



République Algérienne Démocratique et Populaire

Ministère de L'enseignement Supérieur et de la Recherche Scientifique

Université Abbes Laghrour Khenchela

Faculté des Sciences et Technologie

Département des Mathématiques et Informatique

**Mémoire Présenté pour l'obtention
Du diplôme de Master Académique**

Option : Sécurité et Technologie Web

Thème

***UN MODELE DE DEEP LEARNING POUR L'ANALYSE
DES SENTIMENTS***

Encadré par

➤ BAKHOUCHE Abdelaali

Réalisé par

* MEGHNI Cherif

* TAFERGHOUST Ahmed

Année universitaire :2020/2021

Remerciements

Nous tenons tout d'abord à remercier Allah, qui nous a donné la force et la patience d'accomplir ce Modeste travail.

En second lieu, nous tenons à remercier notre encadreur Mme: BAKHOUCHE Abdelaali son précieux conseil et son aide durant toute la période du travail.

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

De plus, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours encouragés au cours de la réalisation de ce mémoire.

Notre reconnaissance va aussi à tous ceux qui ont collaboré à notre formation en particulier les enseignants du département d'Informatique, Université abbes-laghrour khenchela

Aussi à nos collègues de la promotion 2020-2021.

Merci à tous et à toutes.

<i>Introduction générale</i>	-1-
CHAPITRE 1 :Analyse des sentiments	
1.1. <i>Introduction</i>	-2-
1.2. <i>opinion mining</i>	-4-
1.2.1. <i>Analyse des sentiments</i>	-4-
1.2.2. <i>La polarité et l'intensité de l'opinion</i>	-4-
1.2.3. <i>Les défis d'analyse les sentiments</i>	-5-
1.2.4. <i>Difficultés d'Analyse des Sentiments</i>	-5-
1.2.5. <i>Domaine d'application d'analyse des sentiments</i>	-6-
1-3- <i>La langue arabe et le dialecte</i>	-6-
1-3-1- <i>Voyellation</i>	-7-
1-3-2- <i>La richesse de la langue arabe</i>	-7-
1-3-3 - <i>Eléments de structure de la langue arabe</i>	-7-
1-3-4 <i>Arabe algérien</i>	-7-
1-3-5- <i>Les difficultés de l'arabe et du dialecte</i>	-8-
1-4- <i>Les approches d'analyse des sentiments</i>	-8-
1-4-1. <i>Approches basées sur le lexique</i>	-8-
1-4-1-1. <i>Approche Basée Sur Le Dictionnaire</i>	-9-
1-4-1-2 <i>Approche Basée Sur Le Corpus</i>	-9-
1-5 <i>Approche D'apprentissage Automatique</i>	-9-
1-5-1. <i>Apprentissage non supervisé</i>	-9-
1-5-2. <i>Apprentissage supervisé</i>	-9-
1.6. <i>Conclusion</i>	-9-
CHAPITRE 2 : Deep learning	
2.1. <i>Introduction</i>	-11-
2-2- <i>Les applications du Deep Learning</i>	-14-
2-2-1- <i>La reconnaissance faciale :</i>	-14-
2-2-2- <i>Le traitement automatique de langage naturel</i>	-14-
2.2.3. <i>Voitures autonomes :</i>	-14-
2.2.4. <i>Recherche vocale et assistants à commande vocale :</i>	-14-
2.2.5. <i>Ajout automatique de sons à des films muets :</i>	-14-
2.2.6. <i>traduction automatique :</i>	-14-
2.2.7. <i>Génération automatique de texte :</i>	-15-
2.2.8. <i>Reconnaissance d'image :</i>	-15-
2.2.9. <i>la description automatique d'image :</i>	-15-
2.2.10. <i>Colorisation automatique :</i>	-15-
2.2.11. <i>la détection du cancer du cerveau :</i>	-15-
2.2.12. <i>Analyse des sentiments du texte :</i>	-15-
2.2.13. <i>Recherche en marketing</i>	-16-
2.3 <i>Le réseau neuronal</i>	-16-
2.3.1 <i>Le neurone</i>	-16-
2.3.2 <i>Les fonctions d'activation :</i>	-17-
2.3.2.1 <i>La fonction Sigmoidé :</i>	-17-
2.3.2.2 <i>La fonction ReLu :</i>	-17-
2.3.2.3 <i>La fonction Softmax :</i>	-17-
2.3.3 <i>Les architectures des réseaux de neurones :</i>	-18-
2.3.3.1 <i>Les réseaux entièrement connectés :</i>	-18-

2.3.3.2 Les réseaux convolutionnels :	-19-
2.4 L'apprentissage en Deep Learning	-19-
2.4.1 Introduction	-19-
2.4.2 Les variantes de la descente de gradient	-20-
2.4.3 Les Algorithmes d'optimisation de la descente de gradient	-20-
2-5- Algorithme d'optimisation par essaim de particules (PSO)	-21-
2-5-1- Introduction	-21-
2-5-2- Notion fondamentale	-21-
2-5-3- Présentation de l'algorithme PSO	-21-
2-5-4- Mode de fonctionnement de l'algorithme PSO	-22-
2-5-5- Notion de voisinage	-23-
2-5-6- Fonctionnement de l'algorithme PSO	-25-
2-5-7- Améliorations apportées au PSO	-27-
2-6- L'analyse du sentiment et le Deep Learning	-30-
2-7- Conclusion	-31-
CHAPITRE 3 : Conception et implémentation	
3- Introduction	-33-
Partie 1	
3-1- Conception	-33 -
3-1-1- Problématique	-33-
3-1-2- Contribution	-33-
3-1-3- Source de données	-33-
3-1-3-1- Caractéristiques des tweets	-34-
3-1-3-1-1- Longueur :	-34-
3-1-3-1-2- Disponibilité des données et modèle du langage :	- 34 -
3-1-4- Annotation	-34-
3-1-5- Création de dictionnaire	-35-
3-1-6- Architecture générale	-35-
3-1-7- Architecture détaillée	-36 -
3-1-8- Méthodologie de recherche	- 36 -
3-1-8-1- Prétraitement de texte :	- 36 -
3-1-8-2- Calcul des scores informatifs	-37-
3-1-8-3- Calcul des scores sémantiques	- 38-
3-1-8-4- Construire les solutions candidates Graph	- 39-
3-1-8-5- Application de l'algorithme PSO	- 39-
3-1-9-. Résultats expérimentaux et évaluation	- 40-
Partie 2	
3-2 implémentation	-41-
3-2-1- Réalisation de système	-41-
3-2-1-1- Ressources utilisées	- 41-
3-2-1-2- Plate-forme utilisé	- 41-
3-2-2- Choix du langage de programmation	- 42 -
3-2-3- Exemples de codes sources	-42-
3-2-4- Exemples de sorties	-45-
3.3. Conclusion	-45-
4-. Conclusion générale	-46-
5-. Bibliographie	-47-

liste des figures

<i>Figure I-1 : les approches d'analyse des sentiments</i>	- 08 -
<i>Figure 2.1 – Les progrès de l'Intelligence Artificielle</i>	- 12 -
<i>Figure 2.2 – La relation entre les 3 concepts</i>	-13 -
<i>Figure 2.3 – Un neurone réel</i>	-16-
<i>Figure 2.4 – Un neurone artificiel</i>	-16-
<i>Figure 2.5 – Représentation graphique de la fonction Sigmoidale</i>	-17-
<i>Figure 2.6 – Représentation graphique de la fonction Relu</i>	-18-
<i>Figure 2.7 – Un réseau entièrement connecté</i>	-18 -
<i>Figure 2.8 – Une illustration du processus de recherche de l'optimum</i>	-19-
<i>Figure 2-9- : Volée d'Anser en formation en V</i>	-21-
<i>Figure 2-10 : Déplacement d'une particule.</i>	-22-
<i>Figure 2-11 : Voisinage géographique à l' instant t et t+1.</i>	-24-
<i>Figure 2-12 : Voisinage en cercle (choix régulier des informatrices).</i>	-24-
<i>Figure 2-13 : Voisinage en cercle (choix aléatoire des informatrices).</i>	-25-
<i>Figure 2-14 : Algorithme d'optimisation par essaim particulaire.</i>	-26-
<i>Figure 2-15 : Organigramme « méthode des essais particuliers »</i>	-27-
<i>Figure 3-1 : Représentation de l'architecture générale du système.</i>	-35-
<i>Figure 3-2- : Représentation de l'architecture détaillée de système.</i>	-36-
<i>Figure 3-3- : Logo anaconda Navigator</i>	- 41-
<i>Figure 3-4- Logo langage python</i>	- 42-
<i>Figure 3-5- import les bibliothèques</i>	- 43-
<i>Figure 3-6- lire le Dataset</i>	- 43-
<i>Figure 3-7 Pré-traitement</i>	- 44-
<i>Figure 3-8 compter les mots négatifs</i>	-44-
<i>Figure 3-9 : Exemple de niveau de sentiments</i>	-45-

liste des tableaux

Tableau 2.1 – <i>Le résumé de l’histoire de deep learning</i>	-13-
Tableau 2.2 – <i>Résumé des résultats obtenus avec la robustesse comme une métrique d’évaluation</i>	-30-
Tableau 2.3 – <i>Résumé des résultats basant sur la robustesse comme une métrique d’évaluation</i>	-31-
Tableau 3.1 : <i>Données collectées.</i>	-34-
Tableau 3.2 : <i>Nombre de commentaires par polarité.</i>	-34-
Tableau 3.3 : <i>Exemple de notation de quelques commentaires</i>	-34-
Tableau 3.4 : <i>Statistiques de notre dictionnaire</i>	-35-
Tableau 3-5-: <i>PSO Paramétrage</i>	-40-
Tableau 3-6-: <i>résumé les quatre résultats de l’approche pour ROUGE -1 et ROUGE-2</i>	-40-
Tableau 3-7 <i>Exemple de sortie</i>	-45-

ملخص

يعتبر تحليل النصوص هو أحد أكبر التحديات التي تواجه خبراء تحليل البيانات، فالبيانات الرقمية مهما كانت كبيرة أو معقدة إلا أنه يمكن إخضاعها لعمليات حسابية ومعادلات رياضية بكل سهولة لاستخلاص النتائج منها. وفي زمن السوشيال ميديا وطفرة شبكات التواصل الاجتماعي التي تتيح للناس التعبير عن آرائهم بلغتهم الخاصة وبتعبيراتهم المتعددة فإن الحاجة لا تزال ملحة لتطوير خوارزميات خاصة بتحليل المعاني وبالتالي استخلاص المشاعر والانطباعات التي عكسها المستخدمون في كتاباتهم.

تهتم الشركات ومراكز دعم اتخاذ القرار بأراء زبائنهم والفئات المستهدفة بالدراسة والتحليل، لذا فإن معرفة ما يعجبهم أو يكرهونه يمثل قيمة عالية بالنسبة لاتخاذ أي قرار قد يزيد من الحصة السوقية للشركة أو يدفع باتجاه اتخاذ قرار لمؤسسة مجتمع مدني أو حتى جهة حكومية.

ما هو تحليل المشاعر والآراء *Sentiment Analysis* ؟

إن تحليل المشاعر والآراء هو عملية معقدة لتحليل البيانات النصية وتحديد وجهات النظر والآراء إلى اتجاهات إيجابية أو سلبية أو محايدة. ويمكن القول أن تحليل المشاعر والآراء هو عملية تلقائية لتحديد واستخراج المعلومات الشخصية التي تشكل أساس النص، وهو ما يمكن أن يكون رأياً في موضوع معين أو حكماً على موقف معين أو شعوراً اتجاه حديث معين. وعند أهل التحليل يطلقون عليه مسمى القطبية، بمعنى أن يكون الشعور أحد ثلاثة اتجاهات: اتجاه إيجابي، أو سلبي، أو محايد، والأخير أحياناً يمكن اعتباره حالة من عدم تكوين الرأي أو الموقف أو التردد في بعض الأحيان، وهذا يرجع إلى طبيعة البيانات والموضوع المراد رصده وتحليله.

ما يقرب من **80% من البيانات الرقمية في العالم غير منظمة**، أي أنها تحتوي على أرقام ورموز وتعبيرات وحروف مختلطة وليست على نسق محدد كما في جداول إكسيل مثلاً أو قوائم المسافرين، رغم أنها أيضاً قد تكون بحاجة لعمليات تنظيف وتهيئة قبل المعالجة. العبارات التالية: “تصميم الهاتف الجديد سيء ولا يلبي حاجتي”، “بعض المميزات لا تعمل والتصميم القديم كان أفضل”، “غير جيد ولن أهدر مزيداً من المال لمجرد هاتف؟” يمكن لخوارزميات تحليل المشاعر أن تميزها بأنها سلبية، وهكذا بالنسبة للجمل التي تحمل مفردات إيجابية. طبعاً هذه عملية ليست سهلة ومباشرة بقدر ما هي خطوات متعددة ومتسلسلة وتعتمد على تحليل مركبات كل جملة وتكرار كل كلمة مع اعتبار ما قبلها وما بعدها، وذلك يستلزم استحضار اعتبارات اللغة وقواعد كتابتها ومعانيها. يضاف إلى كل ذلك أن تكون الكتابة باللغة العامية أو الرسمية أو حتى لغة الدردشة والتي تعتمد كثيراً على الاختصارات والرموز!!

يمكن للشبكات العصبية حل المشاكل بنفس الطريقة التي يمكن حلها من قبل البشر .

في عملنا، استخدمنا هذا النوع من الشبكات لتصنيف التغريدات وتحليل مشاعر المستخدمين.

لذلك اقترحنا مجموعة متنوعة من النماذج المختلفة للحصول على نتائج جيدة.

1-Introduction général

Au cours des dernières années, un grand nombre de personnes ont été attirées par les plateformes de réseau social comme Facebook, Twitter et Instagram. La plupart utilisent des réseaux sociaux pour exprimer leurs émotions, leurs croyances ou leurs opinions sur des choses, des lieux ou des personnalités.

Actuellement, l'un des meilleurs exemples de réseaux sociaux permettant d'observer l'évolution de ces opinions est Twitter [1], du fait de la concision et de la structure des messages publics partagés sur cette plate-forme.

L'analyse des sentiments est récemment devenue l'un des domaines de recherche en croissance liés au traitement du langage naturel et à l'apprentissage automatique.

Elle s'est avérée bénéfique pour plusieurs tâches de traitement du langage naturel (TAL) telles que les systèmes de réponse et l'extraction de l'information. L'extraction d'informations a pour but d'extraire une information pertinente à un sujet particulier ou aux besoins de l'utilisateur. Par exemple, les gens ont tendance à utiliser Internet de nos jours pour diffuser leurs pensées et leurs idées sur des sujets ou des problèmes en utilisant des forums ou d'autres réseaux sociaux. Certaines de ces idées sont positives tandis que d'autres sont plus violentes dans la manière et le contenu.

Par exemple, Twitter qui nous intéresse particulièrement dans ce travail, constitue une source continue et illimitée de données en langage naturel qui est particulièrement difficile à traiter avec les approches classiques de traitement automatique du langage naturel (TAL).

Ce type de langage est très éloigné des normes du langage traditionnel, avec ses conventions (telles que les hashtags, les mentions, les retweet, etc.).

Son lexique particulier est souvent grossier et contient des abréviations, des émoticons, des acronymes.

Sa syntaxe est parcellaire dans le meilleur des cas. Les données extraites de Twitter sont hautement bruitées non-structurées, et courtes (comportant au maximum 280 caractères par tweet).

Dans notre projet nous procédons à l'analyse des sentiments sur Twitter.

Ce document est commencé par une introduction général sur notre travaille, il contient trois chapitres : le premier chapitre sur l'analyse des sentiments en générale, le deuxième sera une description de l'apprentissage approfondi, de ses applications et de ses fondements et pour le dernier chapitre à propos de la réalisation et les résultats, et finalement une conclusion générale.

Chapitre I

Analyse des sentiments

1-1-Introduction

Aujourd'hui, avec l'augmentation de la quantité de données qui circulent sur le web, les utilisateurs partagent leurs sentiments, leurs opinions ou leurs points de vue sur l'Internet presque partout, sous forme de commentaires, questions ou tout autre forme d'expression.

Nous vivons dans une époque où l'information est disponible en grande quantité et sous diverses formes la forme textuelle représente 80% de cette masse d'information, mais le volume massif de ces données n'a aucune utilité si nous ne les explorons pas d'une manière spécifique en utilisant des méthodes visant à extraire une information significative[1].

L'analyse du sentiment, aussi appelée opinion mining, est un sujet qui se concentre sur la réalisation de ce type de tâches. Il existe de nombreux autres noms apparentés se référant légèrement à différentes tâches telles que l'extraction d'opinion, le sentiment mining, l'analyse de subjectivité, l'analyse d'affect l'analyse d'émotion, les avis mining, etc.

Cependant, ils font maintenant tous partie de l'analyse des sentiments ou de l'opinion mining.

Dans l'industrie, le terme d'analyse du sentiment est couramment le plus utilisé, mais dans le milieu universitaire, on utilise fréquemment à la fois l'analyse du sentiment et l'opinion mining [1].

Ce sujet suscite beaucoup d'intérêt parmi des nombreux chercheurs qui ont apporté des contributions à l'aide de diverses techniques.

Nous trouvons de nombreux travaux qui ont commencé plus tôt en 1997 un travail visant à prédire l'orientation sémantique des adjectifs[2], en 2000 un travail proposé également axé sur l'orientation adjectifs et leurs effets sur les phrases[3], en 2001 on trouve deux travaux, le premier est un modèle d'un système opérationnel pour détecter et suivre les opinions[6], le deuxième est un modèle suggéré pour obtenir les opinions des gens à partir des messages[5], en 2004 un projet qui représente un modèle d'extraction de sentiments[4], etc.

Parce qu'elle se concentre sur les données textuelles, l'analyse du sentiment est largement utilisée dans d'autres domaines que le traitement automatique du langage (TAL) comme le data mining, le texte mining et la recherche d'information.

Nomenclature des notions

Sentiment : Expression de la sensibilité, d'un penchant, d'une affection, d'une passion. Synonymes : avis, opinion [4].

Opinion : Jugement, avis, sentiment qu'un individu ou un groupe émet sur un sujet, des faits, ce qu'il en pense [5].

Réseau social : L'ensemble des sites internet qui permettent de relier des personnes physiques ou personnes morales entre elles pour partager des textes, photos, vidéos... [6].

Lexique : Est un ensemble de mots d'une langue (ou dialecte), généralement est un dictionnaire spécialisé dans un domaine particulier [7].

