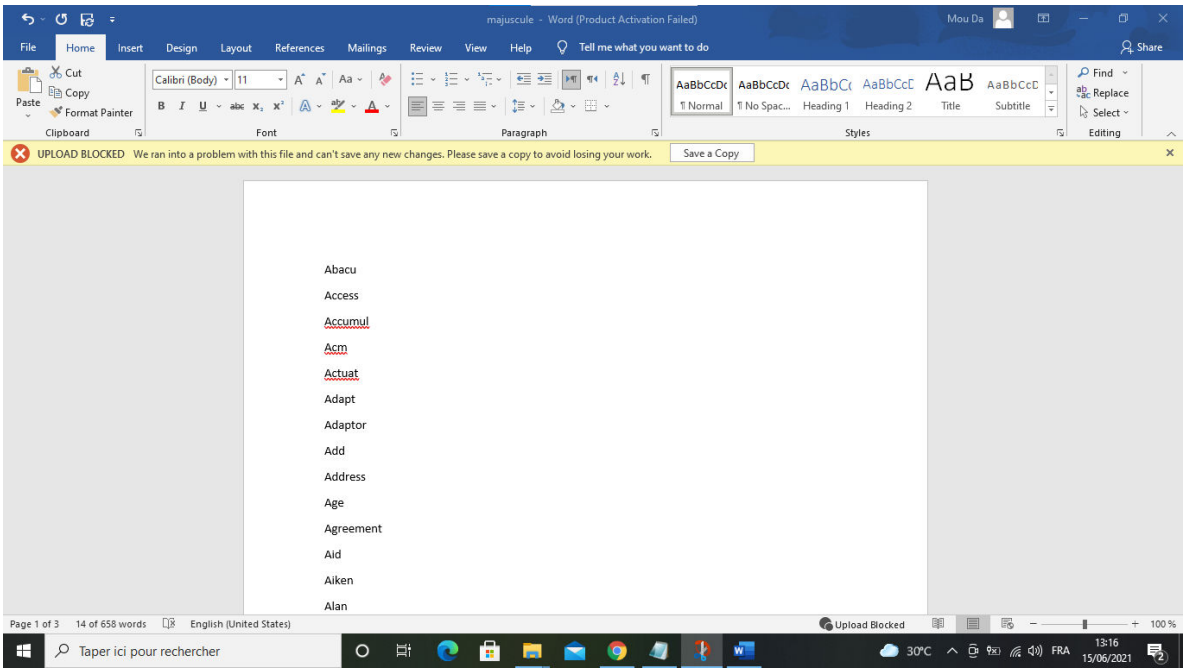

Figure 4.18 : Première lettre de chaque mot en majuscule

- **7ème étape :** Puis nous avons copié les mots qui sont dans le fichier Word dans un fichier texte (voir figure 4.19).

