REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'enseignement supérieur et de la recherche scientifique

Université Saad Dahleb Blida 1 Faculté des Sciences

Département d'Informatique

Projet de fin d'études en vue de l'obtention du

Diplôme de Master II en Systèmes informatiques et réseaux

Thème :

Identification des propagateurs de Hate Speech sur les réseaux sociaux.

Présenté par

TAFAT BOUZID Yousra

Soutenu le 14/07/2021.

Devant le jury :

Présidente :	Ykhlef Hadjer	MCB	Université Saad dahleb
Examinatrice :	Guessoum Dalila	MAA	Université Saad dahleb
Promotrice :	Madani Amina	MCB	Université Saad dahleb
Co-promotrice :	Boumahdi Fatima	MCA	Université Saad dahleb

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon travail et qui m'ont aidée lors de la rédaction de ce mémoire.

Je voudrais dans un premier temps remercier Madame **MADANI Amina** pour sa patience et surtout ses judicieux conseils, qui ont contribué à mon avancement durant toute l'année universitaire.

Je remercie également Madame **BOUMEHDI Fatima** pour sa disponibilité à n'importe quel moment.

Je tiens à témoigner toute ma reconnaissance à Monsieur **HENTABLI Hamza** qui n'a pas cessé a nous dirigé coté programmation jusqu'à la dernière minute.

Enfin je remercie vivement **les membres du jury** de nous avoir fait l'honneur d'examiner notre travail.

Dédicaces

Avec l'expression de ma reconnaissance je dédie ce modeste travail à ceux qui quels que soient les termes je n'arriverais jamais à leur exprimer mon amour sincère.

A l'homme, mon précieux offre du dieu, mon épaule solide, à la personne la plus digne de mon estime et de mon respect, qui n'a jamais dit non à mes exigences, aucune dédicace ne saurait exprimer mes sentiments, que dieu te préserve et te procure santé et langue vie, mon papa chéri.

A la femme qui a souffert sans me laisser souffrir, Autant de phrases aussi expressives soient-elles ne sauraient montrer le degré d'amour et d'affection que j'éprouve pour toi. Tu m'as comblé avec ta tendresse et affection tout au long de mon parcours. Tu n'as cessé de me soutenir et de m'encourager durant toutes les années de mes études, tu as toujours été présente à mes côtés pour me consoler quand il fallait.

En ce jour mémorable, pour moi ainsi que pour vous deux. Que dieu le tout puissant vous donne santé, bonheur et longue vie afin que je puisse vous combler à mon tour.

A la mémoire de mon frère « Nour El Islam » que dieu vous accueille dans son vaste paradis, et que ce travail soit une prière pour votre âme.

Résumé :

Les réseaux sociaux sont devenus en très peu de temps un phénomène d'une ampleur inouïe, censé permettre aux inscrits d'interagir entre eux en échangeant des informations, des photos et des actualités de tous ordres.

Cependant, certains utilisateurs utilisent ces réseaux à des fins différentes en diffusant des contenus abusifs, qui jouissent à des personnes, des groupes ou même des communautés entières.

En effet, la détection de ce phénomène nommé par l'hate speech est devenue une tache cruciale et un défi majeur à surmonter.

Notre étude est orientée vers la détection des discours haineux en se basant sur l'apprentissage profond et automatique comme algorithme sur la plateforme « Twitter » et avec la langue anglaise. Nous avons développé deux modèles (Random Forest et la combinaison entre Random Forest et Auto-encodeur) qui nous ont permis d'arriver à classer nos données et nous donner les meilleurs résultats de précision dans ce domaine.

Mots clé : détection des discours haineux, apprentissage profond, apprentissage automatique, hate speech, contenus abusifs, Random Forest, Auto-encodeur.

Abstract:

Social networks have quickly become a phenomenon of unprecedented magnitude, supposed to allow registrants to interact with each other by exchanging information, photos and news of all kinds.

However, some users use these networks for different purposes by distributing abusive content, which is enjoyed by individuals, groups or even entire communities.

Indeed, the detection of this phenomenon named by hate speech has become a crucial task and a major challenge to overcome. Our study is oriented towards the detection of hate speech based on deep and machine learning as an algorithm on the “Twitter” platform and with the English language.

We have developed two models (Random Forest and the combination between Random Forest and Auto-encoder) which allowed us to classify our data and give us the best precision results in this area.

Keywords: abusive content, hate speech, deep learning, machine learning, Random Forest, Auto-encoder.

ملخص

سرعان ما أصبحت الشبكات الاجتماعية ظاهرة ذات حجم غير مسبوق، من المفترض أن تسمح للمسجلين بالتفاعل مع بعضهم البعض من خلال تبادل المعلومات والصور والأخبار من جميع الأنواع. ومع ذلك، يستخدم بعض المستخدمين هذه الشبكات لأغراض مختلفة من خلال توزيع محتوى مسيء يتمتع به الأفراد أو المجموعات أو حتى المجتمعات بأكملها.

في الواقع، أصبح اكتشاف هذه الظاهرة التي يطلق عليها خطاب الكراهية مهمة حاسمة وتحديًا كبيرًا يجب التغلب عليه.

دراستنا موجهة نحو الكشف عن خطاب الكراهية على أساس التعلم العميق والآلي كخوارزمية على منصة "Twitter" وباللغة الإنجليزية.