Corpus : Un corpus est un ensemble fini de textes préparé pour but d'analyse, spécifiquement collectivement fini d'énoncés considérés comme particularité du type de langue à exploiter [8].

Apprentissage automatique (Machine Learning) : Est un type d'intelligence artificielle qui donne aux ordinateurs la capacité d'apprendre sans être programmés et de trouver des solutions aux problèmes qui ne sont pas résolus par des algorithmes classiques [9].

Traitement automatique des langues naturelles : C'est de faire parler les ordinateurs, et plus précisément à leur donner les capacités linguistiques d'un être humain [10].

Machine à vecteurs de support (SVM Support Vector Machines) : Sont des techniques d'apprentissage statistique proposées par V. Vapnik en 1995 [11].

Elles permettent de résoudre des problèmes tel que la régression, la fusion, ... et d'une part particulière le classement.

Sa principale idée consiste à projeter l'espace d'entrée (données à deux classes différentes non séparables de façon linéaire) dans un espace à grande dimension (espace de caractéristiques) où les données deviennent linéairement séparables, dans ce nouvel espace, la technique de construction de l'hyperplan optimal est utilisée pour séparer les deux classes par une fonction de classement [12].

Arbre de décision (DT Decision Tree) : Un arbre de décision est un graphe non orienté, acyclique et connexe, son ensemble de nœuds est divisé en trois catégories, nœud racine, nœuds internes (enfants) et nœuds terminaux (feuilles), Les arbres de décision (AD) sont utilisés dans l'exploration de données pour prendre une décision [13].

Forêt d'Arbres Décisionnels (RF Random Forest) : Le terme Forêts Aléatoires est un ensemble d'arbres de décision. Où chaque arbre a pour but de décomposer le problème en suite de tests correspondant à partitionner l'espace de données en sous-régions homogènes (en terme de classe), aller de la racine à une feuille en effectuant les tests de nœuds où la classe d'une feuille est la majoritaire parmi les exemples d'apprentissage appartenant à cette feuille [14].

Naïf Bayésien (NB Naive Bayes) : Un classificateur naïf de Bayes est un classificateur probabiliste basé sur l'application du théorème de Bayes [15] avec l'hypothèse naïve (forte indépendance), c'est-à-dire que les variables explicatives (X_i) sont supposées indépendantes conditionnellement à la variable cible (C) il appartient à la famille des classificateurs linéaires (son rôle est de classer dans des classes les échantillons qui ont des propriétés similaires). Ce classificateur est souvent utilisé sur les flux de données pour la classification supervisée [16].

Mesures de performance [17] : Soit : PS : documents pertinents sélectionnés,

PN : documents pertinents non sélectionnés, NS : documents non pertinents sélectionnés, NN : documents non pertinents non sélectionnés.

Nous avons : **Rappel** (recall) = $PS / (PS + PN)$

Précision = $PS / (PS + NS)$, **F-mesure** (F-score) = $2 * \text{précision} * \text{rappel} / (\text{précision} + \text{rappel})$

Accuracy = $(PS + NN) / (PS + NS + PN + NN)$ **N-grammes** : Comme leur nom l'indique, les *n-grams* sont des suites d'un nombre donné de caractères (éventuellement de mots) [18]. À titre d'exemple, le bigramme le plus fréquent de la langue française est « de ».

Fonctionnalité : Une caractéristique ou une propriété qui nous permet de prédire l'appartenance d'un élément à une classe donnée [19]. **MSA** : Modern Standard Arabic, qui est enseignée dans les écoles contemporaines [20].

❖ **Le Microblogging** : Le Microblogage (microblogging en anglais) permet à des internautes de publier des messages courts pour exprimer une opinion, donner un avis ou encore partager un contenu ou une information en temps réel. L'ensemble de ces messages constitue un flux. [5]

❖ **Phrase subjective** : C'est une phrase dans laquelle l'écrivain exprime ses sentiments envers les entités les événements et leurs propriétés.

❖ **Phrase objective** : une phrase objective indique des faits et des informations connues propos du monde.

❖ **Opinion** : C'est un avis ou un jugement basé sur des connaissances spéciales sur un sujet. Les opinions sont parfois exprimées clairement comme : "La qualité vocale de ce téléphone est étonnant. "Mais parfois ils sont cachés dans le sentiment d'une phrase, par exemple : "L'écouteur s'est cassé en deux jours.

Dans cette section, nous allons définir une nomenclature des concepts utilisés dans ce mémoire pour faire l'éclaircir avant de détailler notre travail.

1-2- Opinion mining

L'Opinion Mining devrait "traiter un ensemble de résultats de recherche pour un cas donné, générer une liste des attributs (qualité, caractéristiques, etc.) et agréger des avis sur chacun d'entre eux (mauvais, modéré de bonne qualité).

Les principales tâches de l'Opinion Mining sont l'analyse de l'opinion et l'analyse de la subjectivité. Cette dernière est utilisée pour reconnaître le langage décrit l'opinion afin de distinguer les langues objectives.

Mais comme dans toutes applications qui traitent du texte, la difficulté majeure est d'arriver à interpréter le langage humain qui reste, encore aujourd'hui, un objet complexe, tant par sa richesse que par ses subtilités.

Il est très difficile par exemple de détecter de façon informatique (et même humaine parfois !) l'ironie.

1-2-1 Analyse des sentiments

L'analyse de sentiment est l'étude computationnelle et sémantique des parties de textes en fonction des opinions, des sentiments et des émotions exprimés dans le texte.

Généralement l'expression « analyse des sentiments » est utilisée pour désigner la tâche de classification automatique des unités de texte en fonction de leur polarité (positive, négative, neutre).

1-2-2-. La polarité et l'intensité de l'opinion

La polarité peut être définie par des catégories telles que « positif », « neutre », et « négatif »

La polarité d'une opinion exprime la positivité, la négativité ou une information de cette dernière.

On dit d'une opinion positive qu'elle possède une polarité positive, et inversement, on dit d'une opinion négative qu'elle possède une polarité négative ou neutre possède une information

L'intensité décrit à quel point la polarité d'une opinion est forte. Par exemple, dans une opinion à polarité positive, aimer est plus intense qu'apprécier, ou encore, dans une opinion de polarité négative, haïr est plus intense que de ne pas aimer [01].

1-2-3- Les défis d'analyse les sentiments

Lorsque ont traité ce domaine ont trouvé des défis comme :

- ❖ **L'état émotionnel du locuteur** : Les sentiments du locuteur peuvent être compatibles ou contraires aux déclarations faites par le locuteur.
- ❖ **Succès ou échec d'un côté avec respect à un autre** : Exemple 'Yay! France beat Germany 3-1': si en a un supporteur de la France émotion positif, si en a un supporteur de l'Allemagne émotion négative.
- ❖ **Déclaration neutre des informations validées** : Si en a aucune indication sur l'émotion mais en a des citations valides, donc on ne sait pas et ce que en a des déclarations neutre ou une déclaration négatif d'émotion.
- ❖ **Sarcasme** : Est de déclarer des sentiments positifs, même s'ils sont négatifs. Exemple : « croyez-vous vraiment ce que vous dites ? »
- ❖ **Différent sentiment vers différent cible d'opinion** : L'orateur peut exprimer l'opinion à propos de cibles multiples, et le sentiment envers les différentes cibles pourrait être différent.
- ❖ **Déterminer précisément la cible de l'opinion** : Parfois, il est difficile de préciser identifier la cible d'opinion.

Par exemple, sur un sujet relatif aux conditions de travail ou aux prestations du restaurant, c'est le salarié concerné qui sera interrogé. [02]

1-2-4-. Difficultés d'Analyse des Sentiments

L'extraction du sentiment ou d'opinion consiste à déterminer la polarité d'une telle opinion. Dans ce qui suit nous citons quelques difficultés de cette procédure. [03]

- Ambiguïté de certains mots positifs ou négatifs selon les contextes et qui ne peut pas toujours être levée.
- Difficulté due aux structures syntaxiques et sémantiques d'une phrase et l'expression de l'opinion. Par exemple " l'histoire du film est intéressante mais les acteurs étaient mauvais ". Dans ce cas la polarité de la deuxième partie est opposée à la première.
- Difficulté due au contexte : la nécessité d'une bonne analyse syntaxique du texte ; analyse qui peut se révéler particulièrement difficile dans des cas de coordination entre plusieurs parties d'une phrase. Par exemple "ma tante a bien préparé le gâteau, son décor est bonne mais je n'ai pas aimée le goût", l'opinion de la dernière partie de la phrase est la plus importante.

- Difficulté due à l'analyse de la phrase par " paquets de mots ". Les deux phrases suivantes contiennent les mêmes paquets de mots sans pour autant exprimer les mêmes sentiments.

La première phrase contient un sentiment positif alors que la deuxième est négative : " Je l'ai apprécié pas seulement à cause de ...", " Je l'ai pas apprécié seulement à cause de ... " ou se présente la gestion de négation.

- Difficulté due au langage qu'utilisent les internautes pour s'exprimer.

Les ponctuations ne sont pas forcément utilisées pour marquer les fins de phrases, des mots spécifiques sont utilisés tel que : «ha ha ha», «Good», «super».

– Difficulté de déterminer un lexique adapté à l'analyse de l'ensemble des textes d'opinion.

1-2-5-. Domaine d'application d'analyse des sentiments

❖ **Politique** : Grâce à l'analyse des sentiments, les décideurs de politique pouvant prendre l'avis des citoyens sur certaines politiques, afin de bénéficier de cette information pour améliorer ou créer une nouvelle politique qui convient avec les citoyens.

❖ **Prise de décision** : L'opinion et l'expérience des gens sont un élément très utile dans le processus de prise de décision.

❖ **Les systèmes de recommandations** : À travers l'analyse des sentiments on peut classer les opinions des gens en positives ou négatives, le système définit qui devrait prendre la recommandation et qui ne devrait pas prendre la recommandation.

❖ **Domaine de Transport**: Pour assembler et analyser les opinions de public sur le statut de transport. Exemple : Système de transport intelligent moderne.

❖ **Domaine médical** : Analyse l'opinion des médecins, patients sur les médicaments et les services hospitaliers. Ainsi sur les documents de l'état de patient qui contiennent le diagnostic et la description du résultat d'examen.

❖ **Domaine éducation** : Développer le niveau d'enseignement à travers l'analyse et l'interprétation de l'opinion de l'étudiant à travers les méthodes d'enseignement est ce qui permet d'améliorer l'enseignement et l'apprentissage.

❖ **Marketing** : Du côté entreprises, permet au fournisseur plus de connaissances à propos des besoins des consommateurs, du côté client il peut donner son opinion, s'inspirer des opinions d'autres clients pour l'aider à sa décision et aussi comparer les produits avant de les acquérir. [04]

1-3- La langue arabe et le dialecte

Les langues sémitiques (en référence au nom de "**Sem**" fils de "**Noé**") forment un ensemble de langues parlées depuis le plus ancien temps au Moyen-Orient, au Proche-Orient ainsi qu'en Afrique du Nord, La langue sémitique est une des branches de la famille des langues afro-asiatiques, où on ne sait pas de manière certaine l'origine et l'expansion géographique de ces langues, soit de l'Asie vers l'Afrique ou le contraire. L'origine du mot "*Arabe*" reste inconnu, malgré il y a beaucoup de recherches, le radical "*arab*" désigne

le désert et c'est un mot araméen "arâbâh", le mot arabe peut dériver aussi de la racine sémitique Abhar "se déplacer", mais l'étymologie arabe considère que le mot "arabe" est dérivé du verbe "exprimer" [51].

1-3-1- Voyellation

Dans la langue arabe nous avons deux types de signes de notation des voyelles, le premier type sont les voyelles brèves, qui sont notées au moyen de signes diacritiques, le deuxième type sont les lettres d'allongement d'une voyelle.

Les voyelles brèves sont : fatha, damma et kassra, se rajoutent sur les lettres, pour donner le vrai sens (la prononciation précise) de chaque mot et pour éviter les ambiguïtés (les conflits) lorsque le contexte n'est pas suffi, il y a aussi des autres diacritiques moins utilisés comme sukun, shadda et tanwn. Les voyelles longues (lettres d'allongement) sont : "Alif", "waw" et "ya", avec les sons respectivement "aa", "ou", "ii" [52].

1-3-2-La richesse de la langue arabe La langue arabe s'écrit au moyen de 28 lettres, le terme

"أ ب ج د" désigne le système d'écriture, elle dispose le plus grand nombre de mots avec plus de 12 millions de mots où l'Anglais 600 000 mots, le Français 150 000, mots, le Russe 130 000 mots...

Elle est une langue très riche, Il y aurait 80 termes différents pour identifier le miel, 200 pour le serpent, 500 pour le lion, 1000 pour le chameau et l'épée et jusqu'à 4400 pour définir l'idée de malheur, les grammairiens arabes prétendent que toutes les racines ont été originalement des verbes, où le nombre de ces racines en réalité est de 6000 [53].

1-3-3 -Eléments de structure de la langue arabe Comme toutes les langues sémitiques, l'arabe se caractérise par l'utilisation de certains schémas morphologiques (modèles de formation des mots), ces modèles permettent de fournir des mots à partir de racines abstraites, ces racines se composent d'habitude de trois consonnes qui forment les unités de base pour obtenir de nombreux mots dérivés de cette racine. On trouve par exemple la racine KTB se retrouve dans le verbe dérivé KaTaB "à écrire" qui peut être conjugué en ajoutant de préfixes et de suffixes convenables. Le schéma KaTaB peut être représentée par le schéma $C_1aC_2aC_3$ où $C_1 = K$, $C_2 = T$, et $C_3 = B$. Le doublement de la consonne médiane de la racine donne " $C_1aC_2C_2aC_3$ " qui se traduit par le schéma $KaTTaB$ qui a le sens causal "a fait écrire". Par allongement de la première voyelle on obtient $KaaTaB$, par l'ajout du préfixe "ta" nous aurons taKaaTaB (notion de réciprocité). Chacun de ces schémas peut être modifié pour montrer le passé, le présent..., par exemple KuTiB "été écrit", yaKTuB "écrit" et yuKTaB "est écrit. Ainsi que le pluriel, par exemple Kutub "des livres", Kuttab "des écrivains", maKTabaat "des bibliothèques", en plus, quelques-uns peuvent être mis au féminin par des terminaisons dédiées, par exemple KaaTiBa "une femme écrivain", muKaaTiBa "correspondantes" et KaaTiBaat "des femmes écrivains". Ces exemples présentent un modèle riche et varié de schémas dérivés d'un seul mot qui permet d'engendrer des centaines de mots ont référencé à la même racine [54].

1-3-4- Arabe algérien

L'Arabe Algérien ou bien le dialecte algérien, est un groupe d'Arabe Nord-Africain, dialectes mélangés avec

différentes langues parlées en Algérie, le frottement de plusieurs langues à travers l'histoire de la région a produit un langage complexe et riche comprenant des mots, des expressions et des structures linguistiques de ces langues tel que le Berbère, le Français, l'Italien, l'Espagnol et le Turc ainsi que des autres langues romanes méditerranéennes. Le dialecte algérien est fortement influencé particulièrement par le français où on peut trouver la commutation de codes et emprunter [55].

Nous pouvons catégoriser les paroles en ce dialecte comme la suite :

MSA encodé en lettres arabiques, par exemple : « بخير وانتم عام كل مبارك عيدكم »

MSA encodé en lettres romanisées, par exemple : « ana said hada elyaoum »

Dialecte encodé en lettres arabiques, par exemple : « شريكي كيراك »

Dialecte encodé en lettres romanisées, par exemple : « maysoumouch bla shour »

Langue étrangère encodée en lettres romanisées, par exemple : « je vais à l'univ »

Langue étrangère encodée en lettres arabiques, par exemple : « قوووول »

Donc chaque texte est l'une de ces catégories ou un mélange d'eux, on peut aussi trouver une autre écriture particulière celle l'utilisation d'un numéro au lieu d'une lettre par exemple : « b1-بيا- bien », « 3adi-عادي - normal », « 5amsa-خمسه- cinq », « 6ariq-طريق- route », « sa7b-صاحب- ami », « 9raya-قراية-étude »...

1-3-5- Les difficultés de l'arabe et du dialecte

Nous pouvons citer les difficultés suivantes :

- Un seul dialecte (par exemple le dialecte algérien) peut contenir plusieurs sous dialectes.
- La grande distance entre MSA et quelques dialectes.
- Une racine peut prendre plusieurs formes en fonction du contexte.
- La répétition d'une lettre plusieurs fois pour intensifier le sens ou le sentiment (bzeeeeef).
- L'arabe a divers signes diacritiques, la présence ou l'absence de tels signes, peut changer totalement le sens des mots.
- Les mots de négation utilisés pour nier les verbes au passé ou au présent, qui changent le sens exactement au contraire [56].

1-4- Les approches d'analyse des sentiments

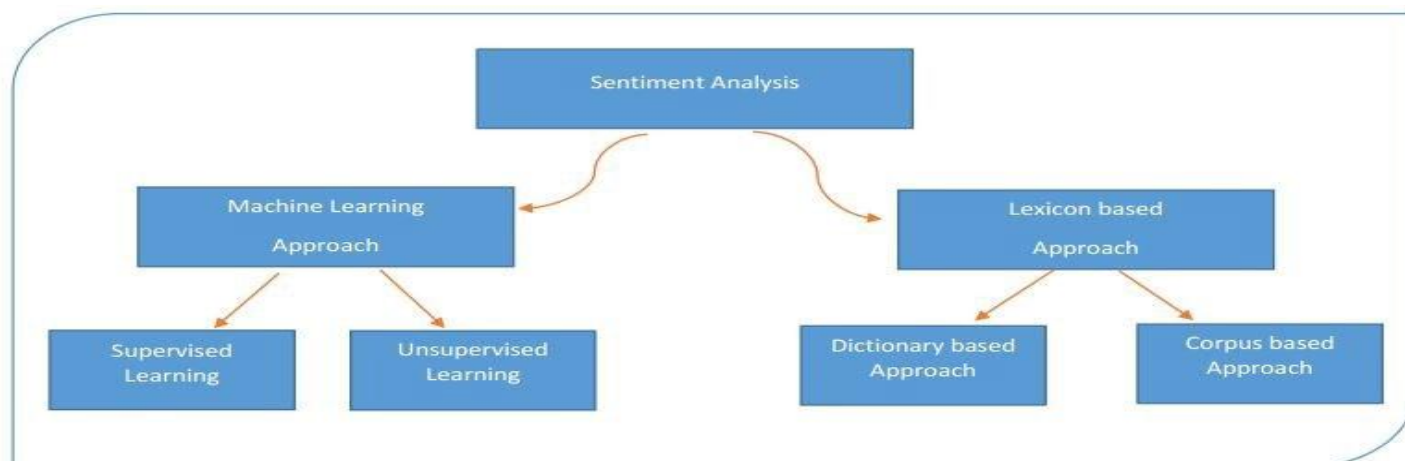


Figure I-1 : les approches d'analyse des sentiments

1-4-1- Approches basées sur le lexique

La méthode basée sur le lexique. Il utilise un lexique composé de termes avec des scores de sentiment respectifs pour chaque terme. Le terme peut être associé à un seul mot ou une phrase. Le sentiment est défini en fonction de la présence ou de l'absence de termes dans le lexique. L'approche basée sur le lexique inclut l'approche basée sur le corpus et l'approche basée sur le dictionnaire.