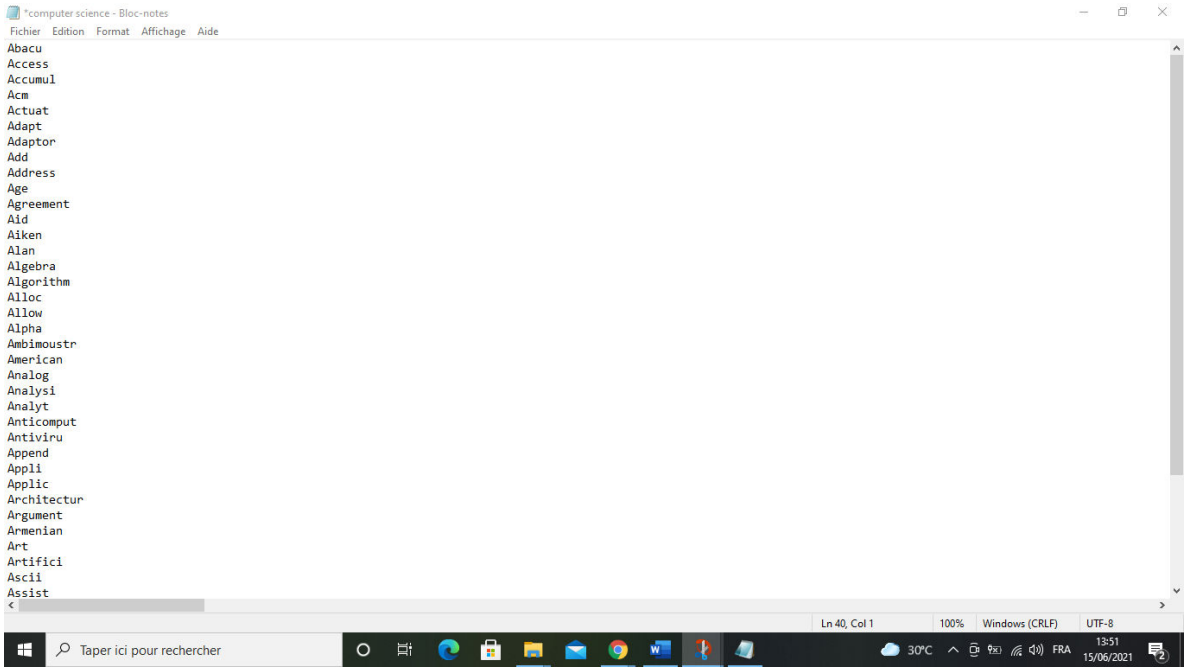

Figure 4.19 : Fichier texte contient les mots de notre domaine informatique

Et ce processus sera répété avec tous les domaines de notre dataset.

- **8ème étape :** nous avons enregistré tous les fichiers textes des domaines choisis dans un document « domaine », et ce dernière sera sauvegardé dans un autre document « Data » (voir figure 4.20 ,4.21 ,4.22).

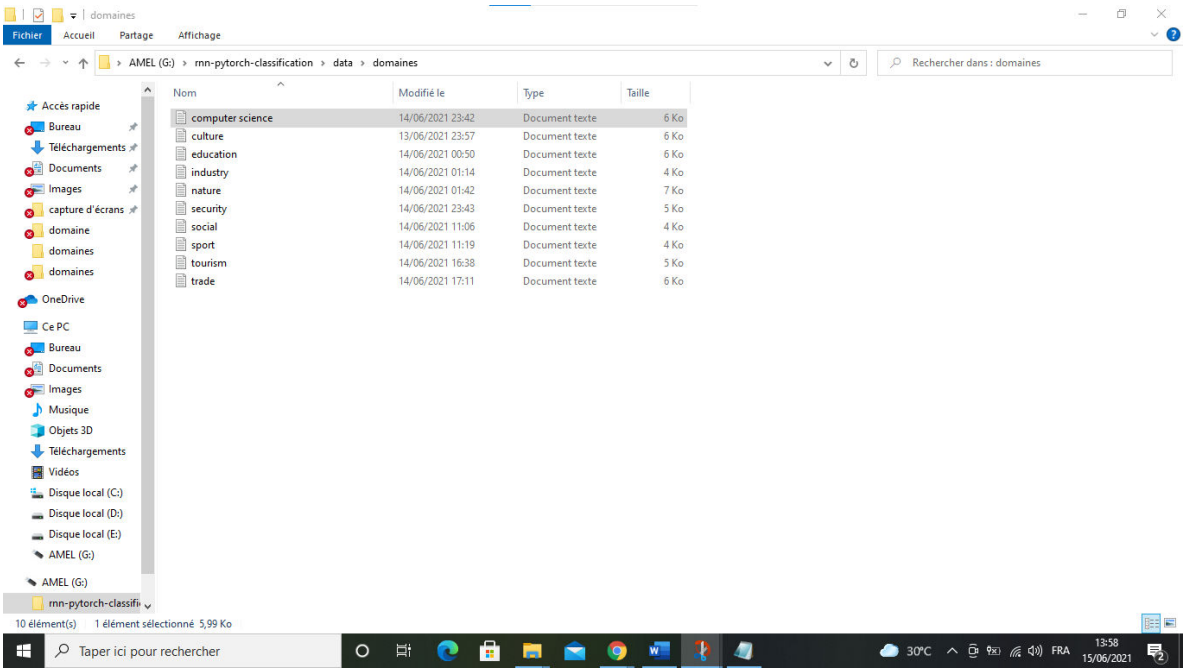

Figure 4.20 : Présentation des domaines de notre dataset.

