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

N° d'ordre : SIOD28M2/2023

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Systèmes d'information, Optimisation et Décision (SIOD)**

Optimisation du système de recherche d'informations émotionnelles en utilisant les méthodes d'apprentissage profond

---

Par :

YAGOUB SALSABIL

Soutenu le 20 Juin 2023 devant le jury composé de :

Tibermasine Okba	MCA	Président
Berima Salima	MAA	Rapporteur
Sahli Sihem	MAA	Examinateur

Année universitaire 2022-2023

Résumé

L'analyse de sentiment est le processus qui consiste à déterminer l'opinion, le jugement et l'émotion qui se cache derrière le langage naturel.

L'analyse de sentiment se révèle redoutablement utile lorsque on est face à un grand volume de données textuelles et qu'on doit extraire des informations et les généraliser.

Aujourd'hui, l'analyse des sentiments a une grande importance surtout dans les domaines tel que la politiques, productions et services...etc. Actuellement, les réseaux sociaux pleins des textes dans lesquelles, les internautes s'expriment en différents sujets, l'intérêt de leurs opinion est considérable, où la compréhension du contenu véhiculé par ces textes est un élément essentiel.

La motivation derrière ce travail est d'utilisé l'une des techniques d'exploration de données, à savoir les réseaux de neurones, qui permettent d'analyser et classifier un ensemble de publications dérivées des réseaux sociaux. Les classes que nous avons cibles sont : la classe positive, négative ou neutre.

Nous avons utilisé trois types de fonctions d'activation de réseaux de neurones , puis décidé celles qui donnent les meilleurs résultats. La qualité des résultats obtenus est très convaincante, et conduit à plus de perspectives.

Mots-clés : Analyse des sentiments, Le traitement automatique du langage naturel, Les algorithmes de classification, L'apprentissage automatique ; réseau de neurone.

Abstract

Sentiment analysis is the process of determining the opinion, judgment, and emotion behind natural language.

Sentiment analysis is incredibly useful when faced with a large volume of textual data and you need to extract information from it and generalize it.

Today, sentiment analysis has great importance especially in fields like policies, productions and services...etc . Currently, social networks full of texts in which Internet users express themselves on different subjects, the interest of their opinions is considerable, where understanding the content conveyed by these texts is an essential element.

the motivation behind this work to use one of the techniques of data mining, namely neural networks, which allow to analyze and classify a set of publications derived from social networks. The classes we have defined are : the positive, negative or neutral class.

We used three types of neural network activation functions and then decided which ones give the best results. The quality of the results obtained is very convincing, and leads to more prospects.

Keywords : Sentiment Analysis, Natural Language Processing, Classification Algorithms, Machine Learning ; neural network.

ملخص

تحليل المشاعر هو عملية تحديد الرأي والحكم والعاطفة وراء اللغة الطبيعية.

تحليل المشاعر مفيد بشكل كبير عند مواجهة حجم كبير من البيانات النصية والتي تحتاج إلى استخراج المعلومات منها وتعميمها.

اليوم، تحليل المشاعر له أهمية كبيرة خاصة في مجالات مثل: السياسة والإنتاج والخدمات ... إلخ. حاليًا، الشبكات الاجتماعية مليئة بالنصوص التي يعبر فيها مستخدمو الإنترنت عن أنفسهم حول مواضيع مختلفة، فإن الاهتمام بآرائهم كبير، حيث يعد فهم المحتوى الذي تنقله هذه النصوص عنصرًا أساسيًا.

الدافع وراء هذا العمل هو استخدام إحدى تقنيات التنقيب عن البيانات، وهي الشبكات العصبية، والتي تسمح بتحليل وتصنيف مجموعة من المنشورات المستمدة من الشبكات الاجتماعية. الفئات التي حددناها هي: الفئة الإيجابية أو السلبية أو المحايدة.

استخدمنا ثلاثة أنواع من وظائف تنشيط الشبكة العصبية ثم قررنا أي منها يعطي أفضل النتائج. جودة النتائج التي تم الحصول عليها مقنعة للغاية، وتؤدي إلى المزيد من التطلعات.

الكلمات المفتاحية:

تحليل المشاعر، معالجة اللغة الطبيعية، خوارزميات التصنيف، التعلم الآلي، الشبكة العصبية.

Remerciements

*Tout d'abord, je tiens à remercier **ALLAH** qui m'a donné la force, la volonté et le courage pour terminer ce modeste travail.*

*Ensuite, Mes premiers remerciements vont tout naturellement à **Mme Berima Salima**, qui a accepté d'être mon encadreur, qui m'a guidé, et surtout pour la confiance qu'il ne cesse de me témoigner.*

Je remercie également les membres du jury qui m'a fait l'honneur d'accepter de juger mon mémoire

*Aussi, ne pas oublier de remercier **Mes parent** pour leurs sacrifices, soutien et compréhension durant toutes mes années d'études.*

Dédicaces

Je dédie ce travail

A ma maman **Souad**, celle qui s'est toujours sacrifiée pour me voir réussir, qui m'a soutenu et encouragé durant toutes ces années d'études.

À mon papa **Saadane**, qui ne m'a jamais laissé manquer de quoi que ce soit, qui m'a toujours poussé et motivé dans mes études et ma vie quotidienne.

À mes sœur Nour, Chaima et Imen.

À mon frère Abd errahman.

À mes toutes mes amies.

Et en fin Je le dédie à tous ce qui m'a donné leur moindre coup de pouce pour réussir ce travail

Table des matières

Liste des figures

Liste des tableaux

Introduction général

Depuis le tournant du siècle, comme des millions d'opinions d'utilisateurs sont disponibles sur le web, l'analyse des sentiments est devenue l'un des domaines de recherche les plus fructueux dans le traitement du langage naturel (PNL). La recherche sur l'analyse des sentiments a couvert un large éventail de domaines tels que l'économie, la politique et la médecine, entre autres.

L'analyse des sentiments, aussi appelée exploration des opinions, est le domaine d'étude qui analyse les opinions, les sentiments, les évaluations, les attitudes et les émotions des gens envers les entités et leurs attributs exprimés dans un texte écrit. Les entités peuvent être des produits, des services, des organisations, des individus, des événements, des problèmes ou des sujets.

Au fil des ans, les approches d'analyse des sentiments sont passées de règles simples à des techniques avancées d'apprentissage automatique, comme l'apprentissage profond, qui est devenu une technologie émergente dans de nombreuses tâches de PNL. L'analyse des sentiments n'est pas sans ce succès, et plusieurs systèmes basés sur l'apprentissage profond ont récemment démontré leur supériorité par rapport aux anciennes méthodes, obtenant des résultats de pointe sur des ensembles de données d'analyse des sentiments standards.

La quantité de données disponibles sur le Web augmente de façon exponentielle. Cependant, ces données sont principalement décrites dans un format non structuré et ne peuvent donc pas être traitées que par machine. Par conséquent, les techniques de traitement du langage naturel (PNL) peuvent contribuer à la distillation des connaissances et des opinions à partir de l'énorme quantité d'informations présentes sur le Web.

L'analyse des sentiments peut améliorer les capacités des systèmes de gestion de la relation client et de recommandation, par exemple en aidant à découvrir les caractéristiques qui intéressent particulièrement les clients ou en excluant les éléments qui ont reçu des critiques défavorables des publicités. Nous constatons, à cet effet, que la plupart des applications d'analyse des

sentiments impliquent des réseaux sociaux à l'instar de Twitter, FacTwitter, Facebook.

Pour résoudre ce problème nous avons réaliser modèle capable d'analyser et d'extraire l'opinion d'un commentaire (Positif/Négatif/Neutre). Afin d'atteindre cet objectif nous suivrons le mémoire comme suit :

- Dans le premier chapitre, nous avons présenté les définitions, le traitement du langage naturel et le champ d'application Nous avons vu l'importance et quelqu'un outil.
- Dans le deuxième chapitre, L'analyse des sentiments, nous avons abordé l'histoire de l'analyse de sentiments, et différentes approches et les travaux connexes sans oublier les sources des donnés.
- Dans le troisième chapitre, nous présentons et identifions nos modèles proposés Réseau de neurone.
  - Le principe de fonctionnement.
  - L'algorithme principal.
  - Les différéntes fonctions.
- Les deux dernier chapitres sont destinés à l'évaluation du modèle proposé en Conception. Nous commençons par la description des outils utilisés, ainsi que les expérimentations réalisées et les résultats obtenus.

Nous cloturons ce mémoire par une conclusion général resumant le travail les resultats et les perspective.

Chapitre I

Le traitement du langage naturel

I.1 Introduction

Les gens communiquent de différentes manières : en parlant et en écoutant, faire des gestes, en utilisant des signaux manuels spécialisés (par exemple lors de la conduite ou de la direction), en utilisant des langues des signes pour les sourds, ou à travers diverses formes de texte. Par texte, nous entendons les mots qui sont écrits ou imprimés sur une surface plane (papier, carte, panneaux de signalisation et ainsi de suite) ou affichés sur un écran ou un appareil électronique afin d'être lus par leur destinataire prévu (ou par quiconque passe par là).

Dans ce chapitre, nous allons définir Le traitement du langage naturel et mentionner pourquoi il est important et étudier certains de ses outils.

I.2 Définition

Le traitement automatique des langues (Natural Language Processing ou NLP) est la branche de l'intelligence artificielle qui vise à traiter les langues parlées par les humains. Cela couvre un grand nombre d'applications, comme retranscrire de l'oral en écrit, générer automatiquement une synthèse, traduire un document, chercher de l'information ou encore «Comprendre» ce que veut dire un texte. Dans ce dernier cas, on emploie aussi souvent les termes voisins d'analyse sémantique (semantic analysis) et de fouille de textes (text mining).[1]

I.3 Importance

- Le traitement du langage naturel aide les ordinateurs à communiquer avec les humains dans leur propre langue et met à l'échelle d'autres tâches liées au langage. Par exemple, NLP permet aux ordinateurs de lire du texte, d'entendre la parole, de l'interpréter, de mesurer le sentiment et de déterminer quelles parties sont importantes.
- Les machines d'aujourd'hui peuvent analyser plus de données linguistiques que les humains, sans fatigue et de manière cohérente et impartiale. Compte tenu de la quantité stupéfiante de données non structurées générées chaque jour, des dossiers médicaux aux médias sociaux, l'automatisation sera essentielle pour analyser pleinement les données textuelles et vocales de manière efficace. [2]
- Vous pouvez également intégrer le traitement du langage naturel (NLP) dans les applications orientées clientes pour communiquer plus efficacement avec les clients. Par exemple, un chatbot analyse et trie les requêtes des clients, répond automatiquement aux questions fréquemment posées et transmet les requêtes complexes au support client. Cette automatisation permet de réduire les coûts, de faire gagner du temps aux agents sur les demandes fréquentes et d'améliorer la satisfaction des clients. [3]

I.4 Outils

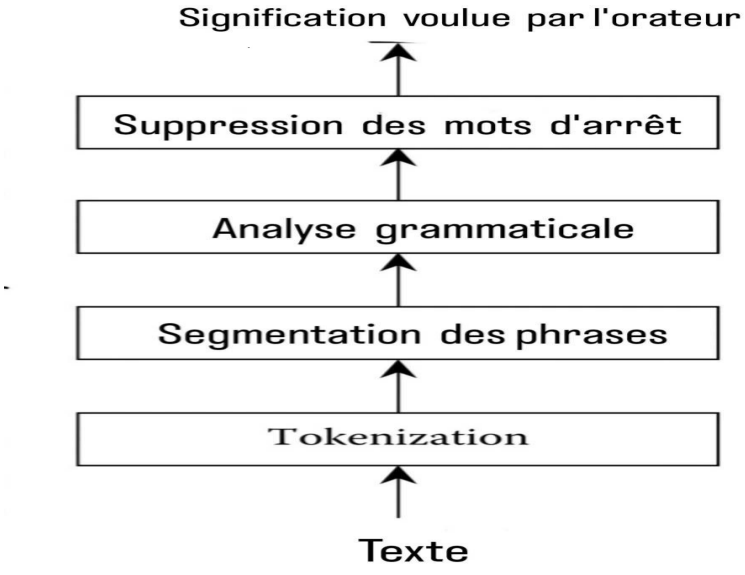


FIGURE I.1 – Les étapes de l'Analyse dans le traitement naturel

I.4.1 La tokenisation

Cette étape permet de décomposer une chaîne de caractères (message ou commentaire) en mots appelés « Tokens ». La tokenisation est encore plus importante dans l'analyse des sentiments que dans d'autres domaines de la NLP, car les informations sur les sentiments sont souvent mal représentées.

Exemple :

Aujourd'hui il fait beau.

Peut-être tokenisé comme dans l'exemple suivant :

' Aujourd'hui ' ' il ' ' fait ' ' beau '. [4]

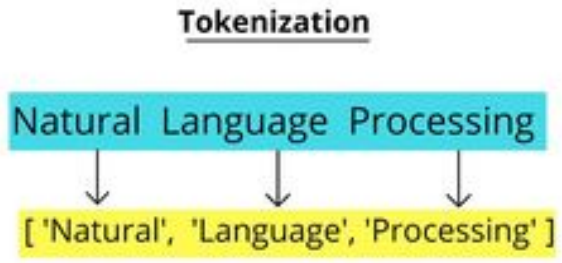


FIGURE I.2 – Figure montrant la tokenization.

I.4.2 Segmentation des phrases

La segmentation des phrases est le processus de détermination des unités de traitement plus longues composées d'une ou de plus de mots. Cette tâche consiste à identifier les limites de phrase entre les mots de différentes phrases. Étant donné que la plupart des langues écrites ont des signes de ponctuation qui se produisent aux limites des phrases, la segmentation des phrases est souvent appelée détection des limites de phrase, désambiguïsation des limites de phrase, ou la reconnaissance des limites de la peine.

Tous ces termes désignent la même tâche : déterminer comment un texte doit être divisé en phrases pour un traitement ultérieur. [5]

I.4.3 Analyses grammaticales

L'analyse grammaticale en anglais (Part Of Speech (POS)) consiste à faire connaître l'espèce ou la nature de chacun des mots dont une phrase se compose, et à expliquer leurs formes ou modifications, ainsi que le rôle ou la fonction qu'ils remplissent dans cette phrase.[6]

Nom : poisson, livre, maison, enclos, procrastination, langue.

Verbe : aime, déteste, étudie, dort, pense, est.

Adjectif : grincheux, somnolent, heureux, timide.

Adverbe : lentement, vite, maintenant, ici, là.

Pronom : Je, toi, il, elle, nous, nous, il, ils.

Préposition : dans, sur, à, par, autour, avec, sans.

Conjonction : et, mais, ou, sauf. [7]

I.4.4 Suppressions des mots d'arrêt

Un mot d'arrêt est un mot inutile et non significatif apparaissant dans un texte, d'où la nécessité de l'éliminer de notre corpus. A cet effet, pour des langues telles que l'Anglais, le Français ou l'Arabe standard moderne, il existe des listes de mots d'arrêt bien connus. Ces listes/outils sont disponibles gratuitement comme NLTK. Néanmoins, il n'y a pas de ressource définie ou élaborée pour les mots vides d'AlgD (Dialectes algériens) à considérer d'où l'obligation de les créer. [4] Exemple de ces mots d'arrêt :

	Les mots d'arrêt							
Français	A	Bien	Ce	Clic	Des	En	Ect	Mais
	celui	cher	delà	devra	Fais	font	huit	Leurs
	mien	mon	ouf	nos	passé	peu	rien	sans
English	About	And	Been	Can	Does	Exactly	For	Gone
	became	best	cause	could	dare	each	else	few
	five	given	hade	help	him	if	into	just
Arabic	في	هناك	ولم	فيها	الا	بين	ذلك	حين
	الوقت	مع	واحد	لدى	كانت	عدد	اول	بسبب
	حتى	غدا	الى	صباح	ثم	صفر	زيارة	او

TABLE I.1 – Exemple des mots d'arrêt.