1-4-1-1- Approche Basée Sur Le Dictionnaire

L'idée principale derrière l'approche basée sur un dictionnaire est d'utiliser des bases de données lexicales avec des mots d'opinion pour extraire le sentiment du document. La procédure de recherche est itérative.

À chaque itération, l'algorithme prend un ensemble de mots mis à jour (ensemble étendu) et effectue une nouvelle recherche jusqu'à ce qu'il n'y ait plus de nouveaux mots à inclure. En fin de compte un ensemble de mots de sentiment peut être examiné dans le but de supprimer les erreurs.

1-4-1-2-. Approche Basée Sur Le Corpus

Dans Bing Liu [05] indique que l'approche basée sur le corpus peut être appliquée dans deux cas. Le premier cas est une identification des mots d'opinion et de leurs polarités dans le corpus de domaine en utilisant un ensemble donné de mots d'opinion. Le second cas concerne la construction d'un nouveau lexique dans un domaine particulier à partir d'un autre lexique utilisant un corpus de domaine. Les résultats suggèrent que même si les mots d'opinion dépendent du domaine, il peut arriver que le même mot ait une orientation opposée selon le contexte.

1-5- Approche D'apprentissage Automatique

Appelée aussi approche statistique, cette approche se base sur l'apprentissage automatique. Elle utilise la technique de classification pour classer le texte en des classes différentes. Il existe principalement deux types de techniques d'apprentissage.

1-5-1- Apprentissage non supervisé Est basé sur des entrées simples, sans aucune mention de cibles. Cela dépend simplement du regroupement.

1-5-2- Apprentissage supervisé Définit les cibles prédéfinies qui doivent être atteintes, ainsi que les entrées. Les ensembles de données sont formés pour atteindre des résultats significatifs lorsqu'ils sont rencontrés lors de la prise de décision.

1-6- Conclusion

Dans ce chapitre, nous avons vu des définitions de base dans le domaine de l'analyse des sentiments.

Nous avons également discuté des difficultés de ce domaine lors de son application sur la langue arabe et du dialecte. Dans le chapitre suivant, Nous présentons une vue générale sur le deep learning, qui est la technologie que nous adopterons dans notre travail.

Chapitre II

Deep Learning

2.1 Introduction

Avant d'arriver à ce qu'est l'apprentissage profond (DL : Deep Learning), nous devons introduire deux concepts principaux : le premier est le ML que nous introduisons dans le dernier chapitre et le second est le concept d'intelligence artificielle.

L'Intelligence Artificielle (IA) est un vaste domaine, où nous essayons d'imiter le comportement humain dans le but de rendre les machines si puissantes pour accomplir de nombreux types de tâches telles que la résolution de problèmes, la représentation des connaissances, la reconnaissance vocale, et autres. L'idée de base est de mettre les connaissances dans la machine.

Grâce à ces deux domaines., il existe des systèmes sophistiqués capable de changer leur comportement sans qu'il soit nécessaire d'apporter des modifications à leur code, mais uniquement sur leurs données d'entraînement. Ainsi, avec cette vague de techniques avancées d'apprentissage machine qui donne à l'IA un pas en avant, où est la place de DL ou qu'est-ce que le DL apporte dans ce domaine.

De nos jours, DL est au centre de l'attention puisque sa réalisation est beaucoup plus importante que tout autre algorithme d'apprentissage machine dans des tâches aussi complexes, par exemple, nous marquons ce qui suit :

_ Le traitement d'images et la reconnaissance d'objets dans [36] qui nous montrent un progrès en utilisant des réseaux convolutifs profonds pour la reconnaissance d'objets, et l'adoption de l'apprentissage profond par la communauté de computer vision.

_ La reconnaissance vocale et le traitement du signal dans [37] qui nous présentent les résultats obtenus dans la classification phonétique pour la reconnaissance automatique de la parole comme la première application industrielle de l'apprentissage profond.

DL est l'une des raisons qui ont conduit les récents mouvements et progrès de l'IA, et c'est la principale cause qui fait penser que finalement, il existe une possibilité pour l'IA de devenir plus réaliste.

Alors, qu'est-ce que DL?

Selon les fondateurs Yann LeCun, Yoshua Bengio Geoffrey Hinton dans [38] :

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

Autre définition par les auteurs dans [40] : "L'apprentissage profond est une classe de techniques d'apprentissage machine, où l'information est traitée en couches hiérarchiques pour comprendre les représentations et les caractéristiques des données dans des niveaux de complexité croissante."

En d'autres termes, DL est un sous-ensemble des méthodologies et techniques de ML qui utilisent le réseau neuronal artificiel (ANN). C'est l'adaptation des réseaux neuronaux qui imite la structure du cerveau humain. La force de DL réside dans le fait que la machine peut **extraire** des caractéristiques

et apprendre toute seule, indépendamment de l'intervention d'un expert.

Il a été appliqué dans de nombreux domaines différents (traitement des images, textes, paroles et vidéos). Le succès de DL appartient à la disponibilité de plus de données d'entraînement.

Google, Facebook et Amazon a déjà commencé à l'utiliser pour faire l'analyse de leurs énormes quantités de données [91] [92].

La figure suivante représente un résumé des progrès de l'IA depuis ses débuts jusqu'à l'apparition de DL :

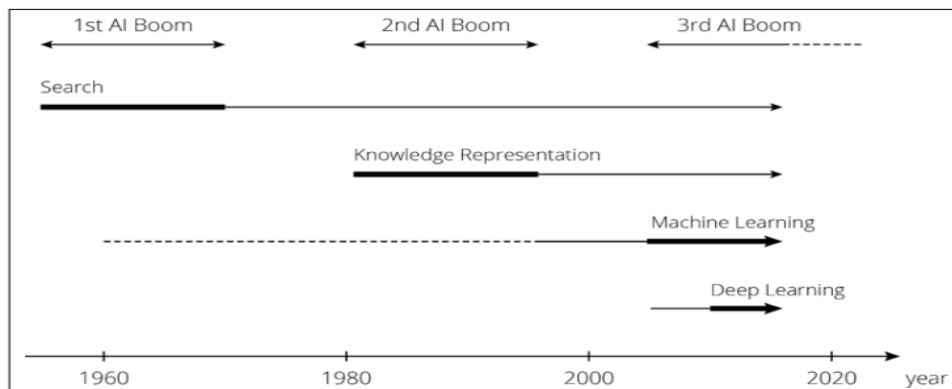


Figure 2.1 – Les progrès de l'Intelligence Artificielle

Dans [68], un article représente un aperçu du contexte historique de DL, il présente les étapes majeures qui mènent à ce que nous avons maintenant. Ces étapes sont résumées dans le tableau suivant :

L'année	Contributeur	Contribution
300 AC	Aristotle	-Introduction de l'associationnisme, début de l'histoire des humains qui essayent de comprendre le cerveau
1873	Alexander Bain	-Introduction du Neural Groupings comme les premiers modèles de réseaux de neurones
1943	McCulloch et Pitts	-Introduction du McCulloch Pitts (MCP) modèle considéré comme L'ancêtre des réseaux de neurones artificielles
1949	Donald Hebb	-Considérer comme le père des réseaux de neurones, il introduit la règle d'apprentissage de Hebb qui servira de fondation pour les réseaux de neurones modernes.

1958	Frank Rosenblatt	-Introduction du premier perceptron
1974	Paul Werbos	-Introduction de la retro propagation
1980	Teuvo Kohonen	-Introduction des cartes auto organisatrices
1980	Kunihiko Fukushima	-Introduction du Neocognitron, qui a inspiré les réseaux de neurones convolutif
1982	John Hopfield -	-Introduction des réseaux de Hopfield
1985	Hilton et Sejnowski	-Introduction des machines de Boltzmann
1986	Paul Smolensky	-Introduction de Harmonium, qui sera connu plus tard comme machines de Boltzmann restreintes
1990	Michael I. Jordan Yann LeCun	-Définition et introduction des réseaux de neurones récurrent -Introduction de LeNet et montra la capacités des réseaux de neurones profond
1997	Schuster et Paliwal	-Introduction des réseaux de neurones récurrent bidirectionnelles
1997	Hochreiter et Schmidhuber	-Introduction de LSTM, qui ont résolu le problème du vanishing gradient dans les réseaux de neurones récurrent
2006	Geoffrey Hinton	-Introduction des Deep Belief Network
2009	Salakhutdinov et Hinton	Introduction des Deep Boltzmann Machines
2012	Geoffrey Hinton	-Introduction de AlexNet qui remporta le challenge ImageNet

Tableau 2.1 – Le résumé de l’histoire de deep learning [68].

La relation entre les trois concepts IA, ML et DL est résumée par les auteurs dans la figure suivante :

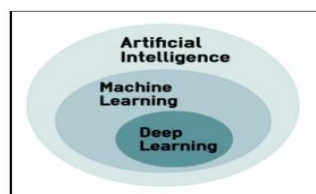


Figure 2.2 – La relation entre les 3 concepts [42]

2.2 Les applications du Deep Learning

2.2.1. La reconnaissance faciale : Les yeux, le nez, la bouche, tout autant de caractéristiques qu'un algorithme de DL va apprendre à détecter sur une photo. Il va s'agir en premier lieu de donner un certain nombre d'images à l'algorithme, puis à force d'entraînement, l'algorithme va être en mesure de détecter un visage sur une image.

2.2.2. Le traitement automatique de langage naturel : Le traitement automatique de langage naturel est une autre application du DL. Son but étant d'extraire le sens des mots, voire des phrases pour faire de l'analyse de sentiments. L'algorithme va par exemple comprendre ce qui est dit dans un avis Google, ou va communiquer avec des personnes via des chatbots. La lecture et l'analyse automatique de textes est aussi Un des champs d'application du DL avec le Topic Modeling : tel texte aborde tel sujet.

2.2.3. Voitures autonomes : Les entreprises qui construisent de tels types de services d'aide à la conduite, ainsi que des voitures autonomes telles que Google, doivent apprendre à un ordinateur à maîtriser certaines parties essentielles de la conduite à l'aide de systèmes de capteurs numériques au lieu de l'esprit humain. Pour ce faire, les entreprises commencent généralement par entraîner des algorithmes utilisant une grande quantité de données. Vous pouvez imaginer comment un enfant apprend grâce à des expériences constantes et à la réplication. Ces nouveaux services pourraient fournir des modèles commerciaux inattendus aux entreprises.

2.2.4. Recherche vocale et assistants à commande vocale : L'un des domaines d'utilisation les plus populaires de DL est la recherche vocale et les assistants intelligents à commande vocale. Avec les grands géants de la technologie ont déjà fait d'importants investissements dans ce domaine, des assistants à commande vocale peuvent être trouvés sur presque tous les smartphones. Le Siri d'Apple est sur le marché depuis octobre 2011. Google aujourd'hui, l'assistant à commande vocale pour Androïde, a été lancé moins d'une année après Siri. Le plus récent des assistants intelligents à commande vocale est Microsoft Cortana.

2.2.5. Ajout automatique de sons à des films muets : Dans cette tâche, le système doit synthétiser des sons pour correspondre à une vidéo silencieuse. Le système est formé à l'aide de 1 000 exemples de vidéos avec le son d'une baguette frappant différentes surfaces et créant différents sons. Un modèle DL associe les images vidéo à une base de données de sons pré-enregistrés afin de sélectionner le son à reproduire qui correspond le mieux à ce qui se passe dans la scène. Le système a ensuite été évalué à l'aide d'un test de contrôle, comme une configuration dans laquelle les humains devaient déterminer quelle vidéo comportait le son réel ou le son factice (synthétisé). Ceci utilise à la fois les réseaux de neurones convolutionnels et les réseaux de neurones récurrents à mémoire à court terme.

2.2.6. traduction automatique : Il s'agit d'une tâche dans laquelle des mots, expressions ou phrases donnés dans une langue sont automatiquement traduits dans une autre langue. La traduction automatique existe depuis longtemps, mais DL permet d'obtenir les meilleurs résultats dans deux domaines spécifiques :

_ Traduction automatique de texte

_ Traduction automatique d'images La traduction de texte peut être effectuée sans aucun traitement préalable de la séquence, ce qui permet à l'algorithme d'apprendre les dépendances entre les mots et leur correspondance avec une nouvelle langue.

2.2.7. Génération automatique de texte : C'est une tâche intéressante, où un corpus de texte est appris et à partir de ce modèle, un nouveau texte est généré, mot par mot ou caractère par caractère. Le modèle est capable d'apprendre comment épeler, ponctuer, former des phrases et même capturer le style du texte dans le corpus. Les grands réseaux de neurones récurrents sont utilisés pour apprendre la relation entre les éléments dans les séquences de chaînes d'entrée, puis pour générer du texte.

2.2.8. Reconnaissance d'image : Un autre domaine populaire en matière de DL est la reconnaissance d'image. Son objectif est de reconnaître et d'identifier les personnes et les objets dans les images, ainsi que de comprendre le contenu et le contexte. La reconnaissance d'image est déjà utilisée dans plusieurs secteurs tels que les jeux, les médias sociaux, la vente au détail, le tourisme, etc. Cette tâche nécessite la classification des objets d'une photo parmi un ensemble d'objets connus auparavant. Une variante plus complexe de cette tâche, appelée détection d'objet, consiste à identifier spécifiquement un ou plusieurs objets dans la scène de la photo et à dessiner un cadre autour d'eux.

2.2.9. la description automatique d'image : Le sous-titrage automatique des images est la tâche pour laquelle le système doit générer une légende décrivant le contenu de l'image. Une fois que vous pouvez détecter des objets sur des photographies et générer des étiquettes pour ces objets, vous pouvez voir que l'étape suivante consiste à transformer ces étiquettes en description de phrase cohérente. Généralement, les systèmes impliquent l'utilisation de très grands réseaux de neurones convolutifs pour la détection d'objets sur les photographies, puis d'un réseau de neurones récurrent comme une mémoire à court terme à long terme pour transformer les étiquettes en une phrase cohérente..

2.2.10. Colorisation automatique : La colorisation de l'image pose le problème de l'ajout de couleurs aux photographies noir et blanc. Le DL peut être utilisé pour utiliser les objets et leur contexte dans la photographie pour colorer l'image, un peu comme un opérateur humain pourrait aborder le problème. Cette capacité tire parti des réseaux de neurones de convolution de grande qualité et de très grande taille formés pour ImageNet et cooptés pour le problème de la colorisation de l'image. Généralement, l'approche implique l'utilisation de très grands réseaux de neurones convolutifs et de couches supervisées qui recréent l'image avec l'ajout de couleurs.

2.2.11. la détection du cancer du cerveau : Une équipe de chercheurs français a noté qu'il était difficile de détecter les cellules cancéreuses du cerveau invasives au cours d'une intervention chirurgicale, en partie à cause des effets de l'éclairage dans les salles d'opération. Ils ont découvert que l'utilisation de réseaux de neurones conjointement avec la spectroscopie Raman pendant les opérations leur permettait de détecter les cellules cancéreuses plus facilement et de réduire le cancer résiduel après l'opération.

2.2.12. Analyse des sentiments du texte : De nombreuses applications ont des commentaires ou des systèmes de révision basés sur des commentaires intégrés à leurs applications. La recherche sur le traitement

du langage naturel et les réseaux de neurones récurrents ont parcouru un long chemin et il est maintenant tout à fait possible de déployer ces modèles sur le texte de votre application pour extraire des informations de niveau supérieur. Cela peut être très utile pour évaluer la polarité sentimentale dans les sections de commentaires ou pour extraire des sujets significatifs à l'aide de modèles de reconnaissance d'entités nommées.

2.2.13. Recherche en marketing : En plus de rechercher de nouvelles fonctionnalités susceptibles d'améliorer votre application, DL peut également être utile en arrière-plan. La segmentation du marché, l'analyse des campagnes marketing et bien d'autres peuvent être améliorés à l'aide de modèles de régression et de classification DL. Cela vous aidera vraiment beaucoup si vous avez une grande quantité de données. Sinon, vous ferez probablement mieux d'utiliser des algorithmes traditionnels d'apprentissage automatique pour ces tâches plutôt que DL . [93]

2.3 Le réseau neuronal

La pratique, de tous les algorithmes de DL sont des réseaux neuronaux [40]. Les réseaux neuronaux, aussi appelés ANN, sont des modèles de traitement de l'information qui simulent le fonctionnement d'un système nerveux biologique. C'est similaire à la façon dont le cerveau manipule l'information au niveau du fonctionnement. Tous les réseaux neuronaux sont constitués de neurones inter connectés qui sont organisés en couches [40].

2.3.1 Le neurone : Ce qui forme les réseaux de neurones, ce sont les neurones artificiels inspirés du vrai neurone qui existe dans notre cerveau. Les 2 figures suivantes montrent une représentation d'un neurone réel et d'un neurone artificiel :

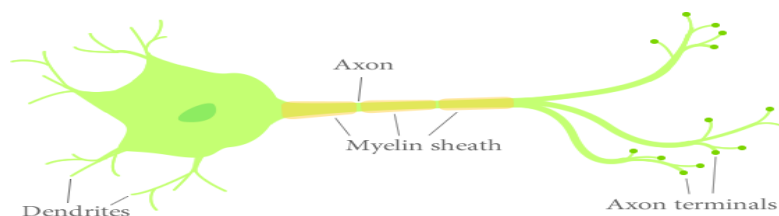


Figure 2.3 – Un neurone réel

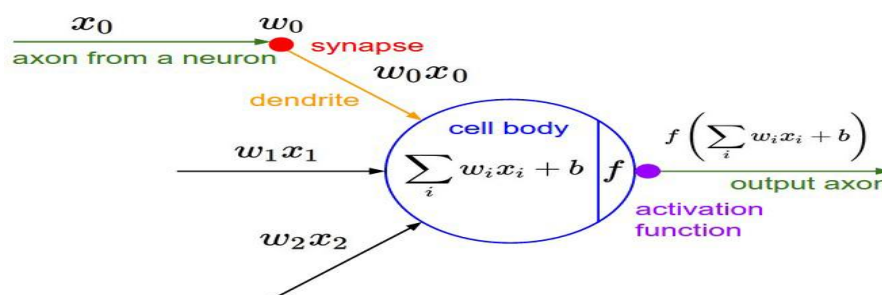


Figure 2.4 – Un neurone artificiel

Les X_i sont des valeurs numériques qui représentent soit les données d'entrée, soit les valeurs sorties d'autres neurones. Les poids W_i sont des valeurs numériques qui représentent soit la valeur de puissance des entrées, soit la valeur de puissance des connexions entre les neurones.