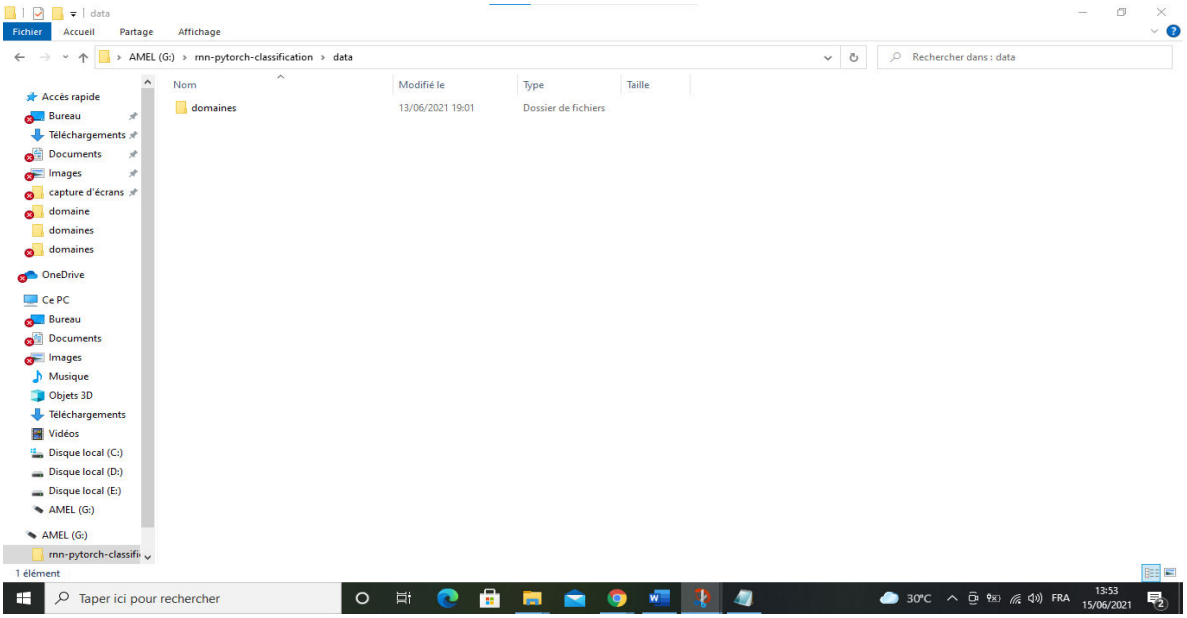

Figure 4.21 : Le déplacement de nos fichiers dans le dossier « domaines »

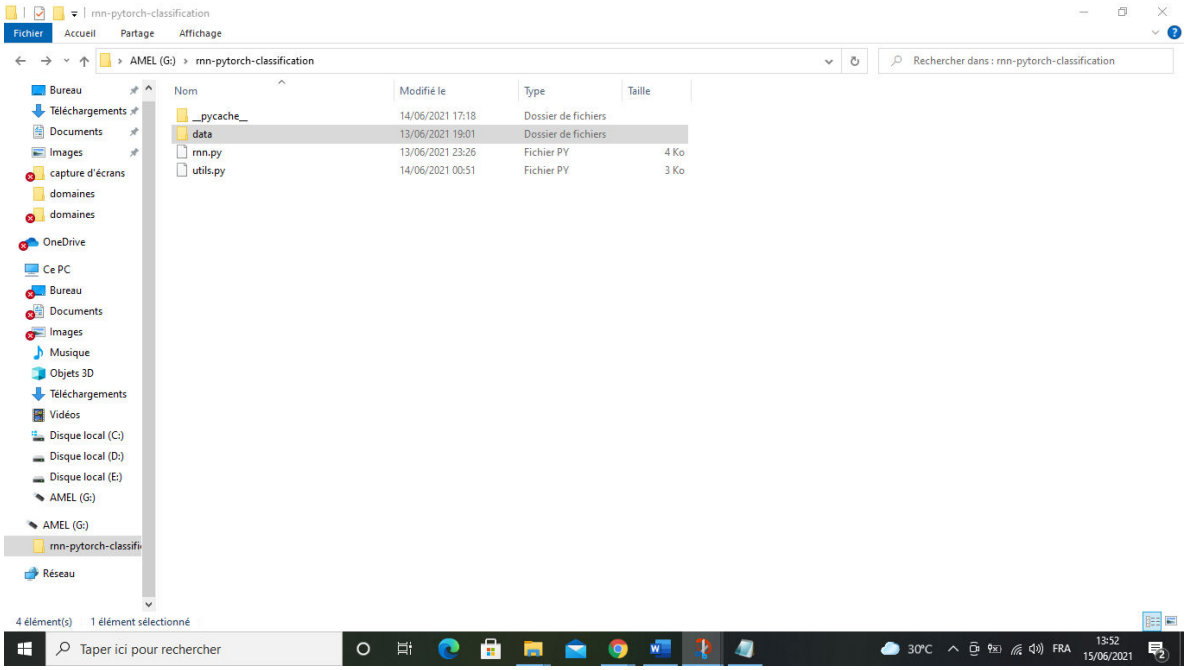

Figure 4.22 : La création de dossier data qui contient les fichiers de chaque domaine de dataset.