I.5 Application

Le traitement du langage naturel peut être appliqué dans divers domaines comme :

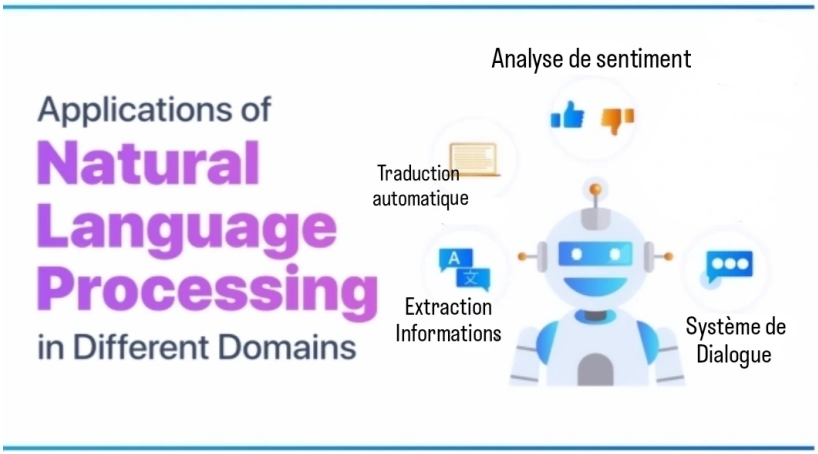


FIGURE I.3 – Les applications de NLP dans différents domaines.

I.5.1 Traduction automatique

Comme la majeure partie du monde est en ligne, la tâche de rendre les données accessibles et disponibles pour tous est un défi. Ce dernier est la barrière de la langue. Il existe une multitude de langues avec une structure de phrase et une grammaire différente. La traduction automatique traduit généralement des phrases d'une langue à une autre à l'aide d'un moteur statistique comme Google Translate. Le défi avec les technologies de traduction automatique n'est pas de traduire directement les mots, mais de conserver le sens des phrases intactes ainsi que la grammaire et les temps. [8]

I.5.2 Extraction informations

L'extraction d'informations concerne identification des phrases d'intérêt des données textuelles. Pour de nombreuses applications, l'extraction d'entités telles que les noms, les lieux, les événements, les dates, les heures et les prix sont un moyen puissant de résumer les informations pertinentes aux besoins d'un utilisateur. Dans le cas d'un moteur de recherche spécifique à un domaine, l'identification automatique d'informations importantes peuvent augmenter la précision et l'efficacité d'une recherche dirigée. Des modèles de Markov cachés (HMM) sont utilisés pour extraire les domaines pertinents des articles de recherche. Ces segments de texte extrait sont utilisés pour permettre la recherche dans des domaines spécifiques et pour fournir une présentation efficace des résultats de la recherche et pour faire correspondre les références aux articles. Par exemple, remarquer les publicités contextuelles sur tous les sites Web affichant les articles récents que vous avez peut-être consultés sur une boutique en ligne avec des remises. [8]

I.5.3 Système de dialogue

Peut-être l'application omniprésente du futur, dans les systèmes envisagés par les grands fournisseurs d'applications pour les utilisateurs finaux. Systèmes de dialogue, qui se concentrent généralement sur une application étroitement définie (par exemple, votre réfrigérateur ou le son de la maison), utilisent actuellement les niveaux phonétique et lexical de la langue. On croit que l'utilisation de tous les niveaux de traitement linguistique expliqués ci-dessus offre le potentiel pour des systèmes de dialogue vraiment habitables.

Des exemples tels que l'assistant de Google, Windows Cortana, Siri d'Apple et Alexa,

Amazon sont les logiciels et les appareils qui suivent les systèmes Dialogue. [9]

I.5.4 Analyses des sentiments

L'analyse des sentiments, qu'on appelle aussi la fouille d'opinions, est l'un des domaines de recherche les plus actifs dans le traitement du langage naturel depuis le début des années 2000.

Le but de l'analyse des sentiments est de définir des outils automatiques capables d'extraire des informations subjectives à partir du texte en langage naturel, telles que les opinions et les sentiments, afin de créer des connaissances structurées et utilisables par un système d'aide à la décision ou par un décideur. [10]

FIGURE I.4 – À la découverte de l'analyse de sentiment

I.5.4.1 Les niveaux des analyses de sentiments

Il existe trois sous-catégories de méthodologie de base pour analyser les sentiments. Chacune répond à un objectif différent. Il convient donc de choisir parmi ces 3 méthodes celle qui correspond le plus au contexte d'utilisation et au type de résultat souhaité.

- **Facile :**

  Détecter si l'appréciation est positive ou négative par exemple : « Le restaurant est situé au centre-ville » « Cette imprimante est très coûteuse » La première phrase exprime une émotion positive, et le deuxième une émotion négative.

- **Moyen :**

  Il existe de nombreux termes intermédiaires que les utilisateurs utilisent dans leurs commentaires et dialogues, tels que « Pas très mal » ou « Très bon ». Ces termes se réfèrent à l'émotion moyenne, c'est-à-dire à la polarité moyenne, donc on ajouté

des niveaux supplémentaires à cette catégorisation comme « Très positif » ou « Très négatif». [11]

- **Difficile :**

  Parfois un texte contient plusieurs opinions ou des appréciations mitigées ou ambivalentes. C'est par exemple le cas de cette phrase : « Le personnel était très sympathique mais nous avons attendu trop longtemps avant d'être servis. ». Être capable d'évaluer et rendre compte de la polarité permet de noter quand il y a à la fois des sentiments positifs et négatifs dans un feedback. Cela permet d'éviter que le commentaire soit faussement classé en « Neutre ». [12]

I.6 Conclusion

Bien que NLP s'agisse d'un domaine de recherche et d'application relativement récent par rapport à d'autres, il y a eu suffisamment de succès à ce jour pour que suggèrent que les technologies d'accès à l'information fondées sur la NLP continueront d'être un domaine important de la recherche et du développement dans les systèmes d'information aujourd'hui et dans un avenir lointain.

Dans le chapitre suivant, nous approfondirons l'étude de l'analyse des sentiments.

Chapitre II

L'analyse des sentiments

II.1 Introduction

L'Internet contient un nombre énorme d'informations, et pour la plupart d'entre nous c'est le premier lieu pour trouver ces informations, réserver l'avion ou l'hôtel, acheter des produits, consulter les avis d'autres utilisateurs sur les produits qui nous intéressent, lire les commentaires avant de choisir le film à voir au cinéma, voir des propositions d'autres personnes avant de choisir les cadeaux de mariage etc.

L'analyse des sentiments est un domaine devient de plus en plus important car de plus en plus les gens aiment acheter en ligne et donner des commentaires, des critiques sur les produits ou services, où il a été utilisé dans des applications divers.

Dans ce chapitre nous allons parler de tout ce qui concerne l'analyse des sentiments à partir de son histoire et de sa définition, nous parlerons de son motivation et mentionnerons ses sources sans oublié les approches, les défis et les travaux antérieur.

II.2 Définition

Dans la littérature, l'analyse de sentiments (SA) reçoit différentes dénominations ou termes communs, on trouve entre autres l'extraction d'opinions (opinion mining OM), l'analyse de subjectivité, l'analyse des émotions et l'extraction de l'évaluation et autres. Les plus souvent utilisées dans la littérature sont l'analyse des sentiments et l'exploration d'opinions (MO).
Bing Liu à donner une définition à cette tâche : "L'analyse du sentiment, aussi appelée opinion mining, est le domaine d'étude qui analyse les opinions, les sentiments, les évaluations, les

attitudes et les émotions des gens envers les entités et leurs attributs exprimés dans un texte écrit." [13]

D'une autre façon, l'analyse du sentiment concerne le traitement d'un texte d'opinion pour extraire la polarité des sentiments généralement exprimés en termes d'opinion positive ou négative (classification binaire), ou d'une classification multiple, où le sentiment peut avoir une étiquette neutre ou même d'autres étiquettes différentes comme très positive, positive, neutre, négative, très négative, les étiquettes peuvent aussi être associées à des émotions comme la tristesse, la colère, le bonheur, etc. Et même des valeurs numériques. [14]

II.3 Historique

- Des philosophes tels que Leibniz et Descartes ont fait des propositions pour des symboles qui relieraient les mots entre les langues. Toutes ces propositions sont restées en suspens, et aucun d'entre eux n'a abouti à la création d'une vraie machine.
- Depuis les années trente, il y a eu des idées et des conceptions pour une machine de traduction, mais il n'y en avait rien avec une réelle efficacité.
- La soi-disant « Georgetown Experience » a été créée en collaboration avec IBM en 1954 avec une traduction entièrement automatique de plus de soixante phrases. Du russe vers l'anglais. Les auteurs du programme ont affirmé que dans trois ou cinq ans, le problème de la traduction automatique sera résolu.
- En 1968, le programme SHRDLU a été créé par l'ingénieur américain Terry Winograd au MIT, qui a réussi à Établir un dialogue avec l'ordinateur.
- En 1991, un modèle de « psychothérapeute intelligent » a été réalisé, basé sur l'idée d'un chatbot et travaillait sur DOS, et commence en gros :

  HELLO [UserName], MY NAME IS DOCTOR SBAITSO.
  I AM HERE TO HELP YOU. SAY WHATEVER IS IN YOUR MIND FREELY, OUR CONVERSATION WILL BE KEPT IN STRICT CONFIDENCE. MEMORY CONTENTS WILL BE WIPED OFF AFTER YOU LEAVE,
  SO, TELL ME ABOUT YOUR PROBLEMS
- En 2006, la première version du programme géant Watson Dora IBM a été publiée, qui a réussi à exceller Sur les humains en répondant aux questions du célèbre programme américain : Geobardi.

- Lancement de Personal Assistant : Siri 2011, Elisca et Cortana en 2014, Google Assistant en 2016 et Samsung Pixbay en 2017. [15]

II.4 Motivation

La gestion de contenu axée sur l'opinion possède de nombreuses applications importantes comme la détermination des opinions des critiques concernant un certain produit via la classification des critiques de produits en ligne ou l'enregistrement de l'évolution des attitudes du public à l'égard d'un parti politique via l'extraction de sites d'actualités en ligne ou de contenu de blogs. Alors que les applications basées sur l'opinion ou sur le feedback sont plus populaires, le domaine du traitement du langage naturel s'intéresse actuellement aux analyses de sentiments ainsi qu'aux systèmes d'exploration d'opinion. Les principales applications des analyses de sentiments et de l'extraction d'opinions sont données ci-dessous : [16]

Achat de produits ou de services : lorsque vous décidez d'acheter des produits ou des services, prendre des décisions précises n'est plus un travail difficile. Grâce à cette méthode. Les individus peuvent évaluer les opinions et les expériences des autres concernant tous les produits et services et comparer les marques concurrentes.

Amélioration de la qualité du produit ou du service : grâce à l'exploration d'opinions et à l'analyse des sentiments, les fabricants peuvent recueillir les opinions des critiques ainsi que les critiques positives concernant leurs produits ou services et ainsi améliorer la qualité de leurs services.

Recherche marketing : les résultats des analyses de sentiment peuvent être utilisés à des fins d'étude de marché. Grâce à des méthodes d'analyse des sentiments, les tendances récentes des clients concernant des produits ou services particuliers peuvent être examinés. De même, les attitudes actuelles du public à l'égard des nouvelles politiques de l'État peuvent également être facilement examinées.

Détection de flamme : la supervision des sites d'informations, des articles de blogs ainsi que des réseaux sociaux est simplifiés grâce à des analyses de sentiment. L'exploration d'opinions et l'analyse des sentiments sont capables de détecter les mots arrogants, surchauffés, incitant à la haine, jurons utilisés dans les e-mails, les blogs ou les sites d'informations de manière automatisée.

Élaboration de politiques : en utilisant des analyses de sentiment, les décideurs politiques

sont en mesure de prendre en considération les perspectives des citoyens concernant certaines politiques et ces connaissances peuvent être utilisées pour créer de nouvelles politiques en faveur des citoyens.

Systèmes de recommandation : grâce à la classification des opinions des individus comme positives ou négatives, le système peut déterminer laquelle est recommandé et laquelle ne l'est pas.

Détection d'opinion Spam : comme Internet est également accessible à tout le monde, n'importe qui peut télécharger n'importe quoi sur Internet. Cela signifie la probabilité que le contenu soit du spam augmenté de jour en jour. Les particuliers pourraient télécharger du contenu de spam dans le but d'induire les gens en erreur. L'exploration d'opinions ainsi que les analyses de sentiment sont capables de classer le contenu Internet en spam et en non-spam.

II.5 Source des données

Avec la croissance des sites des réseaux sociaux en ligne, par exemple, les forums, les sites de critiques, les blogs et les micros-blogs, l'enthousiasme pour l'extraction d'opinions s'est considérablement développé. Aujourd'hui, les sentiments en ligne se sont transformés en une sorte de profit virtuel pour les entreprises cherchant à commercialiser leurs produits, à reconnaître les nouvelles tendances et à gérer leur position. De nombreuses organisations utilisent actuellement des systèmes d'analyse de sentiments et d'extraction d'opinions pour suivre les entrées des clients dans les sites de vente en ligne et les sites d'évaluation. [14]

II.5.1 Examiner les sites

Donner aux utilisateurs de générer des avis sur les produits et services qu'ils ont achetés est une pratique largement disponible sur Internet. Les données utilisateurs sont analysées pour la classification de sentiment recueillis à partir des sites comme www.gsmarena.com (avis mobiles), www.amazon.com (commentaires sur les produits), www.CNET download.com (avis sur les produits), qui accueille des millions de commentaires sur les produits par les consommateurs. [17]

II.5.2 Micro-blogging

Un outil de communication très populaire parmi les utilisateurs d'Internet est de microblogging. Des millions de messages apparaissent tous les jours dans les sites Web populaires pour microblogging comme Twitter, utilisés comme sources des données pour la classification des sentiments. [18]

II.5.3 Blogs

Le nom associé à l'univers de tous les sites de blog est appelé la blogosphère. Les gens écrivent sur les sujets qu'ils veulent partager avec les autres sur un blog. Blogueur est une chose qui se passe en raison de sa facilité et la simplicité de la création de blog, sa forme libre et la nature inédite. Nous trouvons un grand nombre de poste sur presque tous les sujets d'intérêt sur la blogosphère. [19]

II.6 Approche

Les techniques de classification des sentiments peuvent être grossièrement divisées en une approche d'apprentissage automatique, une approche basée sur le lexique et une approche hybride.