Il existe des opérations qui se passent au niveau du neurone artificiel. Le neurone artificiel fera un produit entre le poids (w) et la valeur d'entrée (x), puis ajoutera un biais (b), le résultat est transmis à une fonction d'activation (f) qui ajoutera une certaine non-linéarité.

2.3.2 Les fonctions d'activation : Après que le neurone a effectué le produit entre ses entrées et ses poids, il applique également une non-linéarité sur ce résultat. Cette fonction non linéaire s'appelle la fonction d'activation.

La fonction d'activation est une composante essentielle du réseau neuronal. Ce que cette fonction a décidé est si le neurone est activé ou non. Il calcule la somme pondérée des entrées et ajoute le biais. C'est une transformation non linéaire de la valeur d'entrée.

Après la transformation, cette sortie est envoyée à la couche suivante. La non-linéarité est si importante dans les réseaux de neurones, sans la fonction d'activation, un réseau de neurones est devenu simplement un modèle linéaire. Il existe de nombreux types de ces fonctions, parmi lesquelles nous trouvons [65] [66] :

2.3.2.1 La fonction Sigmoid : Cette fonction est l'une des plus couramment utilisées. Elle est bornée entre 0 et 1, et elle peut être interprétée stochastiquement comme la probabilité que le neurone s'active, et elle est généralement appelée la fonction logistique ou le sigmoïde logistique.

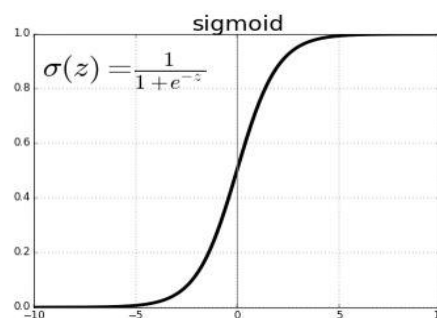


Figure 2.5 – Représentation graphique de la fonction Sigmoïde.

2.3.2.2 La fonction ReLu : La fonction RELU est probablement la plus proche de sa correspondante biologique [66]. Cette fonction est récemment devenue le choix de nombreuses tâches (notamment en computer vision) [65]. Comme dans la formule ci-dessus, cette fonction renvoie 0 si l'entrée z est inférieure à 0 et retourne z lui-même si elle est plus grande que 0.

2.3.2.3 La fonction Softmax : Régression Softmax (synonymes : Logistique multinomiale, Classificateur d'entropie maximale, ou simplement Régression logistique multi-classe) est une généralisation de la régression logistique que nous pouvons utiliser pour la classification multi-classes [67]. Contrairement à d'autres types de fonctions, la sortie d'un neurone d'une couche utilisant la fonction softmax dépend des sorties de tous les autres neurones de sa couche. Cela s'explique par le fait qu'il nécessite que la somme de

toutes les sorties soit égale à 1. Dans [65], nous avons un exemple de reconnaissance de

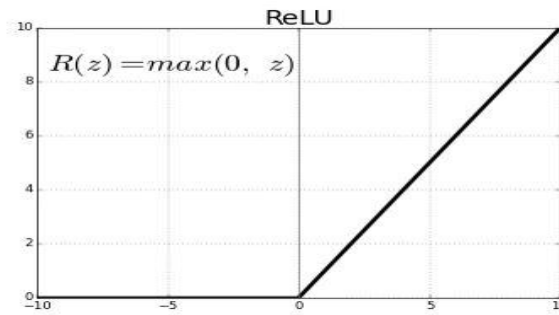


Figure 2.6 – Représentation graphique de la fonction Relu.

chiffres manuscrits. Dans cette tâche, chaque étiquette est comprise entre 0 et 9. Nous devons être confiants dans notre prédiction et cela peut être fait en utilisant une distribution de probabilité. Pour la tâche mentionnée, le vecteur de sortie souhaité est de la forme ci-dessous :

$$[p_0 \ p_1 \ p_1 \ p_2 \ p_3 \ \dots \ p_9]$$

Où :

$$\sum_{i=0}^9 (p_i = 1)$$

2.3.3 Les architectures des réseaux de neurones : La plupart des architectures profondes sont réalisées en combinant et recombinaut un ensemble limité de primitives architecturales (des couches de réseaux neuronaux) [39]. Selon Hinton, dans [42], nous disons que notre réseau de neurones est profond lorsque le nombre des couches cachées est supérieur à 1. Dans cette section, nous présenterons un bref aperçu des structures communes que l'on retrouve dans de nombreux réseaux profonds.

2.3.3.1 Les réseaux entièrement connectés : Un réseau entièrement connecté (Fully Connected en Anglais) permet de transformer une liste d'entrées en une liste de sorties. La transformation est appelée totalement connectée car toute valeur d'entrée peut affecter toute valeur de sortie. Ces couches auront de nombreux paramètres apprenables, même pour des entrées relativement petites [39], mais elles ont le grand avantage de n'assumer aucune structure dans les entrées. Les calculs sont une série de transformations qui changent les similarités entre les cas. Dans ce type de réseaux, les activités des neurones de chaque couche sont une fonction non-linéaire des activités de la couche inférieure [43].

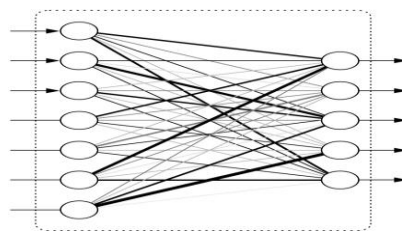


Figure 2.7 – Un réseau entièrement connecté[39].

2.3.3.2 Les réseaux convolutionnels : Un réseau convolutionnel (CNN : Convolutional Neural Networks) suppose une structure spatiale particulière dans son entrée. En particulier, il suppose que les entrées qui sont proches les unes des autres dans l'entrée originale sont sémantiquement liées [39]. CNN est une séquence de couches, et chaque couche transforme un volume d'activations en un autre par une fonction différentiable [70]. Les trois principaux types de couches pour construire ce type de réseau sont :

-La couche convolutive : C'est la couche la plus importante et le coeur des éléments constitutifs du réseau convolutif, et c'est aussi elle qui effectue le plus de calculs lourds.

-La couche de pooling : Il est courant d'insérer périodiquement une couche Pooling dans ce type d'architecture. Sa fonction est de réduire progressivement la taille spatiale de la représentation pour réduire le nombre de paramètres et de calculs dans le réseau, et donc de contrôler également le overfitting.

-La couche entièrement connectée : Comme nous l'avons mentionné précédemment, les neurones d'une couche entièrement connectée ont des connexions complètes à toutes les activations de la couche précédente.

2.4 L'apprentissage en Deep Learning

2.4.1 Introduction

Le processus d'apprentissage dans le DL revient à l'entraînement du réseau neuronal en utilisant des optimiseurs itératifs, qui ne font que conduire la fonction de coût à une très faible valeur [64].

Nous pouvons utiliser différents algorithmes pour effectuer l'apprentissage [40], mais l'algorithme le plus utilisé est l'algorithme itératif d'optimisation par la descente de gradient qui est la méthode la plus utilisée presque sur tous les réseaux neuronaux avec ses différents modèles [64].

Le processus d'apprentissage revient au problème d'optimisation où il s'agit de minimiser ou de maximiser une fonction $f(x)$. Cette fonction est appelée la fonction objective, les auteurs dans [64] l'ont appelée aussi la fonction de coût, la fonction de perte et la fonction d'erreur. Pendant l'entraînement, l'algorithme tente d'identifier le minimum global sans tomber dans le piège du minimum local. Le schéma suivant réalisé par les auteurs en [64] est une illustration de ce concept :

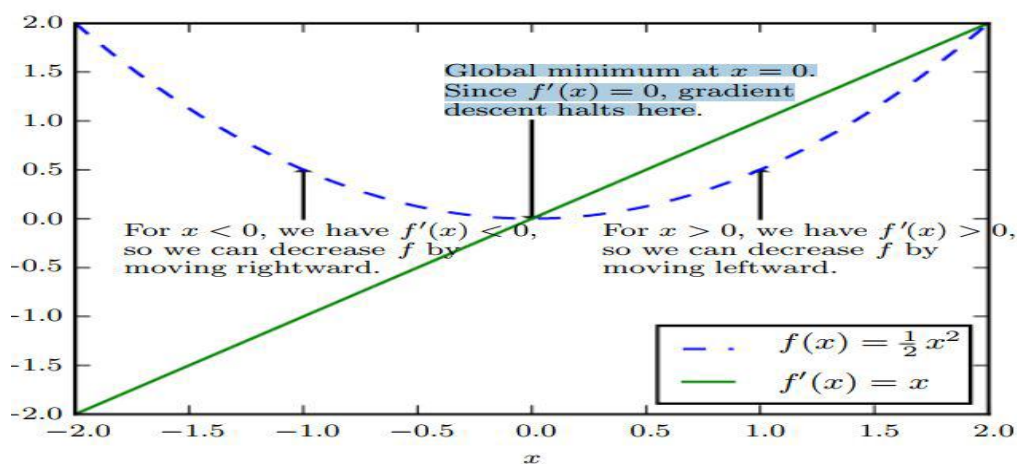


Figure 2-8 – Une illustration du processus de recherche de l'optimum[64].

SVM (Support Vector Machine ou Machine à vecteurs de support) :

Les SVMs ont été développés dans les années 1990. Comme le montre la figure ci-dessous, leur principe est simple : il ont pour but de séparer les données en classes à l'aide d'une frontière aussi « simple » que possible, de telle façon que la distance entre les différents groupes de données et la frontière qui les sépare soit maximale. Cette distance est aussi appelée « marge » et les SVMs sont ainsi qualifiés de « séparateurs à vaste marge », les « vecteurs de support » étant les données les plus proches de la frontière.

2.4.2 Les variantes de la descente de gradient

Il existe trois principaux types de variantes de l'algorithme de descente de gradient. La principale différence entre eux est la quantité de données que nous utilisons lorsque nous calculons le gradient pour chaque étape d'apprentissage [40] [64] [65]. Les algorithmes d'optimisation qui utilisent tout l'ensemble de l'apprentissage sont appelés les méthodes de gradients déterministes ou batch descente de gradient, car ce sont des méthodes où tous les exemples d'apprentissage sont traités simultanément dans un grand batch, tandis que le terme "batch" pour décrire un groupe d'exemples.

Les algorithmes d'optimisation qui n'utilisent qu'un seul exemple à la fois sont parfois appelés méthodes stochastiques ou en online descente de gradient.

La plupart des algorithmes utilisés pour l'apprentissage profond se situent quelque part entre les deux, utilisant plus d'un mais moins que tous les exemples d'entraînement. Ils sont appelés les méthodes stochastiques de minibatch ou de minibatch descente de gradient.

2.4.3 Les Algorithmes d'optimisation de la descente de gradient

Adam : Adam est un algorithme d'optimisation présenté en 2015 [80]. Le nom de cet algorithme est dérivé de Adaptive Moment Estimation. Lors de l'introduction de cet algorithme, les auteurs ont présenté les avantages de l'utilisation d'Adam sur des problèmes d'optimisation non convexes, comme suit :

- _ Simplicité de mise en oeuvre.
- _ Efficacité du calcul.
- _ Peu de mémoire requise.
- _ Bien adapté aux problèmes volumineux en termes de données et/ou de paramètres.
- _ Les hyper-paramètres nécessitent généralement peu de réglages.

Il existe d'autres optimiseurs avec différents mécanismes de fonctionnement, comme :

- _ Adagrad
- _ RMSProp
- _ Adadelta

2-5-Algorithmme d'optimisation par essaim de particules (PSO)

2-5-1- Introduction

Dans ce mémoire nous avons opté d'utiliser l'algorithme d'optimisation par essaim de particules (OEP) pour tenter de résoudre cette problématique.

2-5-2- Notion fondamentale D'après Louis Gacôgne [17], les méthodes de résolution de problèmes peuvent être subdivisées en quatre grandes classes. Tous les problèmes rencontrés ne sont pas forcément en rapport avec l'optimisation mais s'y rapportent pour la plupart, en ce sens qu'il est généralement possible de trouver une fonction à minimiser ou à maximiser. Fréquemment, la difficulté première liée à ce genre de problème est la formalisation (définition de la fonction, puis définition d'un codage de solutions). Il est admis que les quatre procédures sont : les méthodes combinatoires, les méthodes constructives, les méthodes locales, enfin les méthodes évolutionnaires. Etant donné que notre algorithme s'inscrit parmi les méthodes évolutionnaires, c'est sur elles que va être axé notre développement. Avant tout, il est nécessaire de rappeler que les méthodes évolutionnaires sont des méthodes stochastiques et globales faisant intervenir une population de points en s'inspirant de l'évolution des espèces vivantes. Parmi ces méthodes, nous pouvons distinguer celles utilisant des opérateurs explicites (mutation, croisement, etc.) telles que (la programmation évolutionnaire, les algorithmes génétiques, etc.), et celles ayant implicitement des règles de transitions telles que "the Space Partition Algorithm" (SPA), "l'algorithme macro évolutionnaire" (MGA), "l'optimisation par essaim de particules" (PSO), etc.

2-5-3- Présentation de l'algorithme PSO

L'Optimisation par Essaim de Particules (OEP), connu sous le nom anglophone de Particle Swarm Optimization (PSO), est un algorithme inscrit dans la famille des algorithmes évolutionnaires. Il a été proposé par Russel Eberhart (ingénieur en électricité) et James Kennedy (socio psychologue) en 1995 [18]. Cette méthode trouve sa source dans les observations faites lors des simulations informatiques de vols groupés d'oiseaux et de bancs de poissons de Reynold [20], Heppner & Grenander [19]. Autrement-dit, elle s'inspire fortement de l'observation des relations grégaires d'oiseaux migrateurs, qui pour parcourir de « longues distances » (migration, quête de nourriture, parades aériennes, etc.), doivent optimiser leurs déplacements en termes d'énergie dépensée, de temps, (etc.), comme par exemple la formation en forme de V présentée dans la Figure II.1.



Figure 2-9 : Volée d'Anser en formation en V [21]

Le déplacement de ses animaux en essaim est complexe, sa dynamique obéit à des règles et des facteurs bien spécifiques qu'il s'agit de cerner :

- Chaque individu dispose d'une certaine intelligence « limitée » (qui lui permet de prendre une décision).
- Chaque individu doit connaître sa position locale et disposer d'information locale de chaque individu se trouvant dans son voisinage.
- Obéir à ces trois règles simples, « rester proche des autres individus », « aller dans une même direction » ou « voler à la même vitesse ».

Tous ses facteurs et règles sont indispensables pour le maintien de la cohésion dans l'essaim, ceci par l'adoption d'un comportement collectif complexe et adaptatif.

2-5-4- Mode de fonctionnement de l'algorithme PSO

La population dans l'algorithme PSO est nommée essaim, chaque individu du groupe est dit particule. Le déplacement de toute particule (comme indiqué ci-dessus) est régi par des règles et conditions bien spécifiques, influencé par le mouvement des autres particules du voisinage.

Dans un tel contexte ce déplacement à une signification et doit parallèlement répondre à une logique, fondement même du PSO. Il consiste à chercher un optimum dans un voisinage donné, ce déplacement est influé par les trois composantes suivantes :

- Une composante d'inertie : la particule s'efforce de suivre instinctivement son cap de déplacement.
- Une composante cognitive : la particule fait tout pour se diriger vers la meilleure position rencontrée jusqu'à présent.
- Une composante sociale : la particule s'inspire également de l'expérience, du parcours des autres particules, pour se diriger vers la meilleure position rencontrée par ses voisins.

Ce déplacement est illustré dans la Figure II.2.

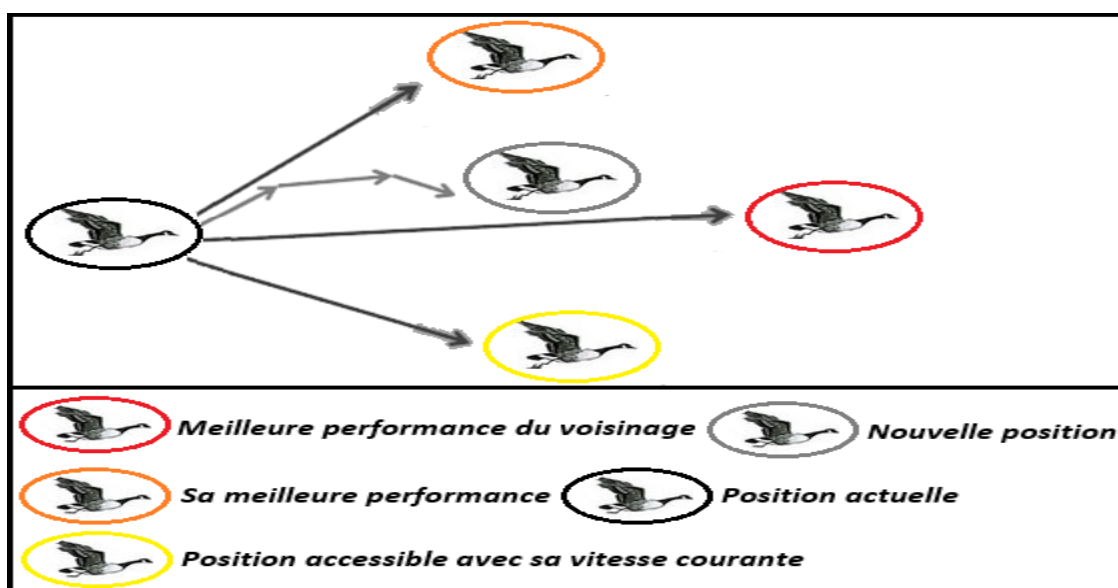


Figure 2-10 : Déplacement d'une particule.