4.2.2. La classification avec réseau de neurone

Après la préparation de notre Dataset, on va faire la classification des pages web quand a déjà calculé leurs fréquences de mot clés dans la première partie à l'aide de l'algorithme TF-IDF. Dans cette étape on va classer les pages selon le mot clé qui a la plus grande fréquence on utilisant les réseaux de neurones.

Les captures ci-dessous montrent les étapes de l'exécution :

On commence d'abord par la lecture de notre dataset en utilisant la fonction « Laod-data (voir figure 4.23).

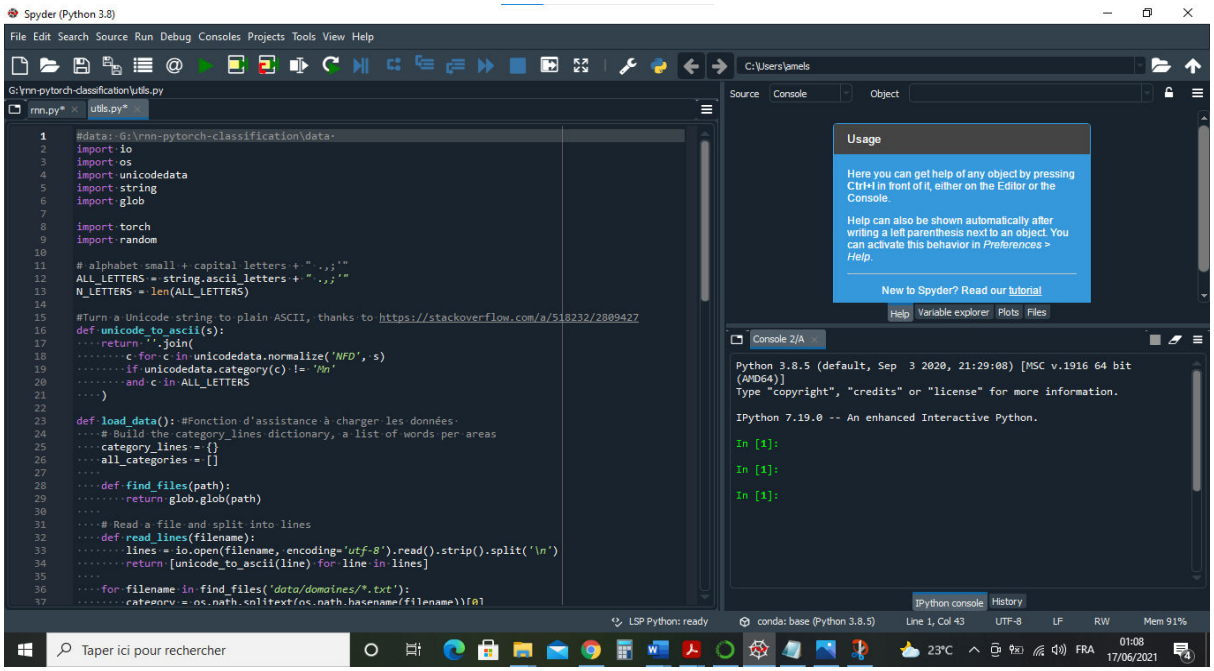

Figure 4.23 : La lecture de donné du dataset.

Après la lecture du dataset, on va faire l'exécution du réseau de neurone, après quelque heures le résultat de classification sera affiché.

Une courbe qui trace toutes les pertes et le taux d'erreur de notre dataset, est présentée par la figure ci-dessous .