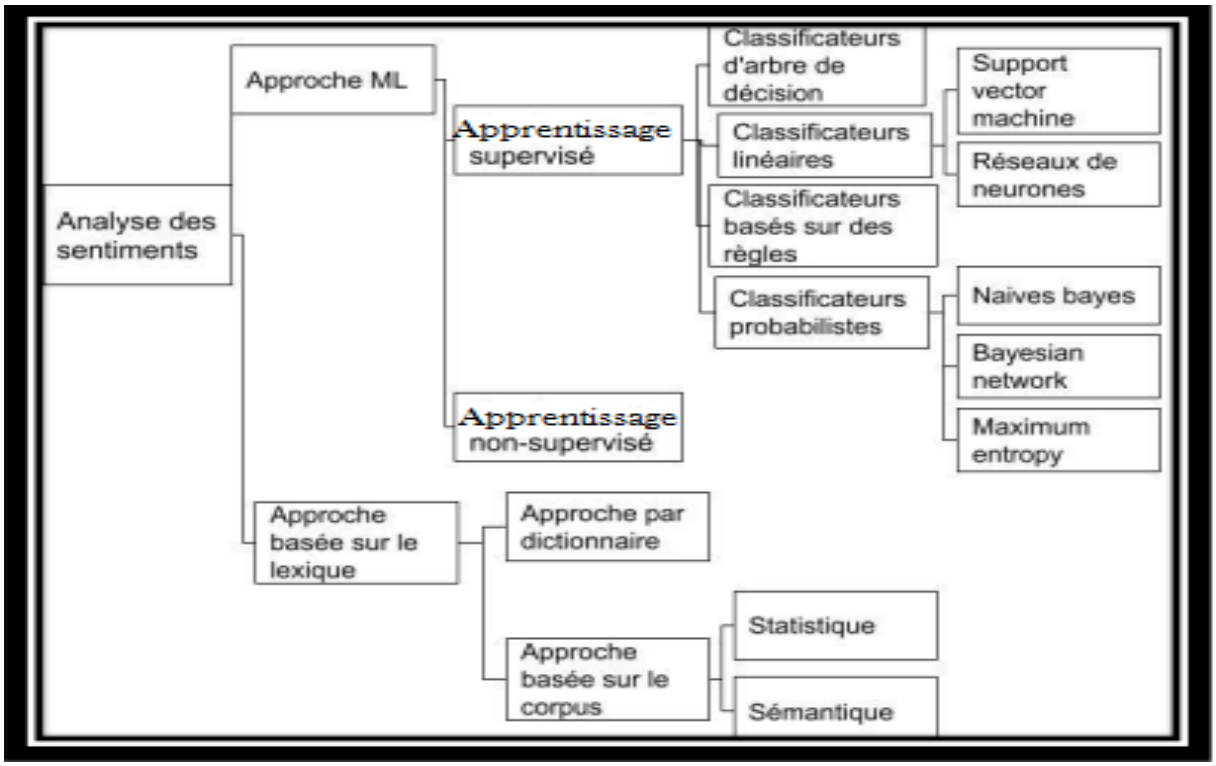


FIGURE II.1 – Hiérarchie des approches de classification des sentiments.

II.6.1 Approche basée sur l'apprentissage automatique

L'apprentissage automatique a été introduite par l'informaticien chercheur Arthur Samual en 1958, Il l'a défini comme étant le domaine d'études qui donne aux ordinateurs la possibilité d'apprendre sans être explicitement programmé. [20]

L'apprentissage automatique (Machine Learning), est l'une des tâches les plus étudiées de nos jours, Il est une partie très importante de l'intelligence Artificielle et doit être une des principales caractéristiques des systèmes intelligents. Par apprentissage, nous pouvons exploiter et construire des modèles de la réalité en se basant sur des expériences, soit en créant un modèle complètement, soit en modernisant un modèle partiellement construit. [21]

L'apprentissage automatique (ou apprentissage artificiel) est, l'étude des algorithmes et des méthodes qui permettent aux programmes de s'améliorer automatiquement par expérience.

On distingue ainsi trois types d'apprentissage :

- L'apprentissage semi-supervisé.
- Apprentissage supervisé.
- Apprentissage non supervisé.

Apprentissage supervisé : La classification supervisée consiste à partir de la description

de l'élément détermine sa classe avec le plus faible taux d'erreurs. La performance de la classification dépend notamment de l'efficacité de la description. De plus, si l'on veut obtenir un système d'apprentissage, la procédure de classification doit permettre de classer efficacement tout nouvel exemple (pouvoir prédictif).

Algorithmes de classification supervisée : Il existe de nombreuses méthodes d'apprentissage supervisé. [22]

- Les machines à support de vecteurs (SVM) :

  Les « Supports Vectors Machines » appelés aussi « Maximum Margin Classifier» Les SVMs sont les classificateurs d'apprentissage automatique largement utilisé pour la catégorisation de texte, qui ont été proposés pour la première fois par Vapnik en 1990. Les SVMs sont des techniques d'apprentissage supervisé qui reposent sur deux notions principales : la notion de marge maximale et la notion de fonction noyau. L'objectif principal des SVM est de déterminer dans l'espace de recherche des séparateurs appelés hyperplans, qui peuvent séparer au mieux les différentes classes.

  Dans le cas où le problème est linéairement séparable, le choix de l'hyperplan séparateur n'est pas évident. Il existe en effet une infinité d'hyperplans capables de séparer parfaitement les deux classes d'exemples. Le principe des SVMs est de choisir celui qui va maximiser la distance minimale entre l'hyperplan et les exemples d'apprentissage. Vapnik a montré qu'il existe un unique hyperplan optimal, défini comme l'hyperplan qui maximise la marge (la marge est la distance entre la frontière de séparation et les échantillons les plus proches) entre les échantillons et l'hyperplan séparateur.

  Afin de pouvoir traiter des cas où les données ne sont pas linéairement séparables, l'idée clé des SVMs est de transformer l'espace de représentation des données d'entrées en un espace de plus grande dimension, dans lequel il est probable qu'il existe une séparation linéaire grâce à des fonctions noyaux qui permettent de transformer un produit scalaire dans un espace de grande dimension. [22]

  Les SVMs sont utilisés dans de nombreuses applications, parmi lesquelles celle de qui a effectué une analyse des sentiments sur les Tweets à l'aide d'un SVM et a atteint une précision d'environ 60% en fonction des fonctionnalités utilisées. De plus, ils ont prétraité les Tweets en remplaçant les acronymes par leur pleine signification et en remplaçant les émoticônes par leur état émotionnel. Il convient de noter que la même méthode a atteint des précisions d'environ 75% lors de la conduite de la classification

binaire, ignorant la classe neutre et n'ayant que des classes positives ou négatives. [14]

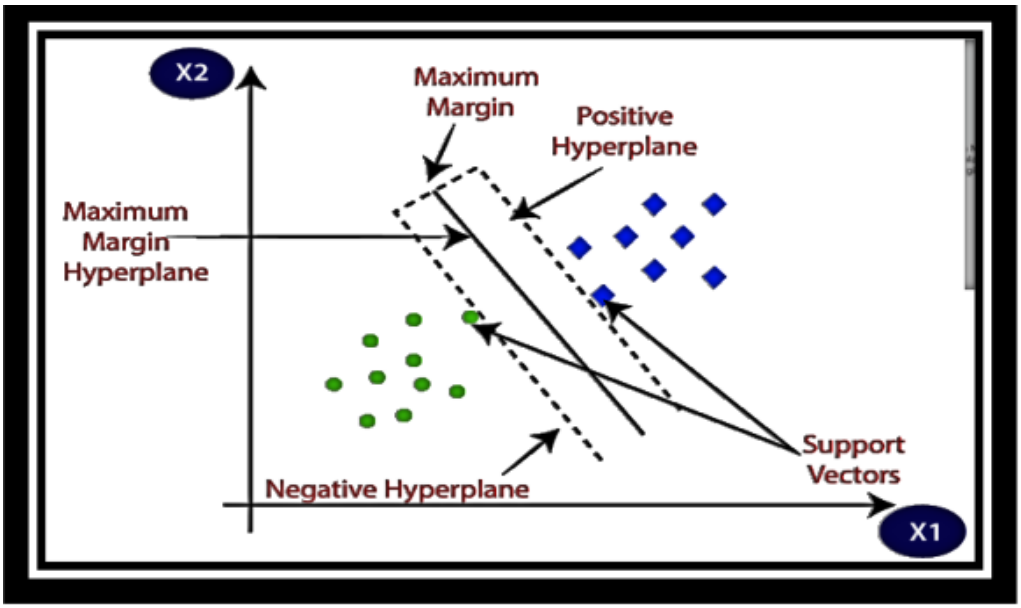


FIGURE II.2 – Le principe de SVM.

- Naïve Bayes

  Les classificateurs bayésiens sont les classificateurs les plus simples en apprentissage supervisé basés sur le théorème de Bayes. Ils peuvent prédire la classe probabilités d'appartenance, telles que la probabilité qu'un échantillon donné appartient à une classe particulière. Les classificateurs supposent que l'effet d'une valeur d'attribut sur une classe donnée est indépendant des valeurs des autres attributs. Cette hypothèse est appelée indépendance conditionnelle de classe.

  Il est fait pour simplifier le calcul impliqué et, en ce sens, est considéré comme «naïf». Soit X = x1, x2,. . . , xn être un échantillon, dont les composants représentent les valeurs faites sur un ensemble de n attributs. En termes bayésiens, X est considéré l'échantillon de données observé. Soit H une hypothèse telle que les données X appartiennent à une classe spécifique C. Pour les problèmes de classification, notre objectif est de déterminer P(H|X), la probabilité que l'hypothèse H soit vérifiée étant donné les l'échantillon de données observé X. En d'autres mots, nous recherchons la probabilité d'appartenance de l'échantillon X à la classe C, étant donné que nous connaissons la description des attributs de X. Selon le théorème de Bayes, la probabilité que nous voulons calculer P(H|X) peut être exprimé en termes de probabilités P(H),

P(X|H) et P(X) comme suit :

$$P(H|X) = \frac{P(H|X)P(H)}{P(X)} \tag{II.1}$$

P (H|X) représente la probabilité postérieure de H conditionnée à X, c'est-à-dire la probabilité qu'une hypothèse se vérifie étant donné la valeur de X, P(H) représente la probabilité antérieure de H, c'est-à-dire la probabilité que H est vrai indépendamment des valeurs de l'échantillon, P(X|H) représente la probabilité postérieure de X conditionnée à H, c'est-à-dire la probabilité que X aura certaines valeurs pour une hypothèse donnée , P(X) représente la probabilité antérieure de X, c'est-à-dire la probabilité que X aura certaines valeurs. [14]

II.6.2 L'approche basée sur le lexique

Repose sur un lexique des sentiments, une collection de termes de sentiments connus et précompilés. Il est divisé en une approche basée sur un dictionnaire et une approche basée sur un corpus qui utilisent des méthodes statistiques ou sémantiques pour trouver la polarité des sentiments.

Les mots d'opinion sont utilisés dans de nombreuses tâches de classification des sentiments. Les mots d'opinion positifs sont utilisés pour exprimer certains états souhaités, tandis que les mots d'opinion négatifs sont utilisés pour exprimer certains états indésirables. [23]

- Méthode basée dictionnaire

  A présenté la stratégie principale de l'approche par dictionnaire. Un petit ensemble de mots d'opinion est collecté manuellement avec des orientations connues. Ensuite, cet ensemble est développé en recherchant dans les corpus bien connus WordNet ou thésaurus leurs synonymes et antonymes. L'approche par dictionnaire présente un inconvénient majeur qui est l'incapacité à trouver des mots d'opinion avec des orientations spécifiques au domaine et au contexte. [22]

II.6.3 L'approche hybride

Cette approche tire profit des deux méthodes précédentes il y a trois façon de faire. La première est d'exploiter les outils linguistiques pour élaborer le corpus puis classer les textes par un outil d'apprentissage supervisé. La deuxième façon est d'utiliser l'apprentissage

automatique pour établir le corpus d'opinion nécessaire à l'approche basée sur lexique. La troisième façon est la combinaison des deux approches précédentes et le conjointement de leurs résultats. [24]

II.7 Défis de l'analyse des sentiments

La détection automatique d'un sentiment dans un texte présente plusieurs défis : Complexité et subtilité de l'utilisation de la langue : [25]

- Certains termes comme les négations et les modaux impactent le sentiment de la phrase, sans eux-mêmes ayant de fortes associations sentimentales. Par exemple : peut être bon, était bon, et n'était pas bon doit être interprété différemment par les systèmes d'analyse des sentiments.
- Les énoncés peuvent véhiculer plus d'une émotion (et à des degrés divers). Ils peuvent transmettre des évaluations contrastées de plusieurs entités cibles.
- Les énoncés peuvent faire référence à des événements émotionnels sans exprimer implicitement ou explicitement les sentiments du locuteur.

Utilisation d'un langage créatif et non standard :

- Les systèmes automatiques de langage naturel ont du mal à interpréter les utilisations créatives de la langue comme le sarcasme, l'ironie, l'humour et la métaphore. Cependant, ces phénomènes sont courants dans l'utilisation de la langue.
- Les textes des médias sociaux regorgent de termes qui ne figurent pas dans les dictionnaires, tels que les fautes d'orthographe (parlament), mots orthographiés de manière créative (happeee), mots hashtaggés (#loveumom), émoticones, abréviations (lmao), etc. Beaucoup de ces termes véhiculent des émotions.

Manque de grandes quantités de données étiquetées :

- La plupart des algorithmes d'apprentissage automatique pour l'analyse des sentiments nécessitent des quantités importantes de données d'entraînement (exemples de phrases marquées avec les émotions associées). Cependant, il existe de nombreuses catégories d'affects, y compris des centaines d'émotions que les humains peuvent percevoir et exprimer. Ainsi, une grande partie du travail dans la communauté a été limitée à une poignée d'émotions et de catégories.

II.8 Travaux antérieur

Mishra et ses collègues ont proposé un système basé sur le support Machine vectorielle (SVM) pour détecter la polarité des critiques de médicaments. Le système a également effectué une analyse des sentiments basée sur les aspects du médicament des examens pour prédire les cotes pour certaines conditions telles que la satisfaction, efcacité et facilité d'utilisation du médicament. Les examens de médicaments ont été symbolisés. Ensuite, SentiWordNet a été utilisé pour attribuer les scores de sentiment pour chaque jeton. [26]

Cette étude présente une vue d'ensemble des méthodes d'analyse de sentiment, des outils et des techniques de traitement de texte utilisés pour extraire et classer les opinions des utilisateurs en fonction de leur positivité ou négativité. [27]

les études mentionné dans référence [28] explore les techniques d'analyse de sentiment en examinant les aspects lexicaux, statistiques et de machine learning de l'analyse de sentiment. Les auteurs présentent également des applications pratiques de l'analyse de sentiment, telles que l'analyse de critiques de films et de produits.

Cette recherche décrit les différentes techniques d'analyse de sentiment et d'émotion, en se concentrant sur l'utilisation de la linguistique computationnelle et des techniques d'apprentissage automatique. Les auteurs présentent également des applications de l'analyse de sentiment, telles que la surveillance de la réputation en ligne et la détection d'émotions dans les médias sociaux. [29]

II.9 Conclusion

Au cours de ce chapitre,nous avons exposé les fondmentans de l'A.S ansi que son importance, sans oublier les sources des donnés. Nous sommes intéressés aux différentes approches de l'analyse des sentiments, et nous avons aussi parlé des difcultés rencontrées dans de domaine. Enfn, nous avons abordé les différents travaux antérieurs effectués sur l'analyse des sentiments.

Dans le chapitre suivant nous étudierons tout ses qui sont liés à réseau des neurones.

Chapitre III

Réseaux de neurones

III.1 Introduction

Ces dernières années, les réseaux de neurones sont devenus une sorte de nouvelle électricité -une technologie révolutionnaire qui a pénétré dans toutes les sphères de l'activité humaine. Et ce n'est pas surprenant, car les solutions technologiques basées sur les réseaux de neurones peuvent effectuer un éventail de tâches extrêmement large - du traitement des maladies les plus complexes.

Dans ce chapitre On va parler de l'histoire des Réseaux de neurones. On va ainsi expliquer en détail son pincipe, sans oublier ses types et ses fonctions activités. Enfin, nous allons identifié ses améliorations et ses inconvénients.

III.2 Historique

Les premiers ont proposé un modèle préliminaire de ce qu'on appelle maintenant réseaux de neurones sont deux bio-physiciens de Chicago, McCulloch et Pitts, qui inventent en 1943 le premier neurone formel qui portera leurs noms (neurone de McCulloch-Pitts).

Quelques années plus tard, en 1949, Hebb propose une formulation du mécanisme d'apprentissage, sous la forme d'une règle de modification des connexions synaptiques (règle de Hebb). Cette règle, basée sur des données biologiques, modélise le fait que si des neurones, de part et d'autre d'une synapse, sont activés de façon synchrone et répétée, la force de la connexion synaptique va aller croissant.