A terme, on se rend compte que toutes les particules, après un certain nombre d'itérations, convergent vers une même position, somme toute la meilleure rencontrée par l'ensemble des particules. Cela ne signifie pas pour autant, dans l'absolu, que c'est la meilleure position du voisinage, juste que c'est la meilleure rencontrée !

a) Composantes de la PSO

Pour être en mesure d'utiliser le PSO, il est indispensable de définir un espace de recherche (composé de particules) et une fonction "objectif" à optimiser. La méthode de l'algorithme consiste alors à déplacer ces particules de telle sorte qu'elles trouvent l'optimum (comme explicité précédemment), elles doivent disposer :

- De données relatives à leurs positions, connaître leurs coordonnées avec comme condition qu'elles soient comprises dans l'espace de définition.
- De la meilleure position qu'elles ont rencontrées.
- De la meilleure position rencontrée par leur voisinage et le résultat de leur fonction «objectif».
- De leur vitesse qui leur permet de se déplacer et de changer de position au fil des itérations.
- D'un voisinage, c'est le sous-ensemble de particules qui interagit directement avec la particule (surtout celle possédant la meilleure position).
- D'après Maurice Clerc et Patrick Siarry [22], l'évolution d'une particule n'est finalement qu'une fusion de trois types de comportements :
 - Egoïste (se déplacer suivant sa vitesse actuelle).
 - Conservateur (revenir en arrière en prenant en compte sa meilleure performance).
 - Panurgien (suivre aveuglement le meilleur de tous en considérant sa performance).

Finalement on remarque un compromis psycho-social entre d'une part la confiance en soi et d'autre part l'influence des relations sociales.

2-5-5- Notion de voisinage

Chaque particule dispose d'un sous-ensemble d'autres particules avec lequel elle est en interaction, c'est le voisinage de la particule. Cet entrelacement de rapports entre toutes les particules est assimilé à la sociométrie ou à la topologie de l'essaim. On dénombre deux types de voisinage :

- **Voisinage géographique** : c'est un voisinage dynamique où les voisins sont les particules les plus proches. A chaque itération, les nouveaux voisins ou groupes doivent être réajustés en se référant à une distance prédéfinie dans l'espace de recherche. C'est donc bien un voisinage dynamique tel qu'illustré sur la Figure II.3.

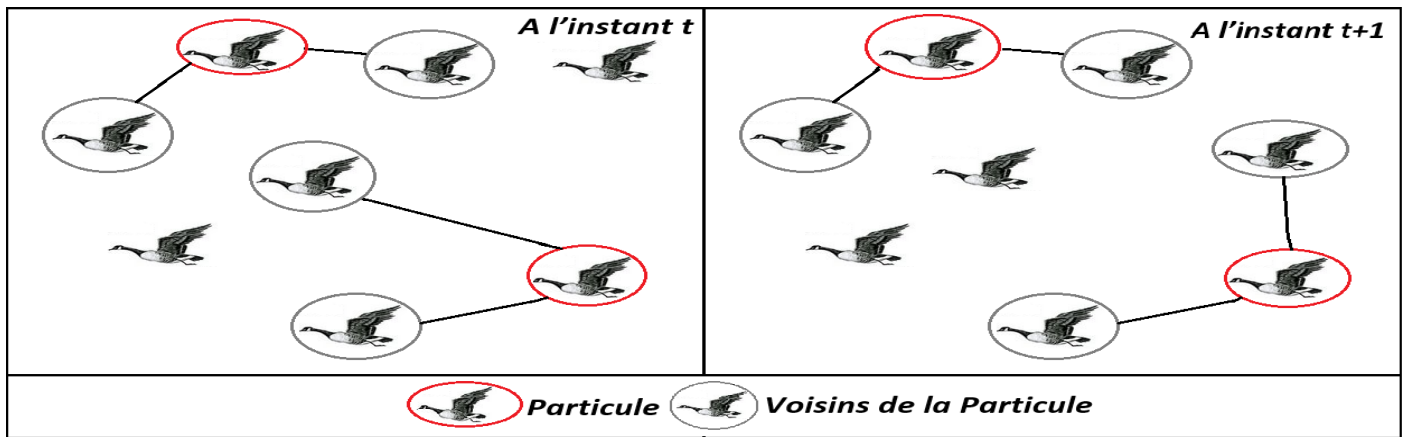


Figure 2-11 : Voisinage géographique à l' instant t et t+1.

Dans cet exemple on va supposer que le voisinage d'une particule est un groupe formé des deux particules les plus proches. Dans la Figure II.3, la notion de voisinage dynamique est mise en évidence puisque pour un même essaim à l'instant « t » et à l'instant « t+1 » le voisinage n'est plus le même.

- **Voisinages sociaux** : ce type de voisinage est considéré comme statique, les voisins restent figés, autrement-dit, ils demeurent inchangés. C'est le voisinage auquel on a le plus souvent recours, en raison :
 - De sa simplicité de programmation.
 - Parce qu'il offre un meilleur rapport temps/coût, en termes de calcul.
 - Dans un scénario de convergence, un voisinage social s'oriente forcément vers un voisinage géographique.

Dans la figure qui va suivre, les particules sont d'abord semées de manière fictive en forme de cercle, ensuite pour la particule étudiée, on insère au fur et à mesure dans ses informatrices, dans un 1^{er} temps elle-même, dans un 2^{ème} temps celles qui lui sont adjacentes, puis de proche en proche jusqu'à atteindre la taille souhaitée, ainsi que définie dans la Figure II.4.

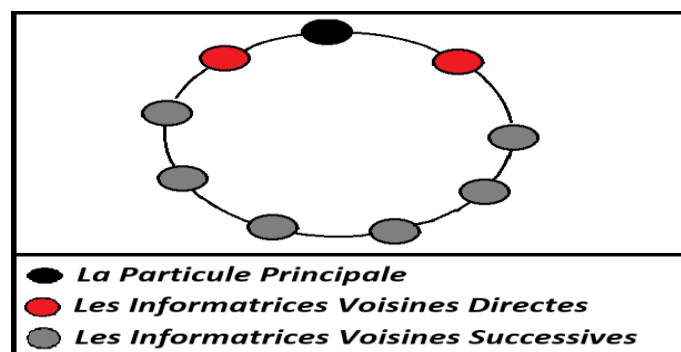


Figure 2-12 : Voisinage en cercle (choix régulier des informatrices).

Dans cet exemple, la particule principale est placée en haut (en noir), ses informatrices correspondent aux deux particules directement à sa droite et à sa gauche (en rouge dans notre figure).

Dans l'ébauche ci-dessous par contre, les informatrices sont choisies de manière aléatoire, telle qu'exposée dans la Figure II.5.

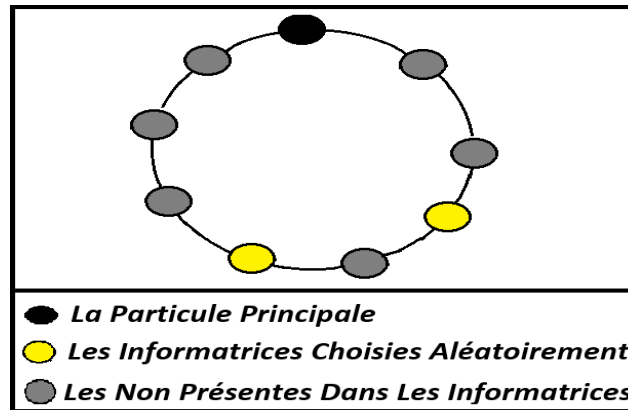


Figure 2-13 : Voisinage en cercle (choix aléatoire des informatrices).

Ici, la particule principale est placée en haut (en noir) et ses informatrices sont choisies de façon aléatoire (en jaune par exemple dans notre figure).

2-5-6- Fonctionnement de l'algorithme PSO

En réalité, il existe deux types d'algorithmes de PSO : le PSO discret, et le PSO continu, nous allons nous intéresser uniquement au PSO continu tel qu'il est défini dans les travaux de Michel Gourgand et Sylvérin Kemmoé Tchomté [23].

Dans l'espace de recherche de dimension \mathbf{D} , une particule i de l'essaim est représentée par son vecteur position et par son vecteur vitesse ; formulés ainsi :

$$\vec{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$$

$$\vec{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$$

L'appréciation de la qualité de sa position est arrêtée par la valeur de la fonction "objectif" en ce point. Il est indispensable que cette particule puisse mémoriser la meilleure position par laquelle elle est déjà passée, formulée comme suit :

$$\vec{pbest}_i = (pbest_{i1}, pbest_{i2}, \dots, pbest_{iD})$$

La meilleure position atteinte par les particules de l'essaim est formulée comme suit :

La notion de **Gbest** (global best), est calquée sur la version PSO Globale (PSOG) où toutes les particules de l'essaim sont issues de la particule i .

$$\vec{Gbest} = (gbest_1, gbest_2, \dots, gbest_D)$$

Au commencement de l'algorithme, les particules de l'essaim sont initialisées de manière aléatoire/régulière dans l'espace de recherche. Par la suite, à chaque itération, les particules se déplacent, en fusionnant les trois composantes citées ci-dessus.

$wv_{i,j}^t$: représente la composante d'inertie du déplacement, où le paramètre w gère l'influence de la direction de déplacement sur le déplacement futur ;

$c_1 r_{1,i,j}^t [pbest_{i,j}^t - x_{i,j}^t]$: représente la composante cognitive du déplacement, où le paramètre C_1 gère le comportement cognitif de la particule ;

$c_2 r_{2,i,j}^t [gbest_j^t - x_{i,j}^t]$: représente la composante sociale du déplacement où le paramètre C_2 gère l'aptitude sociale de la particule.

La fusion de ces trois composantes nous permet d'abord de dégager la vitesse qui elle-même nous offre la possibilité de calculer la position de la particule.

Ces deux variables sont obtenues par l'utilisation de l'équation (1.1) et de l'équation (1.2), successivement comme suit :

$$v_{i,j}^{t+1} = wv_{i,j}^t + c_1 r_{1,i,j}^t [pbest_{i,j}^t - x_{i,j}^t] + c_2 r_{2,i,j}^t [gbest_j^t - x_{i,j}^t], j \in \{1, 2, \dots, D\} \quad (1.1)$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1}, j \in \{1, 2, \dots, D\} \quad (1.2)$$

- w est une constante, appelée coefficient/pourcentage d'inertie.

- c_1, c_2 sont deux constantes, appelées coefficients/pourcentage d'accélération.

- r_1, r_2 sont deux nombres aléatoires tirés uniformément dans $[0, 1]$, et ce à chaque itération t et pour chaque dimension j .

A la fin du déplacement des particules dans une itération donnée, les nouvelles positions sont évaluées et les deux vecteurs $Pbest_i$ et $Gbest$ sont réindexés, conformément aux deux équations (1.3) dans le cas d'une minimisation d'une fonction objective puis (1.4) dans le PSO.

$$\bar{P}best_i(t+1) = \begin{cases} \bar{P}best_i(t), & \text{si } f(\bar{x}_i(t+1)) \geq \bar{P}best_i(t) \\ \bar{x}_i(t+1), & \text{sinon} \end{cases} \quad (1.3)$$

$$\bar{G}best(t+1) = \arg \min_{\bar{P}best_i} f(\bar{P}best_i(t+1)), \quad 1 \leq i \leq N. \quad (1.4)$$

Cette procédure est représentée dans l'Algorithme, où N représente le nombre de particules de l'essaim, voir la Figure II.6.

OEP	
1	Initialiser aléatoirement N particules : position et vitesse.
2	Evaluer les positions des particules
3	Pour chaque particule i , $\bar{P}best_i = \bar{x}_i$
4	Calculer $\bar{G}best$ selon (1.4)
5	tant que le critère d'arrêt n'est pas satisfait faire
6	Déplacer les particules selon (1.1) et (1.2)
7	Evaluer les positions des particules
8	Mettre à jour $\bar{P}best_i$ et $\bar{G}best$ selon (1.3) et (1.4)
9	fin

Figure 2-14 : Algorithme d'optimisation par essaim particulaire .

Ledit algorithme, va servir comme référence pour l'implémentation de notre application.

Nous schématisons l'algorithme PSO à l'aide d'un organigramme se trouvant dans la Figure II.7.

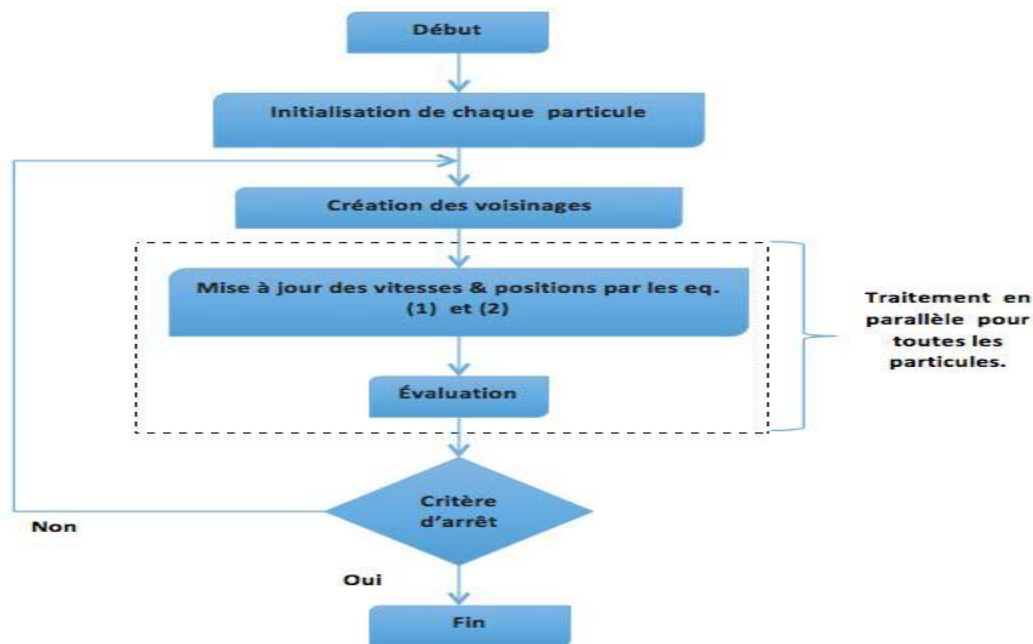


Figure 2-15 : Organigramme « méthode des essaims particulaires » [38]

2-5-7- Améliorations apportées au PSO

a) Cantonnement de particule

Dans l'hypothèse d'un déplacement très rapide de la particule, cela peut l'éjecter de l'espace de recherche, il serait judicieux d'introduire une nouvelle variable V_{max} . Elle aura pour objet de limiter la vitesse sur chaque dimension et ainsi de parer à une éventuelle explosion du système [24].

Cela n'amoindrit nullement les valeurs de x_i à l'intervalle $[V_{min}, V_{max}]$, il borne seulement la distance maximale parcourue par une particule au cours d'une itération. Un tel procédé à l'avantage aussi de contrôler la divergence de l'algorithme et de réaliser un compromis efficace entre intensification et diversification.

Pareille amélioration peut être aussi enrichie par une stratégie de confinement/cantonement des particules, afin d'éviter tout problème de leur sortie.

Parmi les stratégies proposées on peut trouver :

- Une 1^{ère} stratégie qui consiste à réintégrer une particule sortie de son espace de recherche.
- Une 2^{ème} stratégie qui consiste à ce que la particule soit laissée à l'extérieur de l'espace de recherche, sans prendre en compte sa fonction "objectif". Par conséquent elle ne risque pas d'attirer les autres particules (risque d'attraction).
- Dans une 3^{ème} stratégie, la particule est stoppée à la frontière et les composantes associées à sa vitesse sont annulées.

- Enfin dans une 4^{ème} stratégie, la particule rebondit sur la frontière et se trouve bloquée ; les composantes de la vitesse sont multipliées par un coefficient tiré aléatoirement dans l'intervalle [-1,0].

b) Taux de constriction

Comparativement à l'algorithme de base, plusieurs améliorations ont été apportées, surtout sur le contrôle de la divergence, plus précisément grâce à l'introduction du paramètre **Vmax** (comme vu auparavant). Il permet de limiter la divergence des particules d'une part, d'autre part de nombreuses études similaires sur la dynamique des particules tendent à analyser sous-quelles conditions une convergence de l'essaim se produit [25, 26, 27].

Pour obtenir un bon équilibre entre les phases de diversification et d'intensification du processus de recherche une bonne combinaison des paramètres **w**, **C1** et **C2** demeure déterminante [28] [29].

Selon Clerc et Kennedy pour avoir une bonne convergence, il est nécessaire de rendre interdépendants les paramètres **w**, **C1**, **C2**.

L'utilisation d'un taux/coefficient de constriction χ (ou facteur de constriction) permet de surcroit de mieux gérer la divergence de l'essaim et de se libérer de la définition de **Vmax** [29]. Cette variante de PSO est connue sous le nom de canonical PSO.

En utilisant le coefficient de constriction, l'équation (1.1) devient alors :

$$v_{ij}(t+1) = \chi (v_{ij}(t) + \phi_1 r_1 (pbest_{ij}(t) - x_{ij}(t)) + \phi_2 r_2 (gbest_j(t) - x_{ij}(t))) \quad (1.5)$$

Avec :

$$\chi = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} \quad (1.6)$$

où : $\phi = \phi_1 + \phi_2$, $\phi > 4$.

Les valeurs optimales de ϕ_1 , ϕ_2 sont déterminées dans [29], ceci après de nombreux tests. En général, on utilise $\phi = 4,1$ et $\phi_1 = \phi_2$, ce qui donne un coefficient $\chi = 0,7298844$. Dans [30], les auteurs qui se sont penchés sur la question ont indiqué que

l'utilisation d'un coefficient de constriction offre un meilleur taux de convergence, sans avoir à fixer de vitesse Métaheuristique d'optimisation appelée état de l'art maximal **Vmax**.

Dans certains cas, le coefficient de constriction à lui seul ne permet pas la convergence vers la solution optimale pour un nombre d'itérations donné.

Pour remédier à cet handicap, il serait intéressant de fixer : **Vmax = (xmax - xmin)/2** en plus du coefficient de constriction, ce qui, selon les études de Shi et Eberhart, permet d'améliorer les performances globales de l'algorithme. Il est nécessaire de noter que PSO, utilisant un coefficient de constriction, n'est pas la seule version (PSO) qui garantisse la convergence vers un état d'équilibre. D'autres exemples sont disponibles dans [27, 31, 32, 33], elles permettent aussi de provoquer la convergence de l'algorithme.

c) Topologie du voisinage

Comme repris plus haut le choix de la topologie (réseau de communication entre particules) a une influence notable sur la performance du PSO, plus précisément sur le déplacement des particules. Dans la version PSO (voir Figure II.6), les auteurs ont défini une topologie entièrement connectée (chaque particule est reliée aux autres). D'où l'appellation PSO version globale (**Gbest**), chaque particule dispose d'informations sur toutes les particules de l'essaim. Ces données effectivement utilisées sont représentées par le terme **Gbest**.