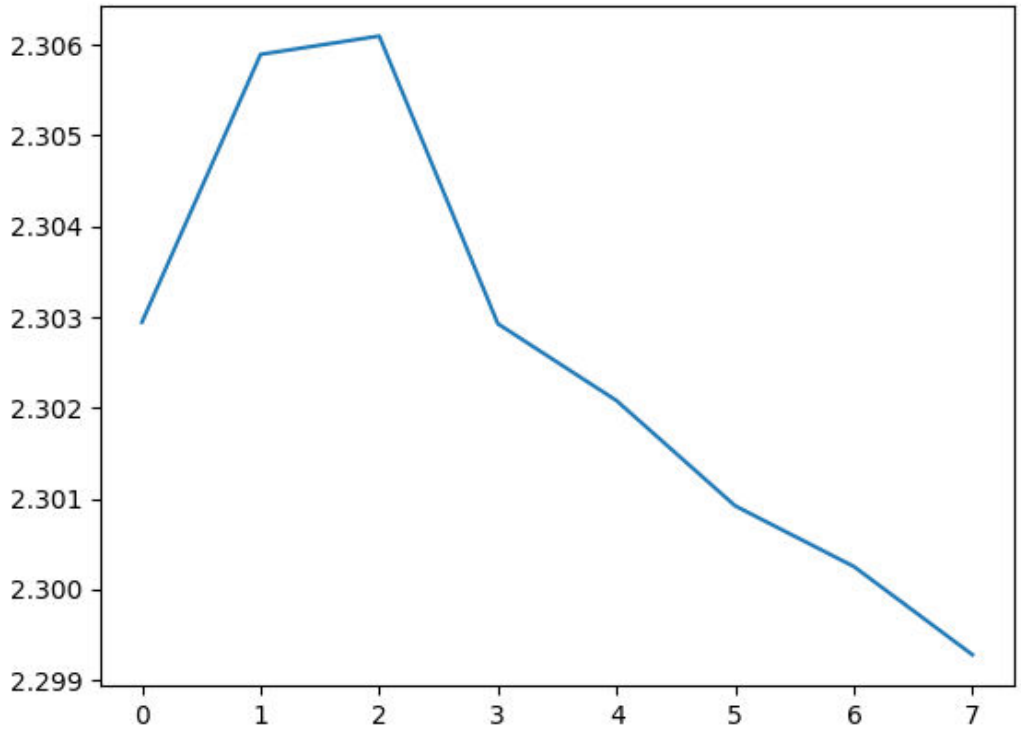

Figure 4.24 : Courbe qui trace toutes les pertes et le taux d'erreur de dataset

- **2ème étape :** nous entrainons notre réseau de neurone jusqu'à l'obtention du bon résultat de classification (voir figure 4.25,4.26,4.27,4.28 ,4.29,4.30)

```
================== RESTART: E:\rnn-pytorch-classification\rnn.py ================
social
5000 62.5 2.2556 Dark  / trade WRONG (sport)
Input:Inform

> Inform
nature
Input:inform

> inform
nature
Input:Softwar
```

Figure 4.25 : L'entrainement de RNN (a)

```
================= RESTART: E:\rnn-pytorch-classification\rnn.py ================
tourism
5000 62.5 2.3049 Self  / sport WRONG (industry)
Input:Inform

> Inform
sport
Input:
```

Figure 4.26 : L'entrainement de RNN (b)

```
================= RESTART: E:\rnn-pytorch-classification\rnn.py ================
industry
5000 62.5 2.3779 Behavior  / social WRONG (education)
Input:Inform

> Inform
trade
Input:
```

Figure 4.27 : L'entrainement de RNN (c)

```
education
5000 62.5 2.1771 Inform  / security CORRECT
Input:inform

> inform
education
Input:inform

> inform
education
Input:inform

> inform
education
Input:Inform
```

Figure 4.28 : L'entrainement de RNN (d)

```
================ RESTART: E:\rnn-pytorch-classification\rnn.py ================
security
5000 62.5 2.2319 Geolog  / sport WRONG (nature)
Input:Inform

> Inform
computer science
Input:
```

Figure 4.29 : L'entrainement de RNN (e)

```
================ RESTART: E:\rnn-pytorch-classification\rnn.py ================
security
5000 62.5 2.2319 Geolog  / sport WRONG (nature)
Input:Inform

> Inform
computer science
Input:Software

> Software
industry
Input:Softwar

> Softwar
tourism
Input:
================ RESTART: E:\rnn-pytorch-classification\rnn.py ================
security
5000 62.5 2.2241 Cuisin  / education WRONG (culture)
Input:Software

> Software
computer science
Input:
```

Figure 4.30 : L'entrainement de RNN (f)