Le premier réseau de neurones artificiels apparait en 1958, grâce aux travaux de Rosenblatt

qui conçoit le fameux Perceptron. Le Perceptron est inspiré du système visuel (en terme d'architeture neuro-biologique) et possède une couche de neurones d'entrée (”perceptive”) ainsi qu'une couche de neurones de sortie (”décisionelle”). Ce réseau parvient à apprendre à identifier des formes simples et à calculer certaines fonctions logiques.

Malgré tout l'enthousiasme que soulève le travail de Rosenblatt dans le début des années 60, la fin de cette décennie sera marquée en 1969, par une critique violente du Perceptron par Minsky et Papert. Ils montrent dans un livre (« Perceptrons ») toutes les limites de ce modèle, et soulèvent particulièrement l'incapacité du Perceptron à résoudre les problèmes non linéairement séparables, tels que le célèbre problème du XOR (OU exclusif). Il s'en suivra alors, face à la déception, une période noire d'une quinzaine d'années dans le domaine des réseaux de neurones artificiels.

Il faudra attendre le début des années 80 et le génie de Hopfield pour que l'intérêt pour ce domaine soit de nouveau présent. En effet, Hopfield démontre en 1982 tout l'intérêt d'utiliser des réseaux récurrents (dits ”feed-back”) pour la compréhension et la modélisation des processus mnésiques. Les réseaux récurrents constituent alors la deuxième grande classe de réseaux de neurones, avec les réseaux type perceptron (dits ”feed-forward”).

En parallèle des travaux de Hopfield, Werbos conçoit son algorithme de rétropropagation de l'erreur, qui offre un mécanisme d'apprentissage pour les réseaux multi-couches de type Perceptron (appelés MLP pour Multi-layer Perceptron), fournissant ainsi un moyen simple d'entraîner les neurones des couches cachées. Cet algorithme de ”back-propagation” ne sera pourtant popularisé qu'en 1986 par Rumelhart. [30]

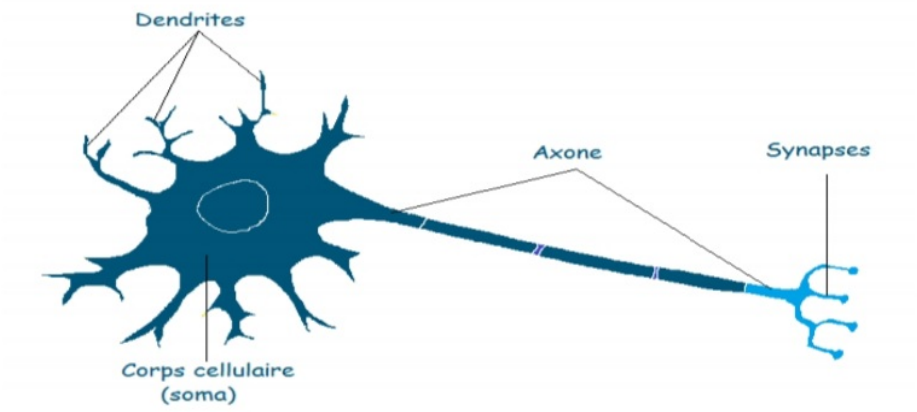


FIGURE III.1 – Neurone biologique.

III.3 Définition

Les réseaux de neurones artificiels (RNA) sont inspirés de la méthode de travail du cerveau humain qui est totalement différent de celle d'un ordinateur. Le cerveau humain se base sur un système de traitement d'information parallèle et non linéaire, très compliqué, ce qui lui permet d'organiser ses composants pour traiter, d'une façon très performante et très rapide, des problèmes très compliqués tels que la reconnaissance des formes. [31]

Un réseau neuronal artificiel (multicouche) est constitué de plusieurs couches de neurones, notamment : une couche d'entrée, une couche masquée et une couche de sortie :

1. Couche d'entrée : la première couche d'un réseau neuronal est la couche de mise, qui est composée d'entrer des neurones et recevoir des données.
2. Couche cachée : cette couche (ou ces couches) se trouve entre les couches d'entrée et de sortie, et c'est dans ces couches que les problèmes sont abordés. Le nombre des couches masquées utilisé dépend de la complexité du problème.
3. Couche de sortie : il s'agit de la dernière couche d'un réseau neuronal artificiel, il produit des extrants du programme.

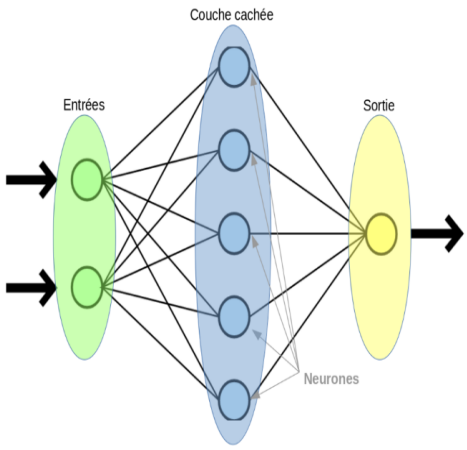


FIGURE III.2 – Les réseaux de neurones artificiels.

III.4 Les fonctions d'activations

La fonction d'activation est une formule mathématique qui aide le neurone à s'allumer et à s'éteindre. Contrairement aux neurones biologiques, qui ont une activation binaire, la fonction d'activation est utilisé pour incorporer la non-linéarité dans le fonctionnement du neurone artificiel.

Il existe différents types de fonctions d'activation, chacune étant utilisée dans une situation :

- Relu Function : l'équation suivante est l'équation de relu function A(x) = max(0,x)
  La fonction d'activation L'unité linéaire rectifiée (ReLU) est une fonction linéaire qui est la fonction d'activation la plus utilisée et populaire. Sa fonctionnalité est basée sur la fait qu'il substitue toute valeur d'entrée négative par 0 tout en laissant la valeur positive inchangé.

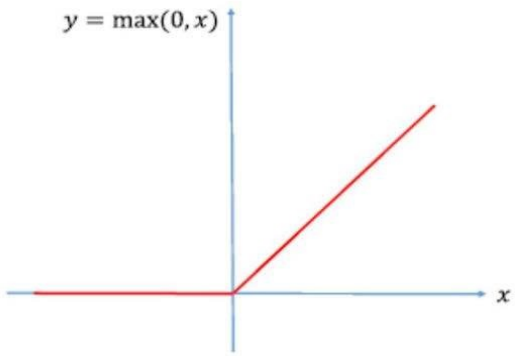


FIGURE III.3 – Fonction relu.

- Fonction sigmoïde : Définie par l'équation suivante :

$$f(x) = \frac{1}{1 - e^{-x}} \tag{III.1}$$

  Cette fonction d'activation a deux valeurs possibles : 0 ou 1, et sa courbe est dans la forme d'un S. Il est employé dans la couche de sortie lorsque nous avons un problème de classification binaire.

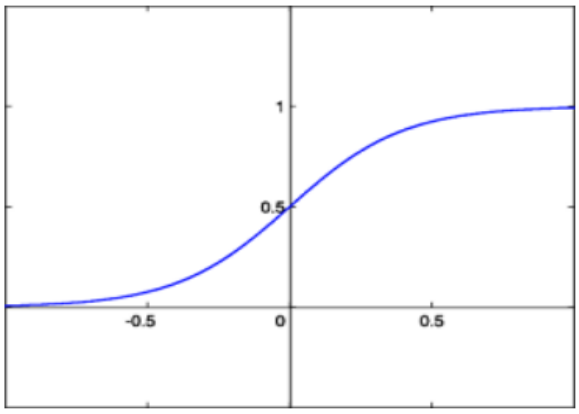


FIGURE III.4 – Fonction sigmoïde.

- Courbe de fonction Tanh : Définie par l'équation suivante :

$$f(x) = \frac{2}{1+e^{-2x}} - 1 \tag{III.2}$$

Cette fonction transforme toute entrée réelle en une valeur entre [-1,1]. Tanh est une variante de la fonction sigmoïde, la relation entre la fonction Tanh et la fonction sigmoïde : tanh(x) = 2sigmoïde(2x) -1.

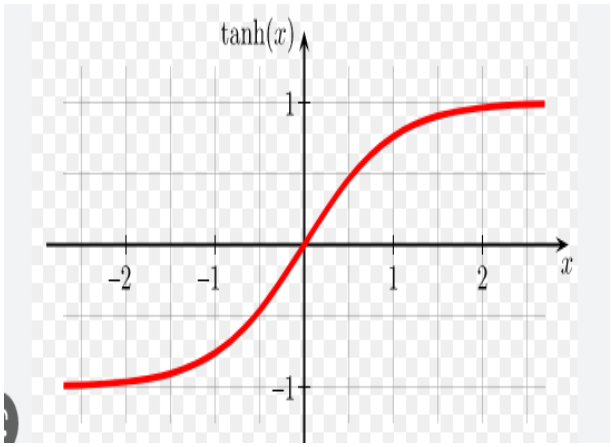


FIGURE III.5 – Fonction Tanh.

- Fonction Softmax : Elle représente une loi catégorique sur un vecteur z = $(z_1, z_2, ..., z_k)$ de K nombres réels en les convertissant en un vecteur (z) avec des probabilités K probables résultats ou la somme des probabilités K égale à 1. Dans le cas de la classification des classes K avec k >= 2, cette fonction est utilisée dans la couche de sortie pour déterminer la probabilité de à quelle classe un zi d'entrée appartient (la classe avec la probabilité la plus élevée). [32]

III.5 Les algorithmes des réseaux de neurones

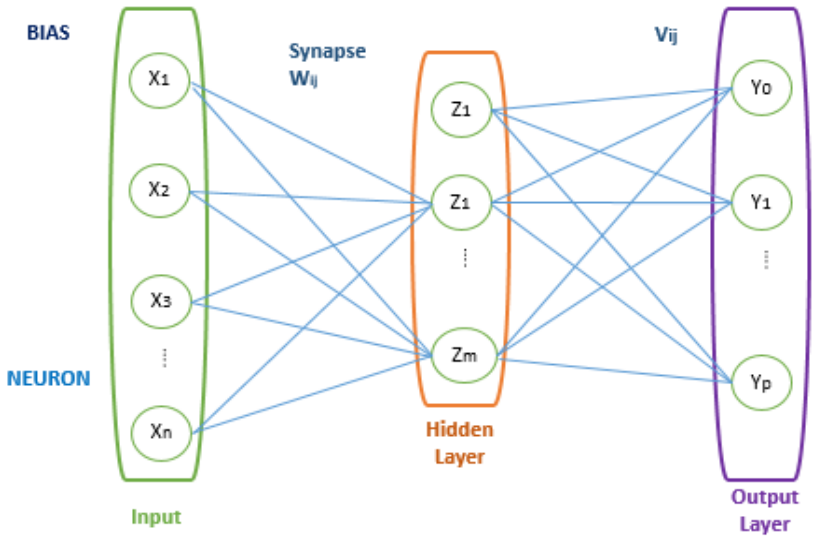


FIGURE III.6 – Schéma fonctionnement.

Soit $X_1, X_2 \ldots, X_n$ les n neurones de la couche d'entrée, $Z_1, , Z_m$ les m neurones de la couche cachée et $Y_1, , Y_p$ les p neurones de la couche de sortie. Soit 1 la valeur de biais des couches d'entrée et caché. Et soit W_{ij} et V_{ij} les poids synaptiques, respectivement entre la couche d'entrée et la couche cachée et entre la couche cachée et celle de sortie.

Chacun des neurones X_n de la couche d'entrée (ainsi que le biais) entre dans chacun des m neurones de la couche cachée et pareil pour la couche suivante . Pour obtenir une valeur de sortie d'un neurone, le processus est le suivant :

- Chaque neurone est associé à un poids synaptique. Lorsqu'une donnée d'entrée entre dans un neurone, le poids sur le neurone est multiplié par sa valeur d'entrée.
- Ainsi on calcule la somme des poids multipliés par les valeurs d'entrée à laquelle on ajoute le biais.
- Enfin, une fonction d'activation est appliquée à cette somme pondérée. Cette valeur de sortie d'un neurone peut ensuite être renvoyée aux neurones de la couche suivante. [33]

III.6 Type de réseaux neuronaux

Il existe de nombreux types de réseaux neuronaux artificiels, dont la complexité varie. Ils partagent l'objectif de refléter la fonction du cerveau humain pour résoudre des problèmes ou des tâches complexes. La structure de chaque type de réseau neuronal artificiel reflète en quelque sorte les neurones et les synapses. Cependant, ils diffèrent en termes de complexité, de cas d'utilisation et de structure. Les différences incluent également la façon dont les neurones artificiels sont modélisés dans chaque type de réseau neuronal artificiel, et les connexions entre chaque nœud. D'autres différences incluent la façon dont les données peuvent circuler à travers le réseau neuronal artificiel, et la densité des nœuds.

III.6.1 Réseaux neuronaux perceptron et multicouches

Un perceptron est l'un des modèles les plus anciens et les plus simples d'un neurone. Un modèle Perceptron est un classificateur binaire, séparant les données en deux classifications différentes. En tant que modèle linéaire, c'est l'un des exemples les plus simples d'un type de réseau neuronal artificiel.

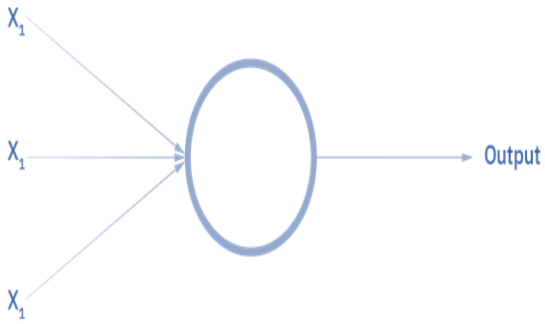


FIGURE III.7 – Réseaux neuronaux perceptron.

Les réseaux neuronaux artificiels Perceptron multicouches ajoutent complexité et densité, avec la capacité de nombreuses couches cachées entre la couche d'entrée et de sortie. Chaque nœud individuel sur une couche spécifique est connecté à chaque nœud sur la couche suivante. Cela signifie que les modèles Perceptron multicouches sont des réseaux entièrement connectés et peuvent être utilisés pour l'apprentissage profond.

Ils sont utilisés pour des problèmes et des tâches plus complexes comme la classification

complexe ou la reconnaissance vocale. En raison de la profondeur et de la complexité du modèle, le traitement et la maintenance du modèle peuvent prendre beaucoup de temps et de ressources.

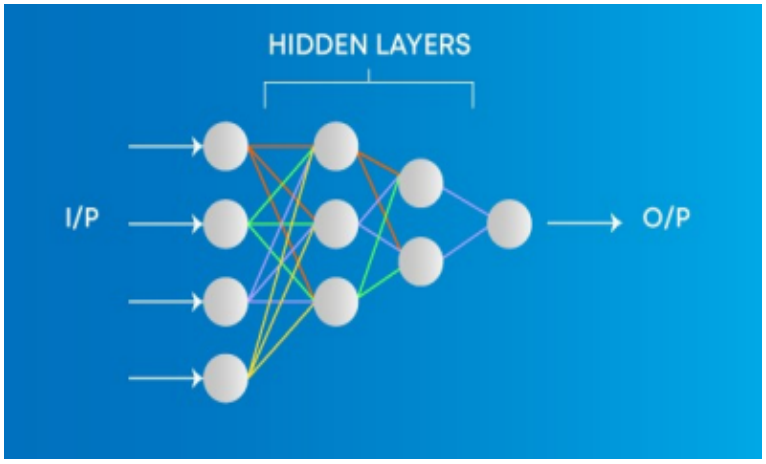


FIGURE III.8 – Réseaux neuronaux multicouches.