Cependant cette version a l'inconvénient de ne pas pousser l'exploration suffisamment loin ; cela débouche sur une stagnation dans un optimum local et donc à une convergence prématurée.

Pour dénouer ce noeud, plusieurs variantes de PSO "originales", dénommées version locale (**Lbest**) ont été mises en place. On peut citer celle proposée dans [24], laquelle recourt à un graphe d'information statique sous forme d'anneau (cette version est connue comme étant la version locale classique). Dans PSOL, le terme **Gbest** est remplacé par **Lbest**. A chaque particule i est défini un voisinage **i.e.**, l'information partagée doit être la meilleure solution trouvée dans le voisinage de chaque particule **Pbest_i**.

d) Taux d'inertie

Introduit par Shi et Eberhart [34], le coefficient d'inertie w contrôle les effets de la direction de la particule sur le déplacement futur. L'ajout de ce paramètre vise un équilibre entre la recherche locale (exploitation) et la recherche globale (exploration).

L'intensité de l'exploration de l'espace de recherche dépend de la valeur du poids d'inertie, si une grande valeur de w facilite une exploration globale, une petite valeur de w facilite quant à elle l'exploration locale. Du fait de son influence sur les performances de l'algorithme PSO, le poids d'inertie a suscité un grand intérêt de la part de la communauté de chercheurs. Y. Shi and E.R.C, dans [34], proposent pour leur part un taux d'inertie dynamique, variable dans le temps. Il débute par une valeur proche de 0,9 et décroît linéairement pour arriver à 0,4. Cette stratégie a beaucoup amélioré les performances de PSO en résolvant bon nombre de problèmes d'optimisation. Le coefficient d'inertie w varie linéairement dans le temps, selon la formule suivante :

$$w = w_{min} + (w_{max} - w_{min}) \cdot \left(\frac{iter}{max_{iter}} \right) \quad (1.7)$$

Où $iter$ est l'itération courante et max_{iter} est le nombre maximal d'itérations. w_{max} et w_{min} désignent respectivement les valeurs maximums et minimums du coefficient w (généralement, $w_{min}, w_{max} \in [0, 1]$).

Dans [35], Chatterjee et Siarry ont utilisé une autre stratégie non-linéaire pour définir un coefficient d'inertie dynamique. Et, Eberhart et Shi [36] ont proposé pour leur part une autre variante, dans laquelle le coefficient d'inertie est choisi au hasard, selon une distribution uniforme, dans l'intervalle **[0,5, 1]**. Cet intervalle est inspiré du facteur de constriction proposé par Clerc et Kennedy (la valeur attendue du coefficient d'inertie, dans ce cas, est égale à $0,75 \approx 0,729$) [37].

2-6- L'analyse du sentiment et le Deep Learning

Le DL a prouvé son efficacité dans de nombreux problèmes complexes avec l'utilisation de réseaux de neurones artificiels pour apprendre et extraire des modèles et des informations significatives depuis les données. Par conséquent, nous trouvons de nombreuses contributions qui tentent d'adapter cette approche comme une solution au problème de l'analyse du sentiment. Récemment, de nombreux articles ont été publiés dans lesquels nous trouvons l'utilisation des différentes architectures de réseaux de neurones mentionnées précédemment :

Dans [77] qui explorait l'application des CNN profonds sur les tweets. Les résultats ont été très bons selon les auteurs et les résultats obtenues ont été classées dans les deux premières en terme de robustesse dans le Semeval-2015 [74].

Dans [78] ils ont proposé un réseau CNN dynamique (appelé DCNN) pour la modélisation sémantique des phrases. Le modèle a réussi à capturer les relations entre les mots.

Les expériences ont été faites sur les tweets data set qui ont une robustesse de 87,4 % et sur les revues de films data set qui ont une robustesse de 86,8 %.

Dans [79], ils ont introduit un modèle de classification des sentiments au niveau du document.

Les auteurs construisent un modèle qui utilise d'abord un réseau de neurones convolutif pour apprendre la représentation des phrases, puis la sémantique des phrases.

Les relations entre les phrases sont codées d'une manière adaptative dans la représentation de documents avec un réseau de neurones récurrent. Les résultats obtenus ont été les meilleurs, comme l'affirment les auteurs utilisant quatre data sets de l'IMDB [76] et de Yelp [75]. Le tableau ci-dessous résume ces résultats :

	Yelp 2013	Yelp 2014	Yelp 2015	IMDB
LSTM-GRNN	65.1 %	67.1 %	67.6 %	45.3 %

Table 2.2 – Résumé des résultats obtenus avec la robustesse comme une métrique d'évaluation.

Dans [87] un travail qui présentait une comparaison empirique entre le SVM et les réseaux neuronaux artificiels (ANN) concernant l'analyse du sentiment au niveau du document.

Pour tester leurs travaux, les auteurs ont expérimenté les modèles sur quatre data sets différents. Le ANN a dépassé SVM d'une manière significative en terme robustesse, comme le montre le tableau ci-dessous:

	ANN	SVM
Movies dataset	86%	85.2%
GPS dataset	87.3%	84.5%
Books dataset	81.8%	80.9%
Cameras dataset	90.3%	89.6%

Tableau 2.3 – Résumé des résultats basant sur la robustesse comme une métrique d'évaluation.

2-7-Conclusion

Dans ce chapitre, nous avons présenté les définitions du Deep Learning et ces applications ,nous avons donné un aperçu plus ou moins détaillé sur un modèle de classification très puissant qui est le réseaux de neurones, Le prochain chapitre, traite les détails de la conception, ainsi que la méthode et les outils utilisés pour la réalisation de notre application.

Chapitre III

conception et implémentation

3. Introduction

Dans une expérience d'analyse de texte, nous devons généralement ; nettoyer et prétraiter un jeu de données de texte ; extraire des vecteurs de fonctions numériques du texte prétraité ; former le modèle d'apprentissage et noter et valider le modèle.

Dans ce chapitre on va expliquer l'architecture détaillée de notre système d'explication, quelques processus, la méthode utilisée pour la transformation de données textuelles, enfin l'algorithme choisi pour l'implémentation, qu'ils consomment beaucoup de temps et nécessitent un grand effort.

3-1-conception

3-1-1-Problématique

L'analyse des big social data est une tâche complexe et est confrontée à plusieurs défis. En effet, il existe des points de vue contradictoires sur la fiabilité de l'analyse des médias sociaux ; alors que certaines études ont démontré une corrélation directe entre les événements et leurs publications connexes, d'autres, comme Gayo-Avello (2012), Tumasjan (2010) et DeLenn Chin, Jessica Zhao and Anna Zappone (2016), ont montré l'inverse. Bien que ces dernières aient été basées sur des méthodes statistiques et / ou sur des dictionnaires référencés, elles ont été vouées à l'échec. En général, ce problème est lié à des défis d'exploration de l'opinion et du sentiment. pour prédire la classe de sentiment des nouveaux documents. Cependant, cette approche devient problématique lorsqu'il est difficile d'obtenir des données d'entraînement fiables et suffisantes comme dans le cas des big social data. De même, les données d'apprentissage ont tendance à être très spécifiques à un domaine ou contexte précis, performantes dans le contexte de l'étude mais inadaptées à un domaine différent. Les big social data sont diversifiés et concernent plusieurs domaines tels que la politique, le marketing, ou la santé. Les systèmes d'analyse des textes issus des réseaux sociaux doivent donc maintenir une performance dans des domaines différents.

3-1-2- Contribution

Nos contributions principales consistent à :

Travailler sur le dialecte Algérien, avec l'exploitation de classificateurs, Support Vecteur Machines avec optimisation d'essaim de particules (PSO)

- Annotation un Dataset qui contient 2891 commentaires.
- Faire un dictionnaire de 1328 mots annotés en dialecte Algérien.

3-1-3- Source de données

Dans le cadre de notre engagement envers cette communauté, nous nous attachons à fournir un accès aux données qui soit ouvert et équitable pour les développeurs, sécurisé pour les utilisateurs de Twitter, et bénéfique pour la plate-forme Twitter dans son ensemble. Pour atteindre ces objectifs, nous avons souhaité que la présente Politique Développeurs constitue un guide permettant à chacun de comprendre nos règles et nos attentes concernant l'utilisation appropriée de l'API et des contenus [Twitter](#).

3-1-3-1- Caractéristiques des tweets

Nous présentons rapidement dans cette sous-partie les principales caractéristiques d'un tweet et de Twitter.

3-1-3-1-1-Longueur : La longueur maximale d'un message posté sur Twitter est de 140 caractères.

D'après Go et al.(Go, Alec and Bhayani, Richa and Huang, Lei, 2009), la longueur moyenne des tweets est de 14 mots ou 78 caractères sur ce corpus. Cette longueur est très courte contrairement à celles utilisées dans d'autres corpus pour la classification de sentiments (comme les critiques de films).

3-1-3-1-2-Disponibilité des données et modèle du langage : Les sujets abordés sur Twitter sont très divers et l'API twitter permet de récolter des millions de messages. En effet, le nombre de tweets postés chaque jour est immense. Les utilisateurs peuvent poster des messages depuis n'importe quel lieu et avec différents appareils. Il est à noter qu'un tweet peut contenir des fautes d'orthographe liées à l'utilisation l'utilisation de smartphones et à la limitation de caractères. De plus, le registre de langue utilisé peut être familier.

Le tableau 3.1 présente le nombre des commentaires et leurs sources.

Source	Normale	Offensante	Obscène	Total de Commentaires
Twitter	2892	1497	611	5000

Tableau 3.1 : Données collectées.

Comme toute opération d'assemblage de données, cette étape requiert un prétraitement, qui consiste à filtrer les téléchargements afin de garder seulement les textes arabes et éliminer les mots vides pour obtenir un corpus propre et prêt à utiliser.

3-1-4- Annotation

L'annotation ou l'étiquetage des opinions est une tâche humaine qui est un peu difficile car elle prend beaucoup de temps pour suivre les commentaires un par un, et elle a besoin des fois une grande discussion pour atteindre une décision finale, est ce que cette opinion est positive ou négative !

Alors, entre trois et quatre semaines et avec deux annotateurs (un troisième pour que le conflit entre les annotateurs va être résolu par un vote à la majorité) nous avons étiqueté toutes les 2891 entrées par l'utilisation de trois polarités positive, négative et neutre (qui ont les valeurs respectivement 1, -1, 0). Le tableau 3.2 divise le Dataset selon les trois valeurs de polarités.

Polarité	Positive	négative	neutre	Total
de commentaires	975	525	1391	2891

Tableau 3.2 : Nombre de commentaires par polarité.

Dans le tableau 3.3, nous avons décrit quelques exemples avec leurs polarités.

Commentaire	Polarité
شفت المواطن مش واعي	négative
في خنشة خويا لعزیز طریق باتنة ؟	neutre
مايسجلش	négative
ربي يكثر من امثالك	positive
مرحب بيكم	positive

Tableau 3.3 : Exemple de notation de quelques commentaires.

3-1-5- Création de dictionnaire

Est un ensemble des mots qu'un classificateur peut les utiliser pour évaluer la polarité d'un texte. Dans la littérature, nous n'avons pas trouvé un lexique spécifique des mots en dialecte algérien. Donc, nous aurons obligé d'établir notre propre dictionnaire. Quatre semaines de travail pris pour ramasser 1328 mots à partir de sites web. Aussi, nous avons demandé l'aide des amis du nord de l'Algérie, du sud, de l'est, de l'ouest afin de couvrir le maximum possible des régions de notre pays, ces mots sont étiquetés comme l'illustre le tableau 3.4.

Polarité	Positive	Négative	Total
Nombre de mots	565	763	1328

Tableau 3.4 : Statistiques de notre dictionnaire

3-1-6- Architecture générale

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

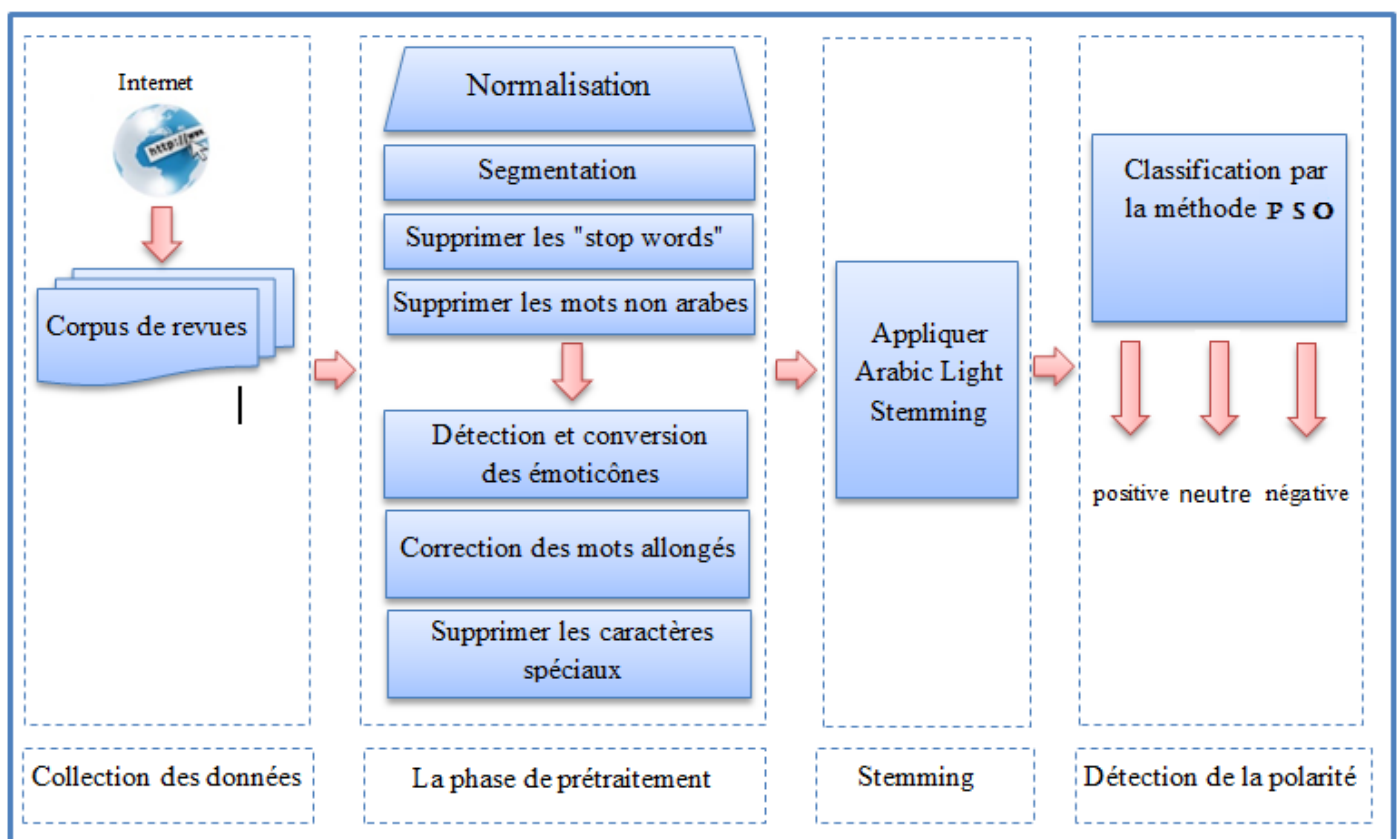


Figure 3-1 : Représentation de l'architecture générale du système.

3-1-7- Architecture détaillée

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

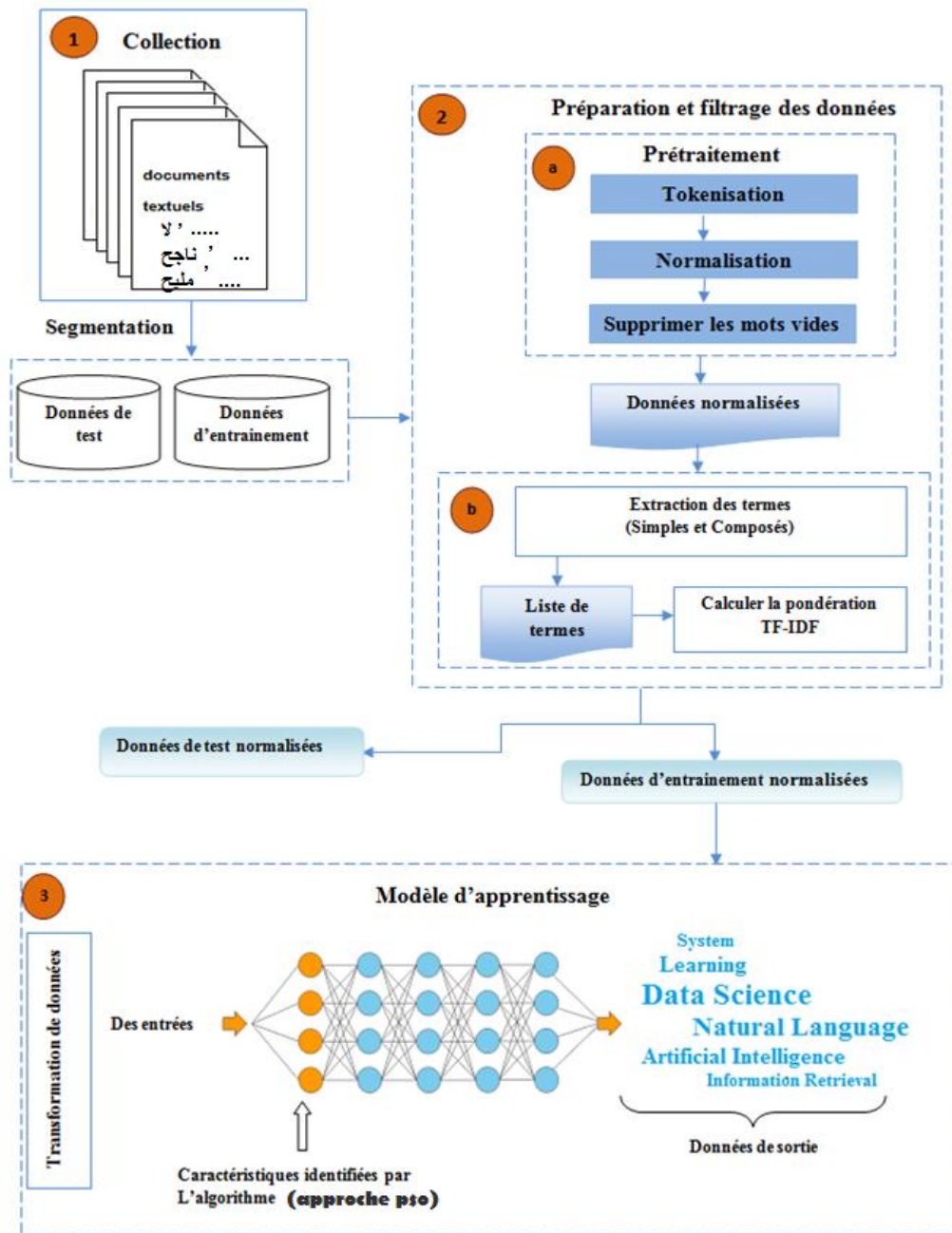


Figure 3-2 : Représentation de l'architecture détaillée de système.