لقد قمنا بتطوير نموذجين (Random Forest والجمع بين Random Forest و Auto-encoder) مما سمح لنا بتصنيف بياناتنا ومنحنا أفضل النتائج الدقيقة في هذا المجال.

الكلمات المفتاحية : محتوى مسيء, خطاب الكراهية, التعلم العميق والآلي, الشبكات الاجتماعية

Sommaire

Chapitre 3 : Détection du Hate Speech - Etat de l'art

Chapitre 4 : Conception d'une méthode de détection du hate speech à partir des réseaux sociaux

Chapitre 5 : Expérimentations et résultats

Liste d’acronymes

HS : Hate Speech (discours de haine).

IA : Intelligence Artificielle.

ML : Machine Learning (Apprentissage Automatique).

DL : Deep Learning (Apprentissage Profond).

ANN : Artificial Neural Network (Réseau de neurones artificiels).

DNN : Deep neural network.

NLP : Natural Language Processing (Traitement automatique des langues).

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

RF : Random Forest (Forêt d'arbres décisionnels).

AE : Auto-Encodeur

CNN : Convolutional Neural Network (Réseau neuronal convolutif).

CONV : Couche convolutionnelle.

RNN : Réseau de neurones récurrents **(**Recurrent Neural Network).

LSTM: Long Short-Term Memory.

BILSTM: Bidirectional Long Short-Term Memory.

TAL : Traitement automatique des langues.

LR: Regression logistique.

CBOW: Continuous Bag of Words.

TF-IDF: Short for term frequency–inverse document.

SemEval: International Workshop on Semantic Evaluation.

CLEF: Conference and labs of the evaluation forum.

Liste des figures

Liste des Tableaux

Introduction générale

Dans notre société, les réseaux sociaux ont une importance capitale. Avec l'essor de la technologie, ils sont présents au quotidien et sont devenus un point de rencontre des individus et de communautés en leur permettant la création et l'échange d'information via internet. Bien qu'ils contiennent beaucoup d'avantages, ces nouveaux moyens de communication ont pleins de risques tels que le discours de haine.

Les réseaux sociaux constituent une arène majeure pour la diffusion de discours de haine en ligne. Cela contribue de manière significative à la difficulté de la détection automatique, car les publications sur les réseaux sociaux incluent des signaux paralinguistiques (par exemple, des émoticônes et des hashtags) et leur contenu linguistique contient beaucoup de texte mal écrit. Une autre difficulté est présentée par la nature dépendante du contexte de la tâche et le manque de consensus sur ce qui constitue un discours de haine, ce qui rend la tâche difficile même pour les humains.

Pour cette raison, cette problématique a attiré beaucoup de chercheurs et de nombreuses méthodes ont été proposées afin de trouver des solutions qui peuvent faire face à ce phénomène pour maintenir un équilibre entre la liberté d'expression , le respect de l'égalité , de la dignité et pour éviter les divisions et des alliances sans précédent sur la sécurité sociale sur le plan international.

Ce travail vise à la conception et le développement d'une nouvelle solution de classification du contenu des tweets dont l'objectif est de pouvoir estimer automatiquement si un message présente un discourt haineux ou pas.

Cette étude permet d'utiliser les techniques d'apprentissage en profondeur et d'apprentissage automatique. Nous avons proposé une approche basée sur deux modèles différents. Le premier est l'algorithme Random Forest d'apprentissage supervisé, Dans le deuxième modèle, nous combinons entre le Random Forest et l'algorithme Auto-Encodeur qui est un algorithme d'apprentissage non supervisé à base de réseaux de neurones artificiels. Nous

avons utilisé deux dataset, le premier est celui venu du site Kaggle[1] contenant 159571 tweets. Le deuxième est un dataset de la tâche « Profiling Hate Speech Spreaders on Twitter **»** de PAN [2] de CLEF 2021 contenant 200 users, 200 tweets pour chacun.

La structure de ce mémoire est organisée comme suit :

Le premier chapitre nommé « **Généralités sur les réseaux sociaux** » dans lequel nous intéresserons aux principes et concepts majeurs des réseaux sociaux.

Dans le deuxième chapitre appelé **« Deep Learning** », nous commençons par définir le Deep Learning ainsi que son utilisation dans les différents domaines, ensuite nous passerons aux techniques utilisées dans le domaine de notre recherche.

Le troisième chapitre est consacré pour «**Hate speech, Etat de l'art** ». Dans ce chapitre, nous définissons le hate speech et son utilisation dans les réseaux sociaux, puis nous présentons quelques travaux récents dans ce domaine.

Dans le quatrième chapitre intitulé « **Conception d'un modèle de détection du hate speech à partir des réseaux sociaux** », nous expliquons la conception et l'architecture de notre modèle proposé en détail.

Le dernier chapitre « **Implémentation et résultats** » décrit notre méthode testée, et à la fin nous partageons nos résultats avec ceux des travaux déjà réalisés.

[1] https://www.kaggle.com/
[2] https://pan.webis.de/

Chapitre 1 : Généralités sur les réseaux sociaux

1.1 Introduction

Depuis les deux dernières décennies, le monde des technologies d'informations a vécu une grande révolution et surtout dans le domaine des réseaux sociaux qui ont poursuivi une amélioration en prenant de plus en plus de place dans la vie habituelle de beaucoup d'individus.

En effet, l'utilisation quotidienne des réseaux sociaux comme Twitter , Facebook et Instagram ne cesse d'augmenter ,l' atteint désormais 4,14 milliards d'individus dans le monde, en hausse de 12,3 % sur un