III.6.2 Réseaux neuronaux artificiels Feedforward

Comme son nom l'indique, un réseau neuronal artificiel Feedforward est lorsque les données se déplacent dans une direction entre les nœuds d'entrée et de sortie. Les données avancent à travers les couches de nœuds, et ne vont pas revenir en arrière à travers les mêmes couches. Bien qu'il puisse y avoir de nombreuses couches différentes avec de nombreux nœuds différents, le mouvement unidirectionnel des données rend les réseaux neuronaux Feedforward relativement simples. Les modèles de réseaux neuronaux artificiels de Feedforward sont principalement utilisés pour des problèmes de classification simplistes.

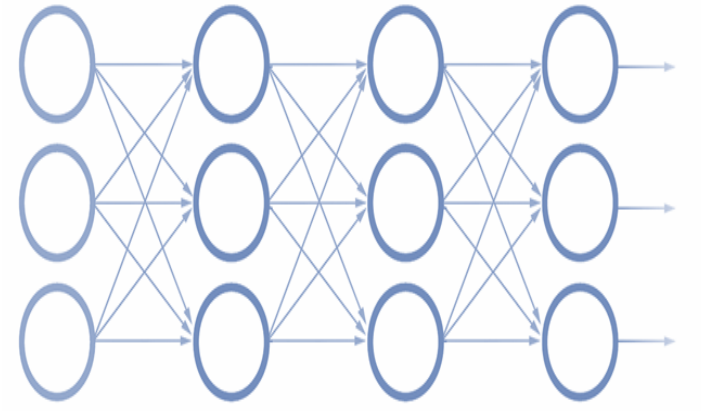


FIGURE III.9 – Feedforward.

III.6.3 Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs sont des réseaux profonds particulièrement bien adaptés au traitement d'images applications de traitement des signaux. Ils ont toutes les caractéristiques des réseaux neuronaux. Ils sont construits en empilant les couches de traitement jusqu'aux niveaux finaux qui effectuent la régression l'apprentissage et le fonctionnement du réseau sont facilités par le partage de paramètres et connexion, ainsi que moins de couches de convolution.

III.6.4 Réseaux neuronaux récurrents

Les réseaux neuronaux récurrents sont des outils puissants lorsqu'un modèle est conçu pour traiter des données séquentielles. Le modèle fera avancer les données et les fera remonter aux étapes précédentes du réseau de neurones artificiels afin de réaliser au mieux une tâche et d'améliorer les prévisions. Les couches entre les couches d'entrée et de sortie sont récurrentes, en ce sens que l'information pertinente est mise en boucle et conservée. La mémoire des sorties d'une couche est renvoyée en boucle à l'entrée où elle est maintenue pour améliorer le processus pour la prochaine entrée. [34]

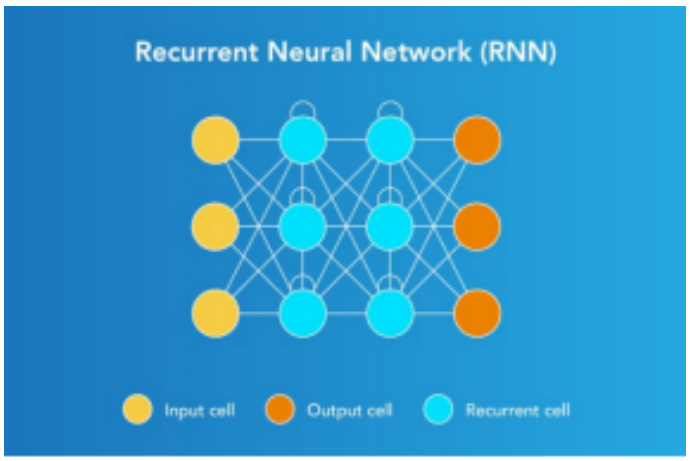


FIGURE III.10 – Réseaux neuronaux récurrent

III.7 Avantages et inconvénients

III.7.1 Avantages

- **Auto-apprentissage :** C'est la principale caractéristique et l'avantage des réseaux de neurones artificiels, qui sont si populaires auprès des programmeurs. Vous créez simplement un algorithme de base, puis lui donnez des exemples à former (par exemple, des photos de personnes, si vous voulez que votre réseau de neurones recherche des personnes sur une photo) et regardez les résultats. Dans le même temps, l'algorithme lui-même décide comment atteindre l'objectif souhaité, trouvant souvent des solutions qui ne sont pas évidentes (pour les gens). De plus, le réseau de neurones n'est pas seulement un auto-apprentissage, il est conçu pour s'auto-apprendre en permanence et améliorer ses résultats. Une fois le système formé, le programme ou l'application devient plus convivial au fur et à mesure de son utilisation. C'est pourquoi le système de recommandation de Google Translate, Netflix ou TikTok s'améliore chaque année.
- **Filtrage efficace du bruit dans les données :** Imaginez n'importe quel endroit raisonnablement bruyant, comme un marché ou un stade. Les gens parlent, la musique joue fort, les voitures passent quelque part et les oiseaux crient - il y a du bruit partout, mais malgré cela, vous pouvez communiquer calmement avec les personnes à proximité. Vos oreilles captent des tonnes de sons inutiles, mais votre cerveau les filtre et ne perçoit que ce que dit votre interlocuteur. Cette propriété se retrouve également dans les réseaux de neurones artificiels. Après la formation, ils sont capables d'isoler uniquement les informations dont ils ont besoin à partir d'un énorme flux continu de données, en ignorant tout bruit parasite. Il s'agit d'une fonctionnalité très utile si vous avez besoin de rechercher des modèles dans d'énormes quantités de données hétérogènes, telles que la recherche médicale non clinique, les prévisions météorologiques, l'analyse du marché économique ou la traduction de texte.
- **S'adapter au changement :** Un autre avantage des réseaux de neurones artificiels est la capacité de s'adapter aux changements dans les données d'entrée. Par analogie, nous pouvons donner un exemple avec la mise à jour des applications. Disons que vous êtes hors ligne depuis longtemps et pendant ce temps, Instagram et TikTok ont été mis à jour avec quelques nouvelles fonctionnalités. Après avoir pris quelques minutes pour lire les instructions, vous serez familiarisé avec toutes les nouvelles fonctionnalités

et continuerez à utiliser Instagram et TikTok. La même chose se produira avec les réseaux de neurones. Après une courte période d'adaptation aux changements, il continuera à fonctionner avec la même efficacité.

- **Tolérance aux pannes :** Les solutions basées sur les réseaux de neurones restent opérationnelles même après la défaillance d'une partie des neurones. Oui, cela peut affecter la précision et/ou la vitesse de l'algorithme, mais ses réponses seront toujours logiques, rationnelles et correctes. C'est une propriété très utile si un appareil avec un réseau de neurones embarqué doit fonctionner dans un environnement agressif (zones radioactives, en temps de guerre, dans des bâtiments ou de l'espace détruits).

III.7.2 Inconvénients

- **Le problème de la boîte noire :** Le défaut le plus notoire de tous les NN est peut-être leur nature de "boîte noire". En termes simples, vous ne savez pas comment et pourquoi votre réseau neuronal arrive à un certain résultat. Par exemple, lorsque vous mettez une photo d'un chat dans un réseau de neurones et qu'il vous indique qu'il s'agit d'un avion, il est très difficile de comprendre ce qui l'a amené à cette conclusion. Vous ne savez tout simplement pas ce qui se passe dans le "cerveau" du réseau de neurones.

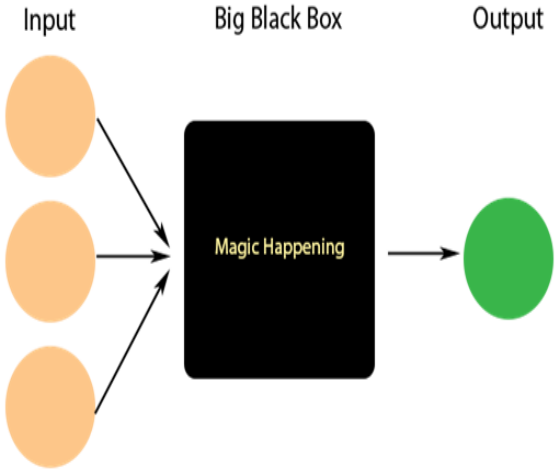


FIGURE III.11 – Problème de la boîte noire.

- **Coûteux en calcul :** Les algorithmes modernes d'apprentissage en profondeur basés sur des réseaux de neurones artificiels nécessitent plusieurs semaines, voire des années, pour apprendre à partir de zéro. Alors que la plupart des algorithmes d'apprentissage

automatique traditionnels nécessitent beaucoup moins de temps pour s'entraîner : de quelques minutes à plusieurs heures. Par exemple, un réseau de neurones avec 50 couches sera beaucoup plus lent qu'un algorithme de forêt aléatoire (une méthode d'apprentissage d'ensemble pour la classification, la régression et d'autres problèmes) avec seulement 10 arbres.

- **Durée de développement :** Bien qu'il existe de nombreuses bibliothèques telles que NeuroLab, ffnet, SciPy, TensorFlow, Scikit-Neural Network, Lasagne, pyrenn, NumPy, Spark MLlib, Scikit-Learn, Theano, PyTorch, Keras qui permettent d' économiser du temps et des efforts lors du développement de réseaux de neurones artificiels , elles ne sont pas toujours applicables. Par exemple, si vous avez besoin de créer une solution nouvelle ou plutôt complexe qui nécessite plus de contrôle sur les détails de l'algorithme.
- **La quantité de données :** Le prochain inconvénient des réseaux de neurones est que leur formation nécessite généralement beaucoup plus de données que les algorithmes d'apprentissage automatique traditionnels. Et comme nous l'avons déjà dit, s'il s'agit de données uniques ou difficiles à collecter, cela peut être un sérieux défi pour les développeurs. Et souvent bien plus que d'écrire le code d'un réseau de neurones artificiels. [33]

III.8 Conclusion

Les réseaux de neurones artificiels sont parfaits pour certaines tâches et moins bons pour d'autres. Cependant, peu de gens comprennent quand ils peuvent réellement apporter une réelle valeur à votre entreprise et quand il est préférable de se tourner vers d'autres options pour mettre en œuvre l'intelligence.

Dans le chapitre suivant, nous allons introduire les concepts et l'implémentation.

Chapitre IV

Conception

IV.1 Introduction

La partie conception dans un projet informatique a une très haute importance, elle permet d'avoir une idée de ce qu'on doit programmer, et déterminer les différentes fonctionnalités de l'application.

Dans ce chapitre nous présentons la conception de notre système en commençant par sa conception générale puis sa conception détaillée, qui se base sur l'analyse des sentiments.

IV.2 Conception globale du système

L'architecture de notre système de classification des sentiments peut être présentée par la figure (IV.1) :

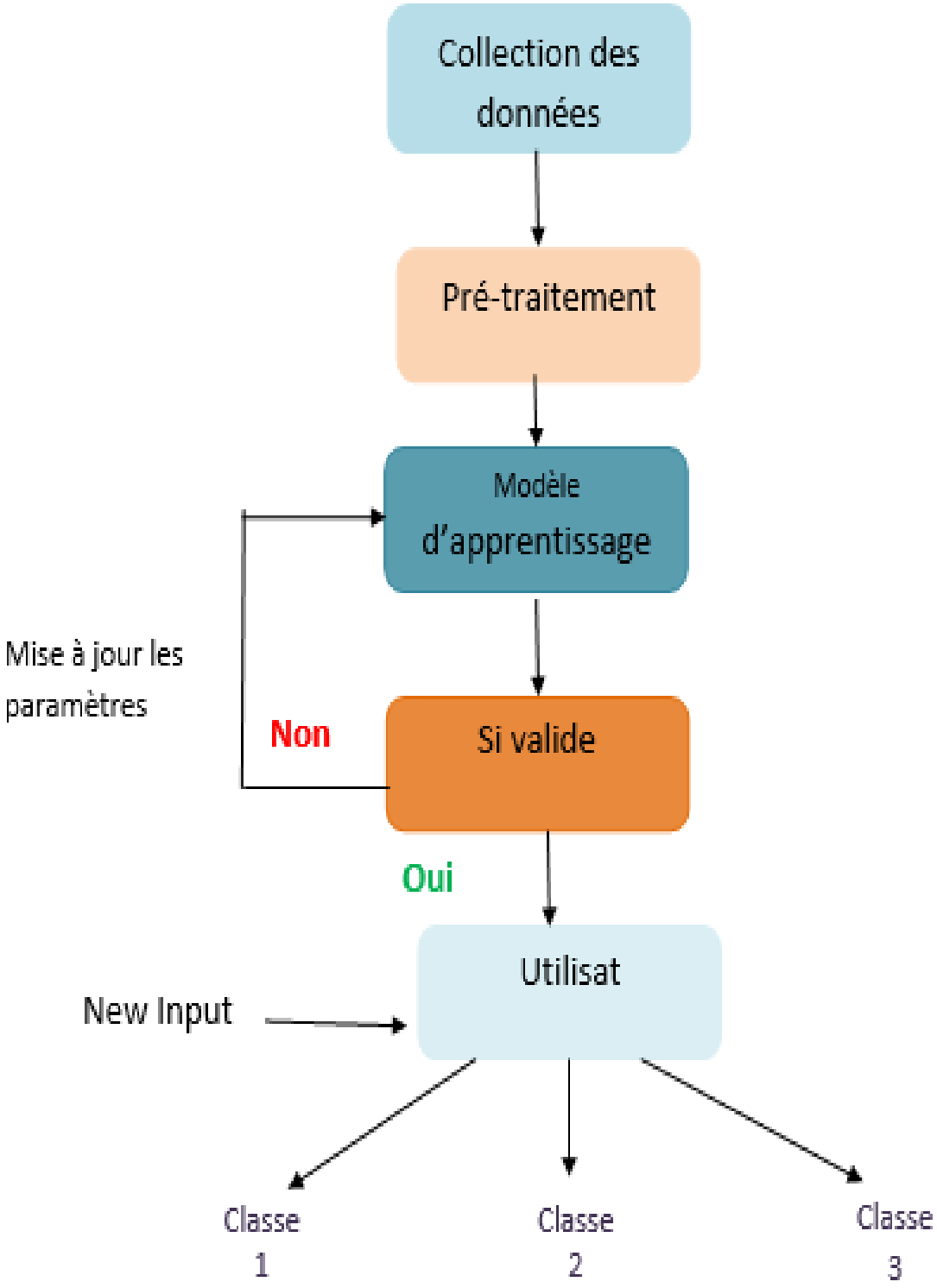


FIGURE IV.1 – Architecture globale du système.