3-1-8- Méthodologie de recherche

Les grandes phases de l'approche PSO proposée sont : le prétraitement du texte, le calcul informatif et sémantique scores, la construction d'un graphique des solutions candidates, puis l'utilisation de l'algorithme PSO pour équilibrer les deux scores informatifs .

3-1-8-1- Prétraitement de texte : Cette étape comprend cinq étapes : segmentation, tokenisation, élimination des mots vides, radicalisation et construction documenter la liste des racines des mots. Dans la

segmentation, le texte original est divisé en phrases. Les signes de ponctuation, tels que des virgules, des points-virgules, des points d'interrogation, des points d'exclamation, des deux-points et des points sont utilisés pour diviser le texte en Phrases. Chaque phrase reçoit un numéro séquentiel (ID). Dans la tokenisation, les phrases sont divisées en mots sur la base d'espaces blancs et de ponctuation. La liste des mots vides est une liste contenant les mots qui sont exclus avant le traitement automatique de la langue des textes pour réduire le contenu du texte à des mots plus utiles pour les résumés. Dans cette étude, nous avons utilisé El-Khair [19] liste de mots vides en arabe. Nous avons également utilisé les instituts de recherche en sciences de l'information (ISRI) Arabic Stemmer [20] aux termes connexes dans le document. Pour créer une liste de tiges de document pour un document donné, nous avons extrait la racine des mots distincts pour créer une liste ordonnée du vocabulaire distinct du document. Chaque phrase du document est représentée comme un vecteur, où chaque vecteur a la même longueur (la taille de la liste) dans la notation sémantique marcher. Plus précisément, nous utilisons la liste des tiges de document pour créer le vecteur IDF (fréquence inverse du document) et pour créer deux vecteurs supplémentaires, le vecteur Terme Fréquence (TF) et Terme Fréquence-Inverse Document Fréquence (TF-IDF) vecteur pour chaque phrase du document.

3-1-8-2- Calcul des scores informatifs

Les caractéristiques structurelles d'une phrase sont utilisées pour calculer le score informatif pour chaque phrase de la source document afin de classer les phrases. Chaque phrase reçoit un score, qui est la somme de tous les scores heuristiques de cette phrase.

Les heuristiques statistiques de phrases suivantes sont utilisées pour attribuer un score à chaque phrase [9][14] :

- A. Caractéristique de similarité de titre (T) : si le mot de la phrase existe dans le titre du texte, cela indique que la phrase est très pertinente pour le document. Le score de similarité de titre (T) est calculé par la formule suivante :

$$T = \frac{\text{no of titlewords in s}}{\text{no of words in title}}$$

- B. Caractéristique de longueur de phrase (L) : la longueur est le nombre de mots dans une phrase, les phrases longues sont plus susceptibles de contiennent des informations importantes, mais les phrases très courtes ne sont généralement pas incluses dans le résumé car elles contiennent moins d'informations. Pour calculer le score de longueur de phrase, la formule suivante est utilisée :

$$T = \frac{\text{no of words in s}}{\text{no of word in longest sentence}}$$

- C. Fonction de localisation de la phrase (Loc) : la localisation indique l'importance des phrases, des informations importantes dans un document est souvent couvert par les rédacteurs au début de l'article. Ainsi, les phrases de début sont supposé contenir le contenu le plus important. Le score

d'emplacement de la phrase est relatif au nombre total de déclarations et calculé par la formule :

$$\log = 1 - \frac{\log s}{N}$$

Où N est le nombre total de phrases et Loc S est l'emplacement de la phrase dans le document.

Fonction de pondération du terme TF-IDF : la fréquence du terme (TF) est la fréquence d'un mot dans une phrase. Inverser le document

fréquence (IDF) désigne le nombre de fois que le mot apparaît dans toutes les phrases du document à résumé. La fonction Term TF-IDF établit un équilibre entre les occurrences de terme locales et globales dans les phrases [21]. Le TF-IDF du terme t dans une phrase s peut être défini comme :

$$\text{TF-IDF (tsi)} = \text{TF(tsi)} * \text{IDF(t)}$$

(tsi) : est le nombre de fois que le terme t apparaît dans la phrase i dans un seul document. IDF (t) : le document inverse La fréquence du terme t peut être définie comme :

$$\text{IDF(t)} = \log = \frac{\log N}{N_t}$$

N : est le nombre de toutes les phrases du document. : est le nombre de phrases qui contiennent le terme t.

Le TF-IDF de chaque phrase est la somme de tous les TF-IDF de ses termes constitutifs.

$$\text{TF-IDF (si)} = \sum_{t=1}^n \text{TF-IDF (tsi)}$$

Où n est le nombre de termes dans la phrase i. Chaque phrase reçoit un score informatif, qui est la somme de tous précédents caractéristiques scores de cette phrase. L'équation utilisée pour calculer le score informatif est la suivante.

$$\text{ISCORE (si)} = T+L+LOC+ \text{TF-IDF (si)}$$

(si) est le score informatif pour la phrase (i), (T) est la caractéristique de similarité du titre, (L) est la longueur de la phrase

3-1-8-3- Calcul des scores sémantiques

Le score sémantique est une valeur numérique qui reflète le degré de proximité sémantique entre deux phrases.

la similarité cosinus est utilisée pour mesurer la corrélation entre les phrases correspondantes (vecteurs).

Pour calculer le score sémantique et construire une matrice de similarité entre les phrases, chaque

déclaration dans le document est représentée comme TF vecteur, le vecteur IDF est créé, puis chaque

déclaration dans le document est représentée comme vecteur TF-IDF. Vecteurs TF-IDF et la similitude

entre les phrases sont calculées à l'aide des formules (8), (9), (10) et [22][23].

$$\text{TFsi}=(\text{TF}(t1si), \text{TF}(t2si), \text{TF}(t3si), \dots, \text{TF}(tnsi)) \tag{8}$$

$$\text{IDFsi}=(\text{IDF}(t1si), \text{IDF}(t2si), \text{IDF}(t3si), \dots, \text{IDF}(tnsi)) \tag{9}$$

$$\text{SIM-SCORE}(si,sj)=\frac{\sum_{t=1}^n \text{TF-IDF}(si)*\text{TF-IDF}(sj)}{\sqrt{\sum_{t=1}^n (\text{TF-IDF}(sj))^2} * \sqrt{\sum_{n=1}^n (\text{TF-IDF}(si))^2}} \tag{10}$$

(si): est la similarité en cosinus entre les deux phrases vecteurs (si), (sj) Où Sim-Score (TF-IDF(si)) TF-IDF(sj)) respectivement.

TF-IDF pour les phrases S est le vecteur TF-IDF et n est le nombre de termes dans, 98 sont TF-IDF de la la phrase.

Le numérateur est un produit scalaire entre les deux vecteurs. Le dénominateur est la multiplication du grandeur de ces deux vecteurs.

3-1-8-4- Construire les solutions candidates Graph

Le graphe acyclique dirigé (DAG) est utilisé pour représenter les relations sémantiques entre les phrases et les documents [24]. Le graphe est composé de deux ensembles (V, E) où V est l'ensemble des sommets et E est l'ensemble des arêtes. Chaque phrase dans le document est considéré comme un sommet du graphe.

Le poids de chaque bord montre la similitude entre deux Phrases si,sj.

Nous devons ajouter chaque phrase du document à la liste des sommets. Alors pour toutes les deux phrases si,sj si phrase si est apparue avant sj dans le document et que son score sémantique n'est pas nul, ajoutez une arête de si a sj au bord.

3-1-8-5- Application de l'algorithme PSO

Le PSO est un optimiseur basé sur la population où il se compose d'un essaim de particules. Le système est initialisé avec un ensemble de solutions potentielles générées aléatoirement, puis PSO recherche la solution optimale de manière itérative. il trouve la solution optimale par essais suivant la meilleure particule.

Dans PSO, chaque solution est appelée une particule dans le espace de recherche. Toutes les particules ont des valeurs de fitness qui sont évaluées par la fonction de fitness à optimiser, et ont vitesses qui dirigent le vol des particules. Les particules volent à travers l'espace problème en suivant le particules optimales actuelles. PSO comprend les étapes suivantes [17] :

- Initialiser la population : recherchez la valeur de fitness de chaque particule de la population à l'aide de la fonction de fitness.
- Trouvez la meilleure solution actuelle : trouvez la meilleure solution actuelle qui a la meilleure valeur de fitness.
- Calculer la vitesse et la position : chaque particule met à jour sa vitesse et sa position en fonction de sa meilleure position, sa vitesse actuelle et quelques informations sur ses voisins. Les équations (12) et (13) sont utilisées pour mettre à jour la vitesse et position respectivement

ALGORITHM
 Step 1: Input data
 Step 2: Stream data.
 Step 3: Pre-process the input data. *Tokenization*→*Stop word removal*→*Stemming*
 Step 4: Build the vector space model TFIDF

$$tf_{t,d} = \frac{f_d(t)}{\min_{\omega \in d} f_d(\omega)}$$

$$idf_{t,d} = \ln \left[\frac{|D|}{|d \in D : t \in d|} \right]$$

$$tfidf\ t,d,D = tf\ t,d \cdot idf\ (t,D)$$

$f\ a\ t$:= frequency of term t in document d D :=Corpus of documents

Step 5: Make the cluster of the features.

Step 6: Label the TF-IDF to the features.

Step 7: Use the meta-heuristic ACO and PSO for the optimization of the features.

Step 8: Classifies the optimized features using Support Vector Machine and Naïve Bayes Classifier.

Step 9: Test the classifier model.

Step 10: Analyse the Accuracy, Precision and Recall.

3-1-9-. Résultats expérimentaux et évaluation

Le corpus EASC [26] et la boîte à outils ROUGE [7] sont utilisés pour déterminer la précision de l'approche proposée.

Les résultats expérimentaux sont présentés et comparés aux résultats de deux approches évolutives clés, l'approche HS [14] et approche GA [15]. Les réglages des paramètres utilisés dans les expériences PSO sont illustrés dans le tableau 1. Le même les réglages des paramètres sont utilisés à la fois dans les tests ROUGE-1 et ROUGE-2.

Parameter	Valeur
Nombre d'itérations	100
Taille Population	25
Coefficient d'accélération personnel (c1)	2
Coefficient d'accélération sociale (c2)	2
Nombre d'itérations	100

Tableau 3-5: PSO Paramétrage

Le tableau 2 : présente les résultats de comparaison pour ROUGE-1 et ROUGE-2 pour l'approche proposée et les deux approches évolutives clés sur le même corpus. Il ressort de ce tableau que la proposition basée sur les OSP approche atteinte La précision et la mesure F ont surpassé ces approches.

Approach	ROUGE	Recall	Precision	F-measure
PSO	ROUGE-1	0.5444	0.5882	0.5532
PSO	ROUGE-2	0.4483	0.4814	0.4538
GA	ROUGE-1	0.5713	0.5658	0.5476
GA	ROUGE-2	0.4710	0.4597	0.4465
HS	ROUGE-1	0.5758	0.5686	0.5495
HS	ROUGE-2	0.4810	0.4679	0.4540

Tableau 3-6-: résumé les résultats de l'approche pour ROUGE -1 et ROUGE-2

3-2 implémentation

3-2-1- Réalisation de système

Après avoir présenté la conception de notre système et mentionné ses étapes les plus importantes. Dans cette section, nous montrerons comment mettre en œuvre ces étapes présenterons les résultats les plus importants obtenus.

3-2-1-1- Ressources utilisées

Hardware :

Dans notre expérimentation, nous avons utilisé un PC était de marque HP pro3330, avec processeurs multi-cœuré I3, et de horloges de fréquence de 2.40 GHZ et de RAM de 4 GO.

Software :

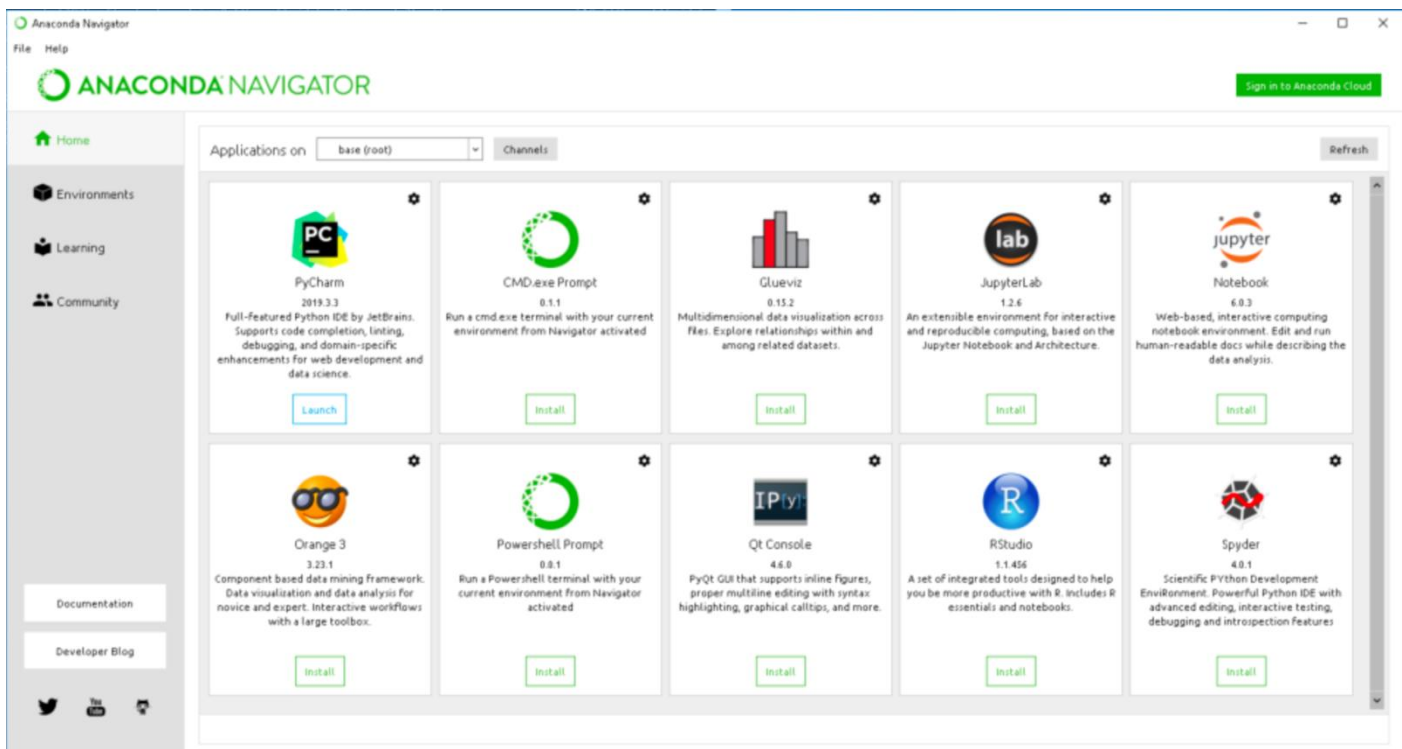
3-2-1-2-Plate-forme utilisé

Anaconda :

Anaconda est une distribution libre et open source dédiée à la programmation Python et R. elle est très utilisée dans la science des données, machine Learning et l'intelligence artificielle.

Cette distribution est devenue indispensable pour n'importe quel développeur dans le domaine de la data science.

Anaconda propose une variété d'outils de collecte et transformation de données à grande échelle. Elle est aussi connue pour sa richesse en modules et bibliothèques de la data science.



. Figure 3-3 : Logo anaconda Navigator

3-2-2- Choix du langage de programmation

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

Python (langage)

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



Figure 3-4 Logo langage python

Ce choix a été motivé par les raisons suivantes :

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

3-2-3- Exemples de codes sources

Dans cette section, nous allons présenter quelques exemples de codes sources.