5. Conclusion

Dans ce chapitre, nous avons présenté l'implémentation de notre méthode, nous avons utilisé pour cela plusieurs outils tel que tel que Spyder et Anaconda, pour l'implémentation nous avons choisis Python. Les résultats obtenus montrent que notre méthode donne de bon résultat

CONCLUSION GENERALE ET PERSPECTIVES

Le travail réalisé dans le cadre de ce mémoire se base principalement sur l'utilisation combinée de deux méthodes : supervisée et non supervisée, la classification des pages web et l'extraction des mots clés. Dans un premier temps, et précisément dans le premier chapitre, nous nous sommes intéressés à dresser un état de l'art sur les techniques de classification des pages web qui existent. De manière générale, cette étape est divisée en deux parties, une consacrée à la présentation des approches de classification en générale (supervisées et non supervisé) et l'autre dédiée aux approches de classification des pages web (approches de base, algorithmes de classification des pages web…).

Ensuite dans le deuxième chapitre nous avons mentionné les différentes méthodes et approches d'extractions des mots clés (supervisé et non supervisé).

Cet état de l'art sur les techniques de classification nous a mené à prendre connaissance de leur complexité. En effet, avec le nombre incroyable des techniques qui existent, le choix de l'une d'entre elles est devenu très difficile. De plus, la plupart de ces derniers sont bien plus liées au jeu de données qu'à l'exactitude théorique de la technique elle-même. Le choix d'une méthode de classification dans ce cas dépend essentiellement des résultats obtenus par la technique d'extraction des mots clés.

La tâche d'extraction automatique de termes-clés consiste à analyser un document pour en extraire les expressions (phrasèmes) les plus représentatives de celui-ci. Les méthodes d'extraction automatique de termes-clés sont réparties en deux catégories : les méthodes supervisées et les méthodes non supervisées. Les méthodes supervisées réduisent la tâche d'extraction de termes-clés à une tâche de classification binaire (tous les phrasèmes sont classés parmi les termes clés ou les non termes-clés). Cette classification est possible grâce à une phase

préliminaire d'apprentissage, phase qui n'est pas requise par les méthodes non-supervisées. Ces dernières utilisent des caractéristiques (traits) extraites du document analysé (et parfois d'une collection de documents de références) pour vérifier des propriétés permettant d'identifier ses termes-clés.

L'objectif principal de ce mémoire est l'extraction des mots-clés pour la classification des pages web est de rassembler les pages similaires selon une certaine catégorie, au sein d'une même classe ou catégorie à l'aide de la technique d'extraction des mots clés TF IDF et d'outil de classification qui est les réseaux de neurones récurrents.

Les résultats montrent que la méthode proposée nous donne de bons résultats en ce qui concerne la classification des pages web.

Perspectives

Le sujet étant très vaste, il reste beaucoup à faire pour améliorer ce travail, on peut donc proposer comme perspectives, d'ajouter d'autres langues pour rendre le système multi-langues, d'intégrer d'autres techniques et méthodes de classification supervisée, ainsi que toute autre idée jugée utile, réalisable et bénéfique.

BIBLIOGRAPHIE

Batagelj V., Zaveršnik M., *Fast algorithms for determining (generalized) core groups insocial networks*, Advances in Data Analysis and Classification, vol. 5, n° 2, p. 129-145, 2011.

Boudin, f. Et morin, e., Keyphrase Extraction for N-best Reranking in Multi-Sentence Compression. *In Proceedings of the North American Chapter of the Association for ComputationalLinguistics (NAACL)*, 2013.

Choi B,Making sense of search results by automatic web-page classification. In: WebNet, Orlando, Florida, USA, pp 184-186, 2001.

Ding, z., zhang, q. Et huang, x., Keyphrase Extraction from Online News Using Binary Integer Programming. In Proceedings of 5th International Joint Conference on Natural Language Processing, 2011.

Dural Burak, Türkçe Arama Motoru Sonucu Kümeleme Çalışmaları, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği, Yayım lanmamış Yüksek Lisans Tezi, İstanbul, 2013.

Ebubekirbuber et Banu diri, Web Page Classification Using RNN, www.Sciencedirect.com , Procedia Computer Science 154 (2019) 62–72, 2019.

Eichler, k. Et neumann, g., DFKI KeyWE : Ranking Keyphrases Extracted from ScientificArticles. *In Proceedings of the 5th International Workshop on Semantic Evaluation,* 2010.