IV.3 La conception détaillée du système

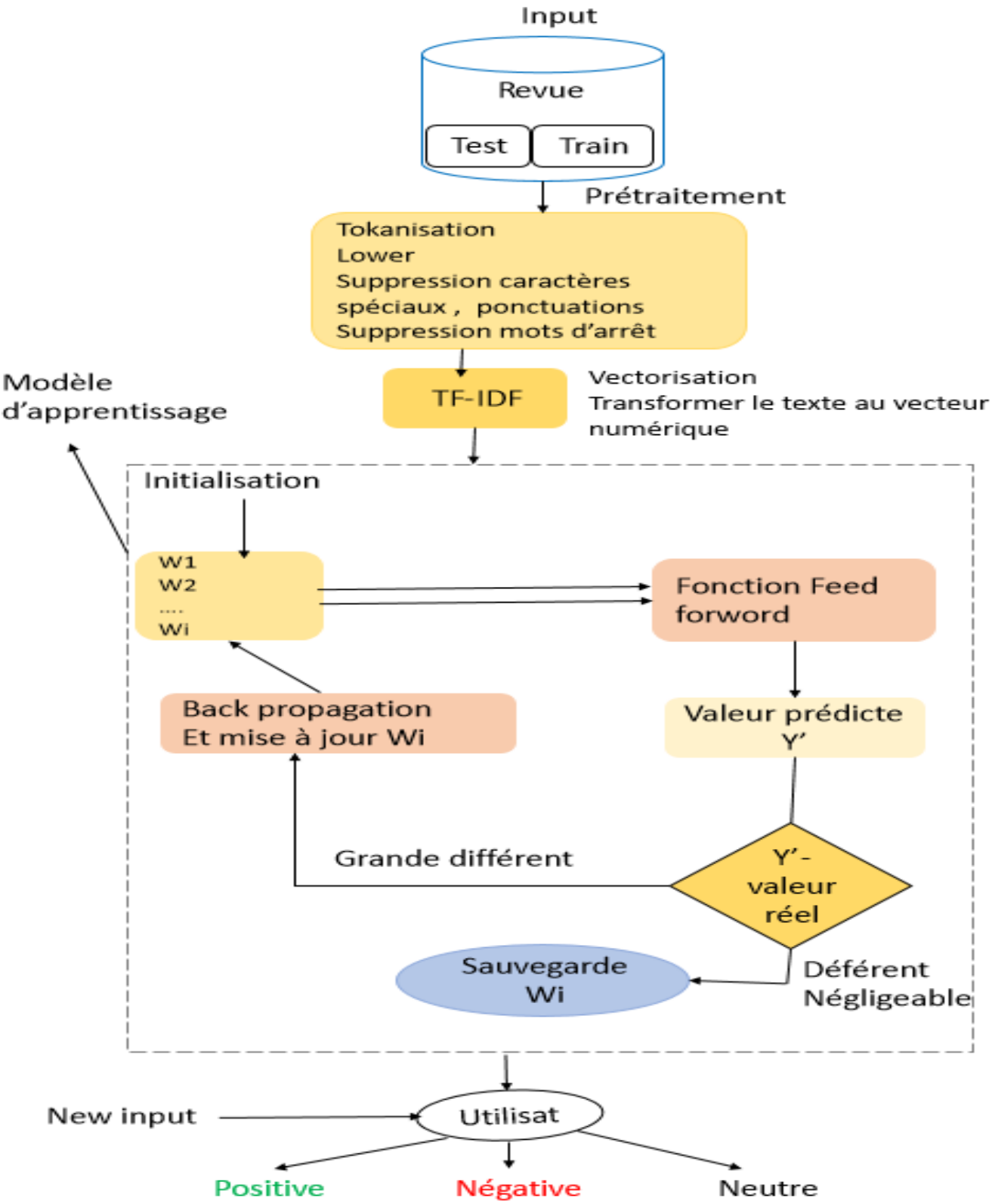


FIGURE IV.2 – Architecture détaillée du système.

IV.3.1 Base d'Apprentissage

Est un ensemble de données d'exemples utilisé pendant le processus d'apprentissage et est utilisé pour ajuster les paramètres.

IV.3.2 Base de Test

Une Base de Test est donc des exemples utilisés uniquement pour évaluer les performances (c'est-à-dire la généralisation) d'un classificateur entièrement spécifié.

IV.3.3 Collection des données

La première étape du processus d'analyse des sentiments consiste à collecter des commentaires. Dans notre cas les données sont collectées et sauvegardées dans un fichier texte qui s'appelle " reviews.txt".

IV.3.4 Le pré-traitement

Cette étape est au cœur de notre approche d'analyse des sentiments. Avant de commencer la classification des messages en positifs, négatifs ou neutres, un ensemble de données propre doit être fourni au modèle d'apprentissage automatique afin d'obtenir un modèle de classification puissant avec un score plus élevé. Dans ce qui suit vous détaillez les étapes de pré-traitement.

IV.3.4.1 La conversation des données textuelles en minuscules

Cette étape consiste de convertir tous les commentaires en minuscule. Pour faire cette étape nous utilisons la fonction "Lower()" en Python . Cette dernière convertie tous les caractères majuscules.

IV.3.4.2 Le nettoyage des données

consiste de supprimer et éliminer toutes les données non nécessaires dans le fichier Ces données comme les hashtags, les émojis...

1. **Suppressions des mots d'arrêt :** Mot qui n'a pas de valeur dans la compréhension d'un texte. Dans l'indexation des informations, les mots vides ne sont pas référencés. [35]

2. **La tokenisation :** La tokénisation est une des premières étapes de tout système de traitement automatique des langues Cette tâche de découpage d'un texte en mots et phrases est donc d'une importance primordiale. [36]
3. **La suppression des caractères spéciaux et des ponctuations :** Les commentaires contiennent généralement les ponctuations et caractères spéciaux. Toutes seront supprimées à l'aide de la syntaxe d'expression régulière.

IV.3.5 Modèle d'apprentissage

IV.3.5.1 Extraction de fonctionnalités

L'extraction de fonctionnalités est une tâche concernant la transformation de données brutes en entrées appropriées qui peuvent être consommées par un algorithme d'apprentissage automatique particulier. Pour faire l'extraction des caractéristiques. Nous avons adopté la méthode TF-IDF.

1. **La méthode TF-IDF**

   Dans la technique TF-IDF (Term Frequency-Inverse Document Frequency), les textes volumineux sont convertis en phrases, puis la fréquence des termes pondérés, et la fréquence des phrases inverses est calculée où la fréquence des phrases est définie comme le nombre de phrases du document, qui impliquent ces termes Les vecteurs des phrases sont calculés et comparés aux autres phrases et sont ensuite notés. Le produit de TF et IDF calcule la valeur TF-IDF d'un mot/terme, où TF (fréquence du terme) est défini comme le nombre de fois qu'un mot apparaît dans un document et IDF est la fréquence inverse du document Les phrases avec le score le plus élevé sont considérées comme les phrases concluantes pour le résumé.

   L'estimation TF-IDF de tout et le mot d'action seraient alors déterminés à partir du récapitulatif prétraité des mots. Les calculs de TF-IDF peuvent être effectués à l'aide de l'équation (4.3). [37]

   Calcule de TF (fréquence de terme) :

$$tf_w = \frac{Nomber\ de\ terme\ dans\ le\ document}{Nomber\ totale\ des\ termes\ dans\ le\ document} \tag{IV.1}$$

Calcule d' idf (fréquence inverse de document) :

$$Idf_i = \log \frac{|D|}{|(d_j : t_i \in d_j)|} \tag{IV.2}$$

Où :

$|D|$: Nombre total de document dans le corpus.

$|(d_j : t_i \in d_j)|$: Nombre de document ou le terme ti apparait.

Calcule de tf-idf :

$$Tf\_idf = tf * idf \tag{IV.3}$$

IV.3.5.2 feed forward

Les Réseau de neurones de types « feed forward » ou a propagation directe sont les réseaux de neurones ou l'information passe de la couche i à la couche j avec i < j. [38]

IV.3.5.3 Back propagation

C'est un algorithme de reviens vers l'arrière (c.-à-d. algorithme de feed-forward inversé avec quelque calculs effectués) qui permet de calculer l'erreur entre le résultat réel et le résultat obtenu pour faire les mises à jours des poids (Wi) pour chaque itération. [38]

IV.3.5.4 Le poids « W » (coefficient synaptique)

Est une valeur numérique associée à une connexion entre deux unités (neurones) qui reflète la force de relation (connexion) entre ces deux unités i et j, et il est noté par Wi.

IV.4 Conclusion

Dans ce chapitre, Nous avons définir l'architecture globale et détaillé de notre système et les différentes étapes pour classification et détection si l'évènement est un positive ou négative ou neutre.

Dans le chapitre suivant on va entamer l'implémentation de notre système.

Chapitre V

Implémentation

V.1 Introduction

Dans ce chapitre, nous allons présenter l'environnement de travail, le langage de programmation, les outils matériels et logiciels que nous avons utilisé pour réaliser ce projet. De plus, nous allons introduit en détaille une analyse exploratoire de données sur les résultats que nous avons obtenue.

V.2 L'environnement de travail et les outils utilisés

V.2.1 L'environnement Matériel

Le matériel utilisé est représenté dans le tableau suivant :

	POSTE DE TRAVAIL
Pc	Toshiba
Système d'exploitation	Windows 10 Professionnel
Processeur	Processeur Intel(R) Core (TM) 2 Duo CPU T6400 @ 2.00GHz
RAM	4,00 Go
Type de système	SE64 bits

TABLE V.1 – Les caractéristiques de Matériel

V.2.2 L'environnement Logiciel

V.2.2.1 Python

Pour atteindre notre but, nous avons utilisé le langage de programmation Python,version 3.6. Ce dernier est développé depuis 1989 par Guido Var Rossum. 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. [39]

FIGURE V.1 – Logo python

V.2.2.2 Anvantage et inconvénients

Anvantage

- Il est multiplateforme. C'est-à-dire qu'il fonctionne sur de nombreux systèmes d'exploitation : Windows, Mac OS X, Linux, Android, iOS, depuis les mini-ordinateurs Raspberry Pi jusqu'aux supercalculateurs.
- Il est gratuit. Vous pouvez l'installer sur autant d'ordinateurs que vous voulez (même sur votre téléphone !).
- C'est un langage de haut niveau. Il demande relativement peu de connaissance sur le fonctionnement d'un ordinateur pour être utilisé.
- C'est un langage interprété. Un script Python n'a pas besoin d'être compilé pour être exécuté, contrairement à des langages comme le C ou le C++.

- Enfin, il est très utilisé en bioinformatique et plus généralement en analyse de données. Toutes ces caractéristiques font que Python est désormais enseigné dans de nombreuses formations, depuis l'enseignement secondaire jusqu'à l'enseignement supérieur. [40]

Inconvénients

- Malgré ses nombreux points forts, Python n'est pas adapté à toutes les tâches. Il s'agit d'un langage « de haut niveau ». Il n'est donc pas adéquat pour la programmation au niveau du système.
- Il n'est pas non plus idéal pour les situations nécessitant des binaires indépendantes cross-plateforms. Une application indépendante pour Windows, macOS et Linux ne sera pas facile à coder en Python.
- Enfin, il vaut mieux éviter Python pour les situations où la vitesse est une priorité absolue pour l'application. Mieux vaut se tourner vers C et C++ ou autre langage du même acabit.
- Chaque fonction et module sont considérés comme des objets par Python. Ceci simplifie l'écriture de code de haut niveau, mais atténue la vitesse. [41]

V.3 Editeur de code

Nous avons utilisé Visual Studio Code pour éditer le code de notre systéme, Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS3.

Est un éditeur de code source qui peut être utilisé avec une variété de langages de programmation, notamment Python, Java, JavaScript, et C++. Elle est multi-plateforme, open source et gratuit. [42]

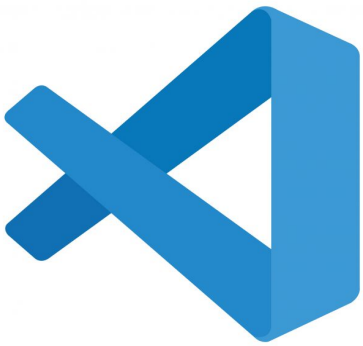


FIGURE V.2 – Logo Visual Studio Code.

FIGURE V.3 – Interface Visual.

V.4 Librairies et bibliothèques Python

FIGURE V.4 – Bibliothèques Python.

Nous avons utilisé des différentes packages et bibliothèques comme :

NLTK :

Le NLTK, ou Natural Language Toolkit, est une suite de bibliothèques logicielles et de programmes. Elle est conçue pour le traitement naturel symbolique et statistique du langage anglais en langage Python. C'est l'une des bibliothèques de traitement naturel du langage les plus puissantes. Cette suite d'outils rassemble les algorithmes les plus communs du traitement naturel du langage comme le tokenizing, le part-of-speech tagging, le stemming, l'analyse de sentiment, la segmentation de topic ou la reconnaissance d'entité nommée. [43]

NumPy :

Numerical Python) est une bibliothèque de python qui comporte des fonctions permettant de manipuler des matrices ou tableaux multidimensionnels. [44]

FIGURE V.5 – Logo Numpy.

Matplotlib : Matplotlib est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques. Elle peut être combinée avec les bibliothèques python de calcul scientifique NumPy et SciPy. Matplotlib est distribuée librement et gratuitement sous une licence de style BSD. Sa version stable actuelle (la 2.0.1 en 2017) est compatible avec la version 3 de Python. [45]

Sklearn :

Scikit-learn est une bibliothèque d'apprentissage automatique gratuite pour Python. Il comporte divers algorithmes tels que la machine vectorielle de support, les forêts aléatoires et les k-voisins, et il prend également en charge les bibliothèques numériques et scientifiques Python telles que NumPy et SciPy. [37]

Keras :

Keras est une bibliothèque de haut niveau qui fournit une API Machine Learning pratique pardessus d'autres bibliothèques de bas niveau pour le traitement et la manipulation des tenseurs, appelées Backends . A cette époque, Keras peut être utilisé sur l' un des trois backends disponibles : tensorflow, Théano et CNTK. [46]

FIGURE V.6 – Logo Keras.

Panda :

- C'est une bibliothèque Python pour la manipulation et l'analyse de données.
- Il est open source, disponible gratuitement et multiplateforme.
- fournit des structures de données et des opérations pour manipuler des tables numériques et des séries chronologiques à analyser.
- Il permet à l'utilisateur d'effectuer des opérations sur les données stockées dans les tables comme le filtrage, le tri, le regroupement, la fusion. [47]

V.5 Implémentation

Algorithme de prédiction de la polarité des commentaires utilisant le Réseau de neurones : -Au début, j'utilisais sigmoïde et son dérivé dans mon réseau de neurones :

```
#sigmoid activation function
    def sigmoid(self, x):
        return 1 / (1 + np.exp(-x))
 #derivative of sigmoid function
    def sigmoid_derivative(self, x):
        return x * (1 - x)
```

Mais j'ai continué à obtenir une faible précision pour mon modèle. Après des recherches, j'ai trouvé que le sigmoïde ne convient qu'à la classification de 2 sorties : "positive" ou "négative" mais dans mon cas je l'utilisais pour classer 3 sorties : "positive" ou "négative" ou "neutre" c'est pourquoi J'ai dû passer à d'autres fonctions.

J'ai expérimenté différentes fonctions telles que relu et softmax. Cependant, j'ai fini par les utiliser tous les deux. Mélange de relu et softmax pour obtenir le meilleur résultat.

```
#ReLU activation function
    def ReLU(self, x):
        return np.maximum(0,x)
    #derivative of ReLU function
    def ReLU_derivative(self, x):
        x[x<=0] = 0
        x[x>0] = 1
        return x #Softmax activation function
    def softmax(self, x):
        exp_scores = np.exp(x - np.max(x, axis=1, keepdims=True))
        return exp_scores / np.sum(exp_scores, axis=1, keepdims=True)
```

```
    # derivative of Softmax activation function
    def softmax_derivative(x):
        s = x.reshape(-1,1)
        return np.diagflat(s) - np.dot(s, s.T)
```

-class NauralNetwork :

Ou debut il ya les constructeurs de la classe NauralNetwork qui sont initialisée par des valeurs random.

```
self.weights1 = np.random.randn(input_size, hidden_size) / np.sqrt
                  (input_size)
self.weights2 = np.random.randn(hidden_size, output_size) / np.sqrt
                 (hidden_size)
```

feedforward function :

Dans la premeire utilisation, j'ai utilisé « sigmoid »

```
# feedforward function
    def feed_forward(self, X):
        self.hidden_layer = self.sigmoid(np.dot(X, self.weights1))
        self.output_layer = self.sigmoid(np.dot(self.hidden_layer,
                            self.weights2))
        return self.output_layer
```

Dans la deuxiem utilisation j'ai combiné ReLu et softmax, dans le hidden_layer j'ai utilisé ReLu et softmax dans output_layer :

```
    def feed_forward(self, X):
        self.hidden_layer = self.ReLU(np.dot(X, self.weights1))
        self.output_layer = self.softmax(np.dot(self.hidden_layer,
                             self.weights2))
        return self.output_layer
```

backpropagation function :

Sigmoid derivative que j'ai utilisé au début de mon travail :

```
def backpropagation(self, X, y, learning_rate, lambd):
        output_error = y - self.output_layer
        output_delta = output_error * self.sigmoid_derivative
                       (self.output_layer)
        hidden_error = np.dot(output_delta, self.weights2.T)
        hidden_delta = hidden_error * self.sigmoid_derivative
                       (self.hidden_layer)
```

Après ça j'ai suivi ReLu derivative :

```
def backpropagation(self, X, y, learning_rate, lambd):
        output_error = y - self.output_layer
        output_delta = output_error * self.ReLU_derivative
                       (self.output_layer)
        hidden_error = np.dot(output_delta, self.weights2.T)
        hidden_delta = hidden_error * self.ReLU_derivative
                       (self.hidden_layer)
```

Pour ajuster les poids weight1 et weight2 :

```
train(self, X_train, y_train, X_val, y_val, epochs, learning_rate,
            early_stopping_patience, lambd):
 best_weights1 = self.weights1.copy()
 best_weights2 = self.weights2.copy()
```

Je faisais face à un problème appelé Over-fitting, J'obtenais une faible précision sur mon modèle à cause de ce problème.