La Figure 3-5 présente un morceau de code qui permet d'appeler les bibliothèques nécessaires pour compiler notre application.

```

[ ]: 1
[ ]: 2 #--- import les bibliothèques -----
3 import pandas as pd
4 import numpy as np
5 import matplotlib.pyplot as plt
6 import string
7 import re
8 from nltk.corpus import stopwords
9 from sklearn.model_selection import train_test_split, GridSearchCV
10 from sklearn.feature_extraction.text import TfidfVectorizer
11 from sklearn.pipeline import make_pipeline
12 from sklearn.linear_model import LogisticRegression
13 from sklearn.ensemble import RandomForestClassifier
14 from sklearn.naive_bayes import MultinomialNB
15 from sklearn.svm import SVC
16 from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
17
18
19 import tweepy
20 import pandas as pd
21 access_token = "*****"
22 access_token_secret = "*****"

```

La Figure 3-5- import les bibliothèques

La Figure 3-6 présente les instructions nécessaires pour lire le Dataset et le dictionnaire.

```

19 import tweepy
20 import pandas as pd
21 access_token = "*****"
22 access_token_secret = "*****"
23 API_key = "*****"
24 API_secret_key = "*****"
25 auth = tweepy.OAuthHandler(API_key, API_secret_key)
26 auth.set_access_token(access_token, access_token_secret)
27 api = tweepy.API(auth)
28 timeline_tweets = api.home_timeline()
29 for key, value in timeline_tweets[0]._json.items():
30     if type(value) is dict:
31         print(key, list(value))
32     else:
33         print(key)
34     variables_set = [[tweet.user.name, tweet.user.location, tweet.text, tweet.retweet_count] for tweet in timeline_tweets]
35 my_data = pd.DataFrame(data=variables_set, columns=['user', 'location', 'tweets', 'retweet'])
36 my_data
37 query = "ar"
38 language = "ar"
39 twitter_data = api.search(q=query, lang=language)
40 variables_set = [[tweet.user.screen_name, tweet.user.location, tweet.text, tweet.retweet_count] for tweet in twitter_data]
41 my_data = pd.DataFrame(data=variables_set, columns=['user', 'location', 'tweets', 'retweet'])
42 my_data

```

```

43 name = "spagov"
44 tweetCount = 20
45 timeline_tweets = api.user_timeline(id=name, count=tweetCount)
46 variables_set = [[tweet.created_at, tweet.text, tweet.retweet_count] for tweet in timeline_tweets]
47 my_data = pd.DataFrame(data=variables_set, columns=['date and time', 'tweets', 'retweet'])
48 my_data
49 import sys
50 from tweepy import Stream
51 from tweepy.streaming import StreamListener
52
53 class Listener(StreamListener):
54     def __init__(self, output_file=sys.stdout):
55         super(Listener, self).__init__()
56         self.output_file = output_file
57     def on_status(self, status):
58         print(status.text, file=self.output_file)
59     def on_error(self, status_code):
60         print(status_code)
61         return False
62     output = open('twitter_data.txt', 'w')
63 listener = Listener(output_file=output)
64
65 stream = Stream(auth=api.auth, listener=listener)
66 #ps:

```

```

63 listener = Listener(output_file=output)
64
65 stream = Stream(auth=api.auth, listener=listener)
66 try:
67     print('Start')
68     stream.filter(track=['ar'])
69 except KeyboardInterrupt:
70     print("Stopped")
71 finally:
72     stream.disconnect()
73     output.close()
74 import os
75 os.chdir('Desktop')

```

La Figure 3-6- lire le Data set

La Figure 3-7 présente le code source qui permet Pré-traitement de texte

```
89
90
91 punctuations = '''+x-“”!|+|~{}',.!"#%&*()_<>:''' + string.punctuation
92 # Arabic stop words
93 stop_words = stopwords.words()
94 arabic_diacritics = re.compile("""
95     ' | # Shadda
96     ' | # Fatha
97     ' | # Tanwin Fath
98     ' | # Damma
99     ' | # Tanwin Damm
100    ' | # Kasra
101    ' | # Tanwin Kasr
102    ' | # Sukun
103    - | # Tatwil/Kashida
104    """, re.VERBOSE)
105
106 def preprocess(text):
107     '''
108     text is an arabic string input
109
110
111
112
113
114
115     text = text.translate(translator)
116
117     # remove Tashkeel
118     text = re.sub(arabic_diacritics, '', text)
119
120     #remove Longation
121     text = re.sub("[\u0640]", "", text)
122     text = re.sub("\u0641", "\u0642", text)
123     text = re.sub("\u0643", "\u0644", text)
124     text = re.sub("\u0645", "\u0646", text)
125     text = re.sub("\u0647", "\u0648", text)
126     text = re.sub("\u0649", "\u064a", text)
127
128     text = ' '.join(word for word in text.split() if word not in stop_words)
129
130     return text
131
132 data['Feed'] = data['Feed'].apply(preprocess)
133 print(data.head(5))
```

Figure 3-7 Pré-traitement

La Figure 3-8 présente le code source qui permet de compter les mots négatifs

```
[ ]: CN =[]
count = 0
for t in corpus.text:
    for word in t.split():
        for word1 in Neg.text:
            if word==word1:
                count+=1
        CN.append(count)
        count=0
corpus["neg_count"]=CN
```

La Figure 3-8 compter les mots négatifs

La Figure 3-9 présente Exemple de niveau de sentiments

1	0.963	حب	1	0.000	وكيل	1	-0.313	هرب
2	0.925	فرح	2	0.000	عند	2	-0.325	صعب
3	0.925	نجاح	3	0.000	احصي	3	-0.375	صعيب
4	0.912	سرور						
5	0.912	مبروك						
6	0.900	صادق						
7	0.900	متفائل						
8	0.900	سعادة						
9	0.875	نعيم						
10	0.875	مبدع						
11	0.875	سلام						
12	0.838	رائع						

Figure 3-9 : Exemple de niveau de sentiments.

3-2-4- Exemples de sorties

Malgré nous avons abouti une bonne mesure, notre application a effectué quelques erreurs. Ces erreurs sont listées dans le tableau 3-7.

Exemple	Publication	Notre annotation	Résultat système
1	عيد سعيد و مبارك و كل عام و أنت بألف خير إن شاء الله	1	1
2	سفيان فيغولي يتطوع لإحدى المائدات الإفطارية للمسلمين في فرنسا ، برافو	1	1
3	هذا فعل لا اخلاقي اين تعليم سيدنا محمد(ص) عندما كان جاره يهودي وعندما مرت جنازة يهودي بالله عليكم ماذا فعل؟؟؟	-1	-1
4	مالقيتو ما ديرو	-1	-1
5	انا حاب كاس رايب و ربع كسرة خير من لفريت	0	0
6	إذا عرفتو في اقل من 5 ثواني دير جام	0	0
7	الجميع يشهد أن # الدون هو الافضل ولكن عبودي الكتلوني لديه رأي آخر	0	1
8	مهابل وتلاقو	0	-1
9	يعمري هديك اتبسيمة	1	0
10	هذا الانسان ماهوش رجل وطني	-1	0

tableau 3-7 Exemples de sorties

3.3 Conclusion

Dans ce chapitre, nous avons fait beaucoup d'effort dans deux volets important dans ce travail. Le premier est l'annotation d'un corpus de 2891 publications écrits en dialecte algérien par positive, négative et neutre. Le deuxième, celui de la création d'un dictionnaire de 1328 mots aussi en dialecte algérien.

L'étape la plus importante pour atteindre les résultats souhaités en utilisant l'apprentissage profond est la bonne connaissance de la sélection des données appropriées et la meilleure façon de les représenter.

Dans ce chapitre, nous allons exposé la conception de notre système avec l'explication de méthode de transformation de notre données ainsi que l'algorithme d'apprentissage choisi.

4-Conclusion générale et perspectives

Notre travail fournit un grand effort pour essayer de lutter contre la propagation du langage abusif dans le dialecte algérien sur les réseaux sociaux. En outre, notre système permet de classifier le langage abusif comme obscène, offensant ou normal.

Nous avons étudié les caractéristiques de la langue arabe en général et le dialecte algérien en particulier et sa complexité, nous avons créé une liste des mots abusifs du dialecte algérien, cette dernière a été utilisée dans la collection de données à partir de Facebook et Twitter. De plus, nous avons utilisé notre modèle PSO afin d'agrandir de façon automatique cette liste. Ce modèle est créé en utilisant seulement les données recueillies par Facebook car les plus part données de twitter représenté que les pays de Golf.

Notre approche exploite différents algorithmes d'apprentissage automatique et profond, qui ont été entraînés sur nos propres données.

Ces algorithmes ont été évalués et on a obtenu des bons résultats dans les algorithmes d'apprentissage automatique et d'excellents résultats dans l'algorithme d'apprentissage profond. Nous ne sommes pas au courant des autres travaux existants de mots ou des contenus abusifs en utilisant le dialecte algérien. Notre travail apporte les contributions suivantes :

- Une liste initiale de 525 des mots abusifs du dialecte algérien de différentes régions.
- Un ensemble des Id de 50 pages algériennes plus connues et utilisée sur Facebook.
- Un ensemble de plus de 80K commentaires et statuts via Facebook.
- Un modèle word2vec à partir d'un corpus de dialecte Algérien qui contient 108K mots. Ce modèle peut être utilisé dans des travaux connexes.
- Un ensemble de plus de 250K tweets via Twitter API.

Les différentes pistes explorées pendant ce travail nous ont amenées à envisager de nombreuses perspectives.

Nous présentons ici celles qui nous paraissent les plus prometteuses.

Nous essayons de généraliser ce travail sur la langue amazighe et d'autres dialectes tels que les dialectes maghrébins dans les pays voisins comme la Tunisie et le Maroc, et aussi d'intégrer d'autres sites des réseaux sociaux qui utilisent d'autres moyens de discours telles que la vidéo et la voix à titre d'exemple YouTube.

Pour les travaux futurs, nous pouvons citer :

- L'enrichissement de notre dictionnaire par plus des mots de dialecte algérien en couvrant plus largement d'autres zones car l'Algérie est très vaste et il contient des dizaines de dialectes,
- L'enrichissement de Dataset par d'autres commentaires en dialecte algérien afin d'obtenir des résultats bien précis.
- L'application d'autres classificateurs et l'utilisation d'autres fonctionnalités.
- L'analyse par l'utilisation de la classe Mixte autre que les classes positive, négative et neutre.
- L'utilisation des autres configurations tel que bigramme, trigramme et mixte.

5- Bibliographie

- [1] Avis client sur Facebook : danger ou opportunité pour les marques, <http://www.brand-advocacy.fr/avis-facebook-marques/>, Visité le 15/02/2019.
- [2] Sébastien Gillot, Fouille d'opinions, Colloque du Master Recherche en Informatique, 2010, 1-35.
- [3] Flora Even, l'influence de Facebook sur les idées politiques, https://www.rtb.be/culture/dossier/chroniques-culture/detail_l-influence-de-facebook-sur-les-idees-politiques-flora-eveno?id=9458372, Visité le 16/02/2019.
- [4] Sentiment définition et synonyme, <https://www.mediadico.com/dictionnaire/definition/sentiment/>, Visité le 20/02/2019.
- [5] Opinion définition, <https://www.larousse.fr/dictionnaires/francais/opinion/56197>, Visité le 22/02/2019.
- [6] Mathieu Troillet, Avantages et inconvénients des réseaux sociaux, en particulier «Facebook », pour la promotion dans les secteurs socioprofessionnels, école supérieure - Sion Suisse, 2015.
- [7] Lexique, <https://www.notrefamille.com/dictionnaire/definition/lexique>, 05-03-2019.
- [8] Qu'est-ce qu'un corpus, http://theses.univ-lyon2.fr/documents/getpart.php?id=lyon2.2002.carras_c&part=53655, Visité le 25/02/2019.
- [9] Bastien L, Machine Learning – Définition, fonctionnement et secteurs d'application, <http://www.artificiel.net/machine-learning-definition>, Visité le 05-03-2019.
- [10] Claire Gardent, Traitement des Langues Naturelles, LORIA Nancy France, 2011.
- [11] Vapnik, <https://scholar.google.com/citations?user=vtegaJgAAAAJ&hl=en>, Visité le 05-03-2019.
- [12] Jamal Kharroubi, Etude de techniques de classement "Machines à vecteurs supports" pour la vérification automatique du locuteur, école nationale supérieure des télécommunications Paris, 2002.
- [13] Abe Burrows, Arbres de décision, <http://cedric.cnam.fr/vertigo/Cours/ml2/coursArbresDecision.html>, Visité le 01/03/2019.
- [14] Fabien Moutarde, Arbre de Décision et Forêts aléatoires, Centre de Robotique CAOR MINES ParisTech ENSMP- PSL research University, 2017.
- [4]. <https://www.larousse.fr/dictionnaires/francais/%C3%A9motion/28829>(consulté le 10 Février 2018).
- [5]. Gabriel Dabi-Schwebel, Microblogage "microblogging". 14 Avril 2014. AGENCE 1MIN30 . <https://www.1min30.com/dictionnaire-du-web/microblogage> (consulté le 15 mai 2018).
- [6]. "Softmax" <http://www.statsoft.fr/conceptstatistiques/glossaire/s/softmax.html>(consulté le 08 juillet 2018).
- [7]. Jason Brownlee, "What Are Word Embeddings for Text?". October 11, 2017 <https://machinelearningmastery.com/what-are-word-embeddings/> (consulté le 20 mai 2018).
- [8]. "word2vec". Jul 30, 2013. <https://code.google.com/archive/p/word2vec/> (consulté le 20 mai 2018).
- [9]. "Words as Vectors". April 13, 2015, <https://iksinc.online/tag/continuous-bag-of-words-cbow/>
- [10]. Serhiy. 12 février 17 à 11:38 .<https://stackoverflow.com/questions/38287772/cbow-v-s-skip-gram-why-invert-context-and-target-words>
- [11]. <https://www.tensorflow.org/>
- [12]. <https://keras.io/>
- [13] <http://qatsdemo.cloudapp.net/farasa/>
- [14]. Sara Rosenthal and Noura Farra and Preslav Nakov , "Data and Tools" August 2017 ,

<http://alt.qcri.org/semeval2017/task4/index.php?id=data-and-tools>

- [15]. Abu Bakr Soliman Mohammad , ‘ ‘ AraVec ’ ’ , November 2017 [https :// www researchgate.net/publication/319880027](https://www.researchgate.net/publication/319880027).
- [17] Yoann D., Marco D., Isabelle T.,2017. Réseaux neuronaux profonds pour l'étiquetage de séquences
- [18] Jiwei L. , Xinlei C., Eduard H., Dan J., Visualizing and Understanding Neural Models in NLP.
- [19] Gobinda G. Chowdhury, Natural Language Processing . Dept. of Computer and Information Sciences,University of Strathclyde, Glasgow, UK
- [20] Akshay K.,Adarsha S., Natural Language Processing Recipes Unlocking Text Data with Machine Learning and Deep Learning using Python
- [21] Taweh Beysolow, Applied Natural Language Processing with Python Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing
- [22] Richard S, Yoshua B., Christopher D. Manning, Deep Learning for NLP.
- [23] Vincent B.,2017. Machine Learning en Finance Vers de nouvelles stratégies.
- [24] Josh Patterson & Adam Gibson ,2017. Deep Learning A Practitioner's Approach, 1ère (Ed), O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472 , Mike Loukides & Tim McGovern, 532 p, (pp. 28)
- [25] Palash G.,Sumit P. & Karan J.,2018. Deep Learning for Natural Language Processing: Creating Neural Networks with Python, India , Celestin Suresh John, 290p,(pp. 1)
- [26] Yann LeCun, Yoshua Bengio & Geoffrey Hinton., may 2015 REVIEW Deep learning
- [27] Ronan C., Jason W., Léon B., Michael K., Koray K. ,Pavel K., 2011. Natural Language Processing (almost) from Scratch
- [28] De <https://link.springer.com/article/>.
- [29] De <https://interstices.info/la-revolution-de-lapprentissage-profond/>.
- [30] De <https://www.gralon.net/articles/materiel-et-consommables/materiel-informatique-etconsommable-informatique/article-comprendre-le-deep-learning-en-6-questions-10956.htm> .
- [31] De <https://www.saagie.com/fr/blog/qu-est-ce-que-le-deep-learning/> .
- [32] De <https://fr.mathworks.com/discovery/deep-learning.html> .
- [33] De <https://stackabuse.com/text-classification-with-python-and-scikit-learn/>
- [34] De <https://www.zdnet.com/article/what-is-deep-learning-everything-you-need-to-know/>
- [35] De <https://www.supinfo.com/articles/single/6041-machine-learning-introductionapprentissage-automatique>
- [36] De <http://www.fullai.org/short-history-artificial-intelligence/> .
- [37] De <https://www.oreilly.com/library/view/python-natural-language/>.
- [38] De <http://blogshells.com/how-deep-learning-and-data-science-work-with-natural-languageprocessing/>
- [39] De <http://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning/>
- [40] De <https://www.gralon.net/articles/materiel-et-consommables/materiel-informatique-etconsommable-informatique/article-comprendre-le-deep-learning-en-6-questions-10956.htm>.
- [41] De <https://www.kaggle.com/13nnys/useful-text-preprocessing-on-the-datasets>
- [42]: Wiebe J. & Mihalcea R. (2006). Word sense and subjectivity. In proceedings of the 21st international

conference on computational linguistics and 44th annual Meeting of the Association for computational linguistics Sydney, Australia. ACL.

- [43]: Tu H. & Hatzivassiloglou V. (2003). *Towards Answering Opinion Questions: Separating Facts from Opinion and Identifying the polarity of opinion Sentences*, in *proceeding of the 2003 conference on Empirical Methods in Natural Language Processing (EMNLP 2003)*, Sapporo, japan. ACL.
- [44]: Pang B. & Lee L. 2004. *A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts*. In *proceedings of the 42nd annual meeting of the ACL, Barcelona Spain, July*. ACL.
- [45]: Stéphane Ferrairi, Yann Mathet, Thierry Charnois: *Analyse d'opinion: discours évaluatif et classification de documents*.
- [46] : Hu M. & Liu B. 2004. *Mining Opinion Features in Customer Reviews*. In *proceeding of nineteenth national conference on artificial intelligence (AAAI-2004)*, San Jose, USA.
- [47] Bing Liu, *opinion mining*, university of Illinois at Chicago, 851 S. Morgan Street, Chicago, IL 60607-0753.
- [48]: Dave, K., Lawrence, S., & M., Pennock D. 2003. *Mining the Peanut Gallery : Opinion Extraction and Semantic Classification of Product Reviews*. Pages 519–528 of : *Proceedings of WWW*. 35, 41, 44
- [49] : Melle Trahi Fatiha, « *Prédiction de l'irradiation solaire globale pour la région de Tizi-ouzou par les réseaux de neurones artificiels. Application pour dimensionnement d'une installation photovoltaïque pour l'alimentation du laboratoire de recherche LAMPA* ». *Memoire de Magister en électronique, soutenue 2011*
- [50] : Melle Kasri Souad, *Etude et Modélisation de Déclin de Potentiel de Surface par les Réseaux de Neurones*, *Présenté en vue de l'obtention du diplôme DE MAGISTER, Université Badji Mokhetar-Annaba, Sciences de l'ingénieur D'Electromécanique, soutenue 2010*.
- [51] : F.Junod, M.Bornoz, « *A la découverte des réseaux de neurones* », Yverdon, le 7 juin 2002.
- [52] : S.M.Ait cheikh, « *Etude, Investigation et conception d'algorithmes de commande appliqués aux systèmes photovoltaïques* », *thèse de Doctorat d'Etat en électronique, Ecole Nationale Polytechnique, soutenue en Décembre 2007*.
- [53] : F.Elle, « *Conception et réalisation d'un système utilisant des réseaux de neurones pour l'identification et la caractérisation, à bord de satellites, de signaux transitoires de type sifflement* », *thèse de Doctorat en électronique, Université d'Orléans soutenue le 23 Septembre 1997*.
- [54] : G.Dreyfus, « *les réseaux de neurones* », *mécanique Industrielle et Matériaux*, n°51, Septembre 1998.
- [55] : O.Benzaim, « *Techniques multi-port pour la conception et la réalisation de systèmes micro-ondes dédiés à l'évaluation non destructive de matériaux* », *thèse de Doctorat en micro et nano technologies, acoustique et télécommunications, Université Lille 1 sciences & technologies, soutenue le 7 juillet 2009*.
- [56] : M.Parrizeau, « *Réseaux de neurones* », *presse de l'université de Laval, Automne 2004*.
- [57] : Dr.Guerram, *cour Master1 Informatique*, 2013.
- [58] : Matthieu vandeweghe, *Introduction au langage java, Presentation & Historique*, 2004.