Ercan, g. Et cicekli, i., Using Lexical Chains for Keyword Extraction, 2007.

Frank, E., Paynter, G., Witten, I., Gutwin, C. Et Nevill-Manning, C., Domain-Specific Keyphrase Extraction, 1999.

Herbrich, r., graepel, t. Et obermayer, k., Support Vector Learning for Ordinal Regression. *In Artificial Neural Networks, 1999*.

Horri Mohamed et Kessi Ali, Calcule des descripteurs dans le but de la classification automatique d'objet 3D, 2017.

Hulth, a. Improved Automatic Keyword Extraction Given More Linguistic Knowledge.*In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*.

Jiang, x., hu, y. Et li, h., A Ranking Approach to Keyphrase Extraction. *In Proceedingsof the 32nd international ACM SIGIR conference on Research and development in informationretrieval*, 2009.

Joachims, t., Training Linear SVMs in Linear Time. *In Proceedings of the 12th ACMSIGKDD international conference on Knowledge discovery and data mining*, 2006.

Kan MY, Thi HO. Fast webpage classification using URL features. In Proceedings of the 14th ACM international conference on Information and knowledge management 2005 Oct 31 (pp. 325-326).

Kessler B, Nunberg G, Schutze H, Automatic detection of text genre. In: Proceeding of 35th annual meeting of the association for computational linguistics. Madrid, Spain, pp 32-38, 1997.

Liu, z., chen, x., zheng, y. Et sun, m., Automatic Keyphrase Extraction by BridgingVocabulary Gap. *In Proceedings of the 15th Conference on Computational Natural LanguageLearning,* 2011.

Liu, Z., Li, P., Zheng, Y. Et Sun, M., Clustering to Find Exemplar Terms for KeyphraseExtraction. *In Proceedings of the 2009 Conference on Empirical Methods in Natural LanguageProcessing : Volume 1*, 2009.

Manning, c.d., p. raghavan, and h. schutze, introduction to information retrieval. irbook, may 27, 2008.

Matsuo, Y. Et Ishizuka, M., Keyword Extraction from a Single Document Using WordCo-occurrence Statistical Information , 2004.

Mihalcea, r. Et tarau, p., Textrank : Bringing Order Into Texts. *In Proceedings of the2004 Conference on Empirical Methods in Natural Language Processing*, 2004.

Mitchell, T. M., Machine Learning. McGraw-Hill, New York, NY, 1997.

Mladenic D, Feature subset selection in text-learning. In: Proceeding of the 10th European conference on machine learning (ECML98). Chemnitz,Germany, pp 95-100, 1998.

Nguyen, t. Et kan, m., Keyphrase Extraction in Scientific Publications. *In Proceedings ofthe 10th international conference on Asian digital libraries : looking back 10 years and forging new frontiers*, 2007.

Ouali Siham et Chekaiem Abdelfattah,L'Extraction de Mots Pertinents pour la Classification de Textes Arabes, 2019.

Paukkeri, m. Et honkela, t., Likey : Unsupervised Language-Independent Keyphrase Extraction. In Proceedings of the 5th International Workshop on Semantic Evaluation, 2010.

Pooja Vinod Nainwani et Purvi Prajapati,Comparative Study of Web Page Classification Approaches,International Journal of Computer Applications (0975 – 8887)Volume 179 – No.45, May 2018.

Quang, C. T., Classification automatique des textes vietnamiens Hanoi, Institutde la Francophonie pour l'informatique, 2005.

Rousseau F., Vazirgiannis M., *Main core retention on graph-of-words for single document keyword extraction*, In Advances in Information Retrieval, p. 382–393, 2015.

S. Prabhu, N. VENKATESAN, Data Mining and Warehousing, New Age International (P) Ltd., Publishers, New Delhi, 2007.

Sarkar, k., nasipuri, m. Et ghose, s., A New Approach to Keyphrase Extraction Using Neural Networks, 2010.

Sarkar, k., nasipuri, m. Et ghose, s., Machine Learning Based Keyphrase Extraction :Comparing Decision Trees, Naïve Bayes, and Artificial Neural Networks, 2012.

Selamat, A., &Omatu, S.,Web page feature selection and classification using neural networks. Information Sciences, 158, 69–88, 2004.

Sujian, l., houfeng, w., shiwen, y. Et chengsheng, x., News-Oriented Key word Indexing with Maximum Entropy Principle, 2003.

T.DERDRA Amel, F.BENSFIA, « La Représentation Conceptuelle pour la Catégorisation des Textes Multilingue », Mémoire de Master, Université Abou Bakr Belkaid– Tlemcen, 2011-2012.

Turney, P., Learning Algorithms for Keyphrase Extraction, 1999.

Turney, P., Coherent Keyphrase Extraction via Web Mining, 2003.

Wan, x. Et xiao, j., Single Document Keyphrase Extraction Using Neighborhood Knowledge. In Proceedings of Association for the Advancement of Artificial Intelligence, 2008.

Wan, x, yang, j. Et xiao, j., Towards an Iterative Reinforcement Approach forSimultaneous Document Summarization and Keyword Extraction. *In Annual Meeting association For Computational Linguistics*, 2007.

Witten, i., paynter, g., frank, e., gutwin, c. Et nevill-manning, c., KEA: Practical Automatic Keyphrase Extraction. *In Proceedings of the 4th ACM conference on Digital libraries*, 1999.

Witten IH, Frank E, Hall MA, Pal CJ. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann; 1 Oct 2016.

Xiaoguang Qi et Brian D. Davison,Web Page Classification: Features and Algorithms,ACM Computing Surveys, 2009.

Zhang, k., xu, h., tang, j. Et li, j., Keyword Extraction Using Support Vector Machine, 2006.

Résumé

Avec l'augmentation du nombre d'internautes, la croissance des sites Web est proportionnelle. En conséquence, le classement des pages Web est devenu un énorme sujet de recherche ces dernières années. Cela a fait une demande toujours croissante pour des techniques de classification automatisées avec une précision de classification élevée. Pour catégoriser et manipuler automatiquement les pages Web, les systèmes actuels utilisent un contenu de page visuel, qui comprend le contenu affiché. Cependant, jusqu'à présent, peu de travaux ont été réalisés sur l'utilisation de contenu textuel et de code HTML.

Dans ce travail, nous proposons une méthode, de classification de pages Web, basée sur leur contenu textuel. Les pages Web présentent en général des informations de différentes classes variées en fonction de leurs sujets spécifique. Cette méthode est basée sur la technique d'extraction des mots clé d'une page web (contenu textuel) combiné avec une approche supervisée d'apprentissage automatique à savoir les réseaux de neurones.

Mots-clés : classification de pages Web, extraction de mots clés, approches supervisées, apprentissage automatique.

ملخص

مع زيادة عدد مستخدمي الإنترنت، يكون نمو مواقع الويب متناسبًا. نتيجة لذلك، أصبح ترتيب صفحات الويب موضوعًا كبيرًا للبحث في السنوات الأخيرة. وقد أدى ذلك إلى زيادة الطلب على تقنيات التصنيف التصنيف. لتصنيف صفحات الويب ومعالجتها تلقائيًا، تستخدم الأنظمة الحالية الآلي ذات الدقة العالية في محتوى الصفحة المرئي، والذي يتضمن المحتوى المعروض. حتى الآن، تم إنجاز القليل من العمل على استخدام المحتوى النصي

في هذا العمل، نقترح طريقة لتصنيف صفحات الويب بناءً على محتواها النصي. تقدم صفحات الويب عادةً معلومات من فئات مختلفة اعتمادًا على موضوعها المحدد. تعتمد هذه الطريقة على تقنية استخراج الكلمات الرئيسية من صفحة الويب (محتوى نصي) بالإضافة إلى نهج خاضع للإشراف للتعلم الآلي، أي الشبكات العصبية.

الكلمات المفتاحية: تصنيف صفحات الويب، استخراج الكلمات المفتاحية، المناهج الخاضعة للإشراف، التعلم الآلي

Abstract

With the increase in the number of Internet users, the growth of websites is proportional. As a result, the ranking of web pages has become a huge topic of research in recent years. This has made an ever-increasing demand for automated classification techniques with high classification accuracy. To automatically categorize and manipulate web pages, current systems use visual page content, which includes displayed content. However, so far, little work has been done on the use of textual content and HTML code.

In this work, we propose a method of classification of Web pages, based on their textual content. Web pages generally present information of various different classes depending on their specific subject matter. This method is based on the technique of extracting keywords from a web page (textual content) combined with a supervised approach to machine learning, namely neural networks.

Keywords: web page classification, keyword extraction, supervised approaches, machine learning