Qu'est-ce que le OVER FITTING ?:

Overfitting ou (Le surajustement) est un comportement d'apprentissage automatique indésirable qui se produit lorsque le modèle d'apprentissage automatique donne des prédictions précises pour les données d'apprentissage, mais pas pour les nouvelles données. Lorsque les scientifiques des données utilisent des modèles d'apprentissage automatique pour faire des prédictions, ils entraînent d'abord le modèle sur un ensemble de données connu.

Toutes les techniques ci-dessous ont été utilisées pour améliorer la précision du modèle :

- La régularisation L2 :

  Également connue sous le nom de régularisation Ridge, est une technique d'apprentissage automatique qui évite le surajustement en introduisant un terme de pénalité dans la fonction de perte du modèle basé sur les carrés des paramètres du modèle. L'objectif de la régularisation L2 est de garder les tailles des paramètres du modèle courtes et d'éviter le surdimensionnement.

  On a **Utilisé dans backpropaggation function :**

  ```
     L2 regularization
   reg_term2 = lambd * self.weights2
   reg_term1 = lambd * self.weights1
  ```

  Learing rate :

Contrôle la rapidité avec laquelle le modèle est adapté au problème (généralement très petit).

Nous mettons le mise à jour de poids W1 et W2 par apport la régularisation L2 :

```
self.weights2 += learning_rate * (np.dot(self.hidden_layer.T,
                 output_delta) - reg_term2)
self.weights1 += learning_rate * (np.dot(X.T, hidden_delta) -
                 reg_term1)
```

Dans la fonction trainfunction j'ai utilisé :

- Cross-entropyloss :

  Ou (La perte d'entropie croisée) est utilisée lors de l'ajustement des poids du modèle pendant l'entraînement. L'objectif est de minimiser la perte, c'est-à-dire que plus la perte est faible, meilleur est le modèle. Un modèle parfait a une perte d'entropie croisée de 0.

```
      # calculate validation loss using cross-entropy loss
           log_probs = -np.log(output)
           cross_entropy_loss = np.sum(log_probs * y_train)
                                / X_train.shape[0]

```

- EarlyStopping :

  Ou (L'arrêt précoce) est une technique de régularisation pour les réseaux de neurones profonds qui arrête la formation lorsque les mises à jour des paramètres ne commencent plus à produire des améliorations sur un ensemble de validation.

```
# early stopping check
    if no_improvement_count >= early_stopping_patience:
    #print("Early stopping, no improvement for",
                             early_stopping_patience, "epochs")
                # mise a jour les poids
                self.weights1 = best_weights1
                self.weights2 = best_weights2
                break
```

- Apprentissage renforcé :

  Il est utilisé en ré-alimentant notre réseau de neurones avec l'ensemble de données alors qu'il est déjà en train d'apprendre.

```
#load learning data for the second time to improve accuracy with
```

```
open('C:/Users/poste/Desktop/pgrm/reviews.txt', 'r') as f:
            lines = f.readlines()
texts = []
        sentiments = []
        for line in lines:
match = re.match(r'^(.*?)\s (Positive|Negative|Neutral)$', line)
            if match:
                text = match.group(1)
                sentiment = match.group(2)
                texts.append(text)
                sentiments.append(sentiment)
data = pd.DataFrame({'text': texts, 'sentiment': sentiments})
data['text'] = data['text'].apply(lambda x: clean_text(x))

        tokenizer.fit_on_texts(data['text'].values)
        X = tokenizer.texts_to_sequences(data['text'].values)
        X = pad_sequences(X)
```

Pour la partie Prétraitement :

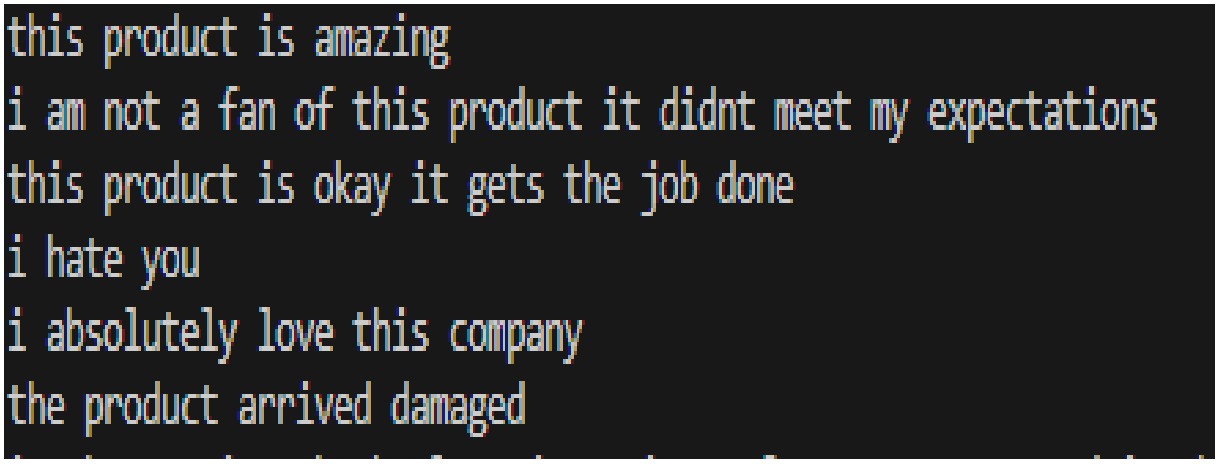
Au début,on doit vérifier qui les input est une string, et puis on les transforme en minuscule suivi par la tokenisation, après avoir effecteé ces opérations on fait la Suppressions des mots d'arrêt.

```
# fonction tokenizationdef tokenize(text):
if not isinstance(text, str):
    raise ValueError("Input must be a string")

text = re.sub(r'[^\w\s]', '', text.lower())

```

Avant :

```
This product is amazing!.

I am not a fan of this product It didn't meet my expectations.

This product is okay It gets the job done.

I hate you.

I absolutely love this company.

The product arrived damaged.
```

Après :




```
    tokens = text.split()

```

Avant :

```
This product is amazing!.

I am not a fan of this product It didn't meet my expectations.

This product is okay It gets the job done.

I hate you.

I absolutely love this company.

The product arrived damaged.
```

Après :

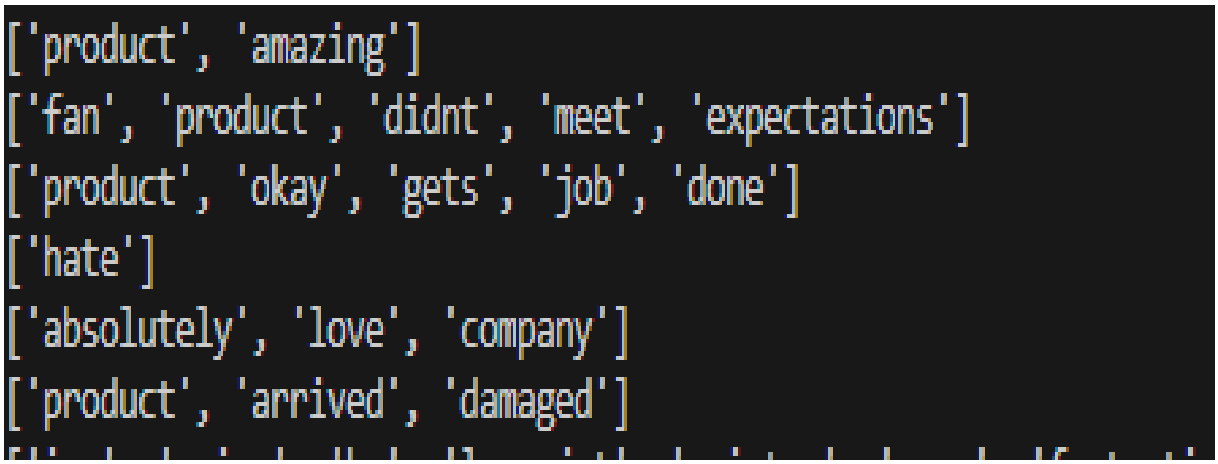
```
['this', 'product', 'is', 'amazing']
['i', 'am', 'not', 'a', 'fan', 'of', 'this', 'product', 'it', 'didnt', 'meet', 'my', 'expectations']
['this', 'product', 'is', 'okay', 'it', 'gets', 'the', 'job', 'done']
['i', 'hate', 'you']
['i', 'absolutely', 'love', 'this', 'company']
['the', 'product', 'arrived', 'damaged']
```

```
    # stopwords
    stop_words = set(stopwords.words('english'))
    tokens = [token for token in tokens if token not in stop_words]
    return tokens
```

Avant :

```
This product is amazing!.

I am not a fan of this product It didn't meet my expectations.

This product is okay It gets the job done.

I hate you.

I absolutely love this company.

The product arrived damaged.
```

Après :



Pour lire la base des données qui est présente dans un fichier texte : En écrivons la fonction load_data, Nous ouvrons le fichier, où il doit lire ligne par ligne et chaque ligne devrait se composer de deux parties partie commentaire et partie sentiment où les commentaires à gauche et polaire sont à droite.

```
def load_data(file_path):
    reviews = []
    labels = []
    with open(file_path, 'r') as f:
        for line in f:
            parts = line.strip().split()
            if len(parts) >= 2:
                review = ' '.join(parts[:-1])
                sentiment = parts[-1]
                if sentiment.lower() == 'positive':
                    labels.append([1, 0, 0])
                elif sentiment.lower() == 'negative':
                    labels.append([0, 1, 0])
                else:
                    labels.append([0, 0, 1])
                tokens = tokenize(review)
                reviews.append(tokens)
    return reviews, np.array(labels)
```

La fonction TF-IDF :

```
def compute_idf(docs):
```

```
    n_docs = len(docs)
    idf_dict = {}
    for doc in docs:
        for word in doc:
            if word not in idf_dict:
                idf_dict[word] = 1
            else:
                idf_dict[word] += 1
    for word in idf_dict:
        idf_dict[word] = np.log(n_docs / (1 + idf_dict[word]))
    return idf_dict
```

Pour convertir dataset du string en numéro on applique la fonction vectorize_data :

```
    def vectorize_data(reviews, vocab):
    X = np.zeros((len(reviews), len(vocab)))
    idf_dict = compute_idf(reviews)
    for i, review in enumerate(reviews):
        word_freq = dict(Counter(review))
        review_len = len(review)
        for word in review:
            if word in vocab:
                tf = word_freq[word] / review_len
                idf = idf_dict[word]
                X[i][vocab[word]] = tf * idf
    return X
```

J'ai calcule la précesion (accuracy) :

```
    accuracy = model.evaluate(X_test, y_test, batch_size=batch_size)
        print('Accuracy: %.2f%%' % (accuracy*100))
```

Dans la partie principale (main) : j'ai donné le chemin de dataset :

```
    file_path = 'C:/Users/poste/Desktop/pgrm/reviews.txt'
```

J'ai éxtrait la vocabulaire après des fichier lu :

```
create vocabulary
    vocab = {}
    for review in reviews:
        for word in review:
            if word not in vocab:
                vocab[word] = len(vocab)
```

L'étape suivant fait le shuffle.

Shuffle : Dans les tâches d'apprentissage automatique, il est courant de mélanger les données, est une opération pour aider les informations à circuler à travers les canaux de fonctionnalités dans les réseaux de neurones.

```
indices = np.arange(len(X))
    np.random.shuffle(indices)
    X = X[indices]
    y = labels[indices]
```

j'ai écrit un objet de la classe neuralnetwork ,qui contient la taille de coche entrée et sortie et caché.

```
    input_size = len(vocab)
    hidden_size = 100
    output_size = 3
    neural_network = NeuralNetwork(input_size, hidden_size, output_size)
```

Dans cette étape, nous allons filtrer notre commentaire on deux partie : l'entrainement =80% et partie test=20% et nous allons donnés les nombre des epochs sans oublie learning rate.

Epochs :

Une epoch comme le nombre de passages d'un dataset d'entraînement par un algorithme.

```
    X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2,
                random_state=42)
    epochs = 70
    learning_rate = 0.01
```

Les résultats :

Apres l'entrainement de modèle nous allons le tester :

```
    new_text = ['the food was terrible, the service was also very bad']
```

Quand j'ai utilisé « function sigmoid » nous avons obtenu :

Accuracy : 40%

Quand j'ai utilisé « function relu et softmax» nous avons obtenu :

Accuracy : 75.04%

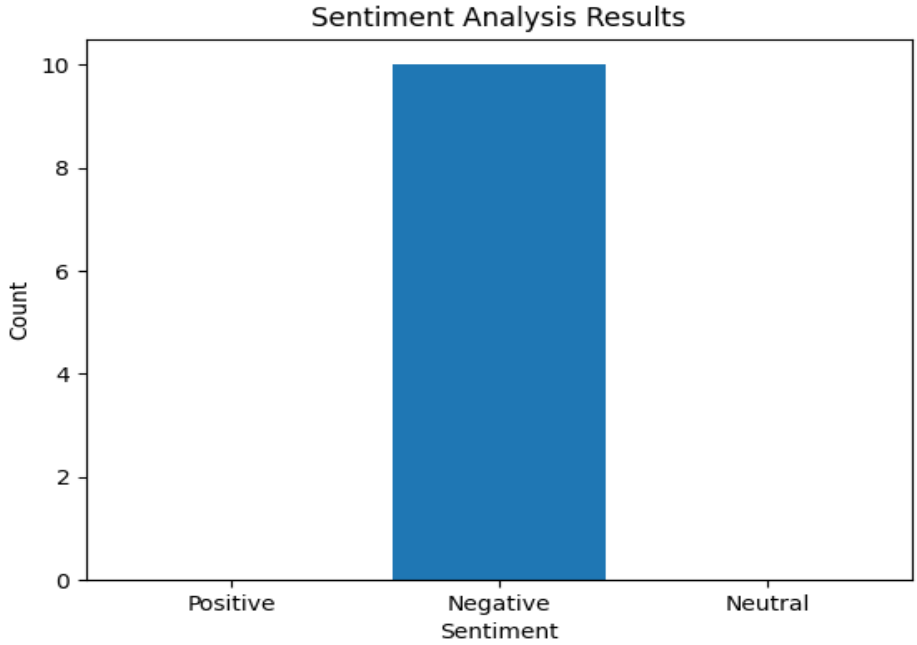


FIGURE V.7 – Le résultat d'exécution de modèle 1.

```
new_text = ['I love my car, it is fast and also very cheap awesome car']
```

Fonction sigmoid :

Accuracy : 45%

Fonction relu et softmax :

Accuracy : 75.04%

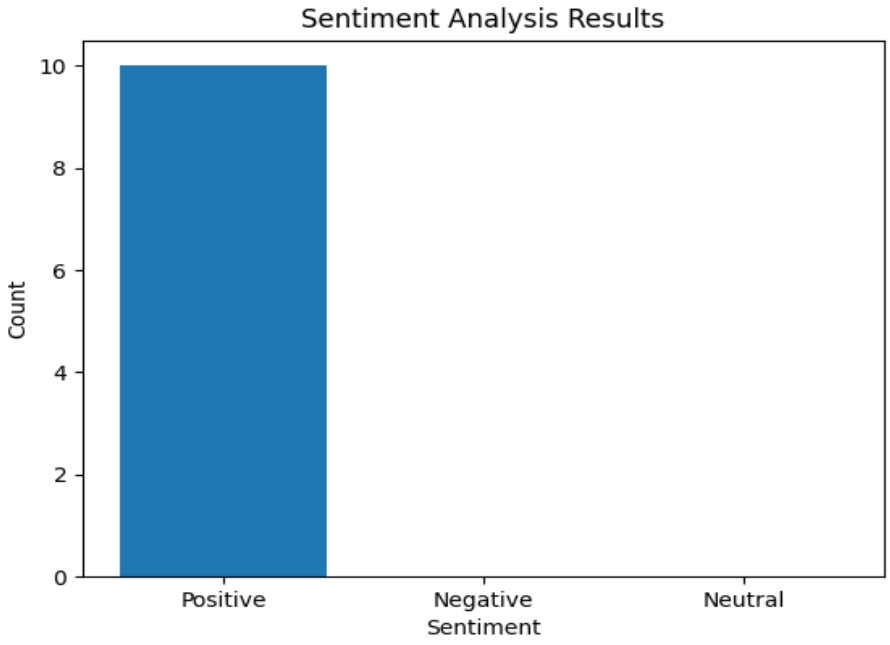


FIGURE V.8 – Le résultat d'exécution de modèle 2.

```
new_text = = ['I have no strong feelings about this topic']
```

Fonction sigmoid :

Accuracy : 22%

Fonction relu et softmax :

Accuracy : 75.04%

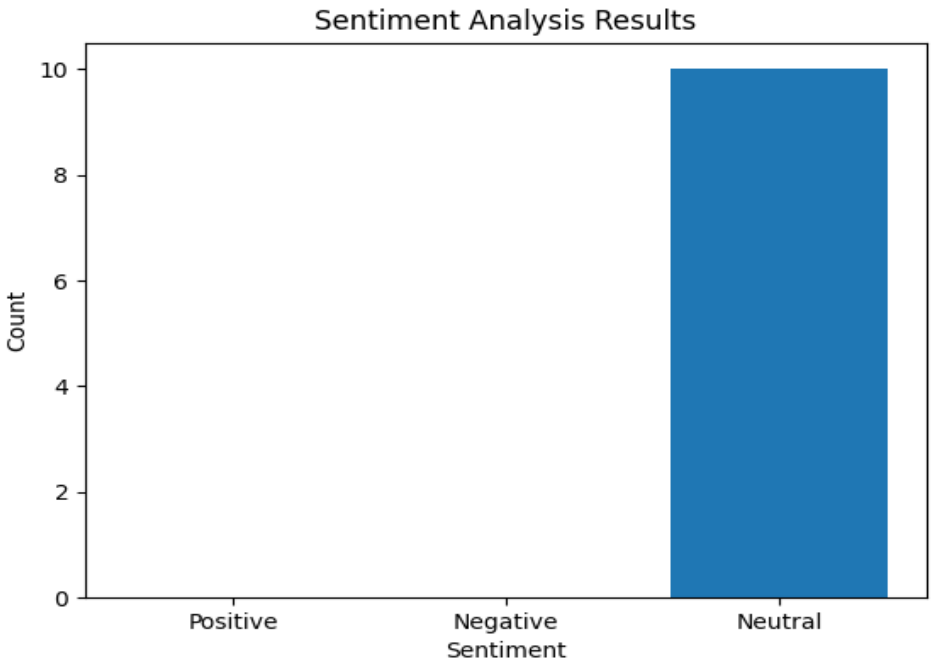


FIGURE V.9 – Le résultat d'exécution de modèle 3.

Après les résultats obtenus, nous concluons que la fonction parfaite qui donne les meilleurs résultats est une combinaison entre les deux fonctions relu et soft max.

V.6 Conclusion

Dans ce dernier chapitre, nous avons entamé le processus de l'implémentation afin de réaliser notre système, nous voyons notre contribution au problème de l'analyse des sentiments, représentant les outils et les ensembles des données que nous avons utilisés, et les étapes que nous avons suivies pour obtenir les résultats que notre modèle montre également.

Conclusion générale

L'objective de ce mémoire est la détection des polarités des publications dans les réseaux sociaux selon trois voies, une publication positive, une publication négative et une publication neutre, en se basant sur l'apprentissage automatique utilisé,nous avons choisi l'algorithme du réseau de neurones vu son efficacité à résoudre ce genre de problématique.

Le but de notre travail est la réalisation d'une application sous Python qui utilise une source de données, contient des textes annotés par des valeurs. Ce rapport a fourni des étapes détaillées du processus que nous avons suivi, y compris des projets de recherche connexes, des définitions de concepts de base liés au traitement du langage naturel et à l'analyse des sentiments.

Le domaine de l'analyse des sentiments se développe très rapidement et vise à utiliser les opinions ou les textes présents dans diverses plateformes médiatiques grâce à des techniques d'apprentissage en profondeur. Il est devenu un domaine de recherche très en vogue. De nombreuses recherches ont été effectuées dans ce domaine, mais comme l'analyse des sentiments traite de données textuelles non structurées, de nombreux problèmes subsistent. Par conséquent, dans ce mémoire, nous introduisons d'abord le processus d'analyse des sentiments, y compris ses applications, tâches et défis de l'analyse des sentiments. De même, nous introduisons des techniques d'apprentissage en profondeur.

La mise en œuvre et la conception de notre projet est de construire un modèle pour analyser les sentiments et les opinions dans un tweets rédigés en langue anglaise. afin qu'il puisse être utilisé dans la planification et la prise de décision dans tous les domaines, à travers un dictionnaire de polarisation qui se compose de "positif et négatif et neutre" au cours du processus de collecter et classer les tweets.

Nous avons identifié les termes positifs et négatifs et neutre et appliqué des modèles d'apprentissage en profondeur pour classer les textes selon un lexique précédemment rapporté.

L'outil que nous avons développé en utilisant le langage python est le fruit d'un travail

aussi bien théorique que pratique. Bien que le sujet traité soit un sujet actuellement en vogue et d'actualité en tant qu'axe de recherche, nous pouvons dire que les résultats obtenus sont satisfaisants et que l'objectif que nous nous sommes fixé est atteint à un degré acceptable.

Toutefois, et comme on dit aucun travail n'est jamais parfait, notre travail reste ouvert à d'autres contributions.

Perspectives :

Afin d'enrichir cette thèse, nous souhaitons quelques perspectives à notre travail et nous citons :

Tester notre modèle sur d'autres ensembles de données

Il serait également important, de développer un système en utilisant l'analyse d'opinions multilingues, vu que les sites web algériens contiennent multiples langues : Français, Anglais, Arabe et dialecte alg

Implémenter d'autres algorithmes de classification plus adaptés pour gérer des langues et des dialectes complexes.

Bibliographie

[1] François-Régis Chaumartin et Pirmin Lemberger. Le traitement automatique des langues. *Préface d'Olivier Delabroy*, 2020.

[2] Natural language processing (nlp) what it is and why it matters. https://www.sas.com/en_us/insights/analytics/what-is-natural-language-processing-nlp.html?fbclid=IwAR3EMGY_u4RlQIGlJzMYBYzcXLUNj25AuZd65NdT-iJImuCxfyjHNeYmfhQ#methods. Last accessed 09 février 2023.

[3] Amazon. Qu'est-ce que le nlp ? https://aws.amazon.com/fr/what-is/nlp/?nc1=h_ls, 2023. Last accessed 09 février 2023.

[4] Chader Asma et Lanasri D et Hamdad L et Belkheir M et Hennoune W. Sentiment analysis for arabizi : Application to algerian dialect. *Ecole Nationale Superieure d'Informatique (ESI)*, 2019.

[5] NITIN INDURKHYA et FRED J. DAMERAU. Handbook of natural language processing. *by Taylor and Francis Group, LLC*, 2010.

[6] C.J Magnan. *L'Analyse Grammaticale et L'Analyse Logique*. A l'Ecole normale et A l'Ecole primaire intermediaire et superieure, 177 rue ST-joseph ST-roch, québec edition, 1907.

[7] R. Kibble. *Introduction to natural language processing*. PhD thesis, University of London International, 2013.

[8] Djerrad Maissa et Zidoune sarah. *Analyse des sentiments des tweets liés au Hirak*. PhD thesis, Université Mohamed El Bachir El Ibrahimi de Bordj Bou Arreridj, 2020-2021.

[9] Natural language processing.

[10] ZIANI Amel. *La recommandation via l'analyse d'opinions*. PhD thesis, Université de Badji Mokhtar Annaba, 2017-2018.

[11] Amazon. Analyse de sentiments. `https://www.voxco.com/fr/blog/analyse-de-sentiments/`, 2020. Last accessed 09 février 2023.

[12] Qualtrics. Utiliser l'analyse des sentiments pour améliorer les expériences. `https://www.qualtrics.com/fr/gestion-de-l-experience/etude-marche/analyse-sentiment`, 2023. Last accessed 09 février 2023.

[13] Bing Liu. Sentiment analysis : mining opinions sentiments, and emotions. *USA*, 2015.

[14] Nedioui Med Abdelhamid. *Techniques d'apprentissage automatique pour l'analyse et la fouille des sentiments dans les réseaux sociaux*. PhD thesis, Université Mohamed Khider – BISKRA, 2020-2021.

[15] *Processing Language Natural*. PhD thesis.

[16] Andrea Esuli et Fabrizio Sebastiani. Sentiwordnet : A publicly available lexical resource for opinion mining. *Proceedings of the fifth International Conference on Language Resources and Evaluation.*, 2006.

[17] G.Vinodhini et RM.Chandrasekaran. Sentiment analysis and opinion mining : A surevey. *Journal of Advanced Research in Computer Science and Software Engineering*, volume 2, 06-2012.

[18] Alexander Pak et Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining.

[19] Singh et Vivek Kumar. A clustering and opinion mining approach to socio-policical analysis of the blogosphere. *computational intelligence and computing Research (ICCIC)*, 2010.

[20] AIBOUD Lila et LASKRI Samia. *Appréciation de la qualité des leads dans le marketing numérique à l'aide de l'apprentissage profond*. PhD thesis, Université Mouloud Mammeri de Tizi-Ouzou, 11-11-2020.

[21] Chefrour Aida. *Approche hybride pour l'apprentissage automatique incrémental*. PhD thesis, UNIVERSITE BADJI MOKHTAR-ANNABA, 2019/2020.

[22] Mosteghanemi Samira et Feroukhi Afaf. *Système de recommandation multilingue basé sur l'analyse des sentiments des opinions dans les commentaires en ligne*. PhD thesis, Université Saad Dahleb Blida -1-, 06-12-2020.

[23] Walat Medhat et Ahmed Hassan et Hoda Korashy. Sentiment analysis algorithms and applications : A survey. *AIn Shams Engineering Journal*, 2014.

[24] Chaima KIHEL. *Analyse des sentiments en utilisant l'apprentissage Profond : Cas de la langue Arabe*. PhD thesis, Université de Mohamed Khider -Biskra-, 2019-2020.

[25] Saif M. Mohammad. Sentiment analysis : Detecting valence, emotions, and other affectual states from text. *journa of Emotion Measurement*, 2015.

[26] Cristóbal Colón-Ruiz et Isabel Segura-Bedmar. Comparing deep learning architectures for sentiment analysis on drug reviews. *Journal of Biomedical Informatics*, 17-08-2020.

[27] B Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 2012.

[28] et Lee. L Pang. B. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2008.

[29] Cambria. E et Hussain. A. Opinion mining, sentiment analysis, and emotion detection. *27(4), 63-79*, 2012.

[30] Rachid Ladjaj. Introduction to neural networks. `https://www.peoi.org/Courses/Coursesen/neural/frame1.html`. Last accessed 12-03-2023.

[31] Réseaux de neurones. `http://www.abdelhamid-djeffal.net/web_documents/coursrna.pdf`. Last accessed 08-03-2023.

[32] KHADIDJA MAAMOULI. *A CNN based architecture for forgery detection in administrative documents*. PhD thesis, Université de Mohamed Khider BISKRA, 06-28-2022.

[33] Yuri Musienko. Avantages et inconvÉnients de l'architecture de rÉseau neuronal. `https://merehead.com/fr/blog/avantages-inconvenients-larchitecture-reseau-neuronal/`, 08-11-2022. Last accessed 20-03-2023.

[34] ANDREI PALEYES. Neural network models explained. https://www.seldon.io/neural-network-models-explained, 04-01-2022. Last accessed 09-03-2023.

[35] Dictionnaire franÇais. https://www.linternaute.fr/dictionnaire/fr/definition/mot-vide/. Last accessed 20-04-2023.

[36] Amalia Todirascu et autres Delphine Bernhard. Problèmes de tokénisation pour deux langues régionales de france, l'alsacien et le picard. *HAL Id : hal-01539160*, 20 Jun 2018.

[37] BERROUBI ABDELAZIZ et BEN LATRACHE SAMIYA. *Vers un Système pour le Résumé Automatique des Textes Arabes*. PhD thesis, UNIVERSITÉ MOHAMED BOUDIAF - M'SILA, 2021/2022.

[38] YASSINE OURIACHI. *Prédiction Criminelle par Réseaux de Neurones*. PhD thesis, Université de Mohamed Khider BISKRA, 2020/2021.

[39] Python : définition et utilisation de ce langage informatique. https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/#:~:text=Qu'est\%2Dce\%20le\%20langage,domaine\%20du\%20d\%C3\%A9veloppement\%20de\%20logiciels. Last accessed 02-05-2023.

[40] Cours de python introduction à la programmation python pour la biologie. https://python.sdv.univ-paris-diderot.fr/, version du 29 août 2022. Université Paris Cité, France.

[41] Python : Focus sur le langage le plus populaire. https://datascientest.com/python-tout-savoir#:~:text=Quels\%2Dce\%20le\%20langage,domaine\%20du\%20d\%C3\%A9veloppement\%20de\%20logiciels. Last accessed 02-05-2023.

[42] ZOUAOUI RANIA. *Blockchain pour gestion des données médicales*. PhD thesis, UNIVERSITÉ MOHAMED BOUDIAF - M'SILA, 2020/2021.

[43] Ntlk : guide de l'outil de traitement naturel du langage en python. https://datascientest.com/nltk, 6 Avr. Last accessed 15 février 2023.

[44] Maîtrisez l'analyse des données avec numpy python. https://www.data-transitionnumerique.com/numpy-python/. Last accessed 03-05-2023.

[45] MAMEN ABDELKARIM. *Développement d'une architecture CNN pour la classification des images radiologiques d'infections pulmonaires*. PhD thesis, Université de Mohamed Khider BISKRA, 2020/2021.

[46] *apprenez keras*. info@zzzprojects.com.

[47] Aide-mémoire sur les pandas. https://geekyhumans.com/fr/aide-memoire-sur-les-pandas/, 24-05-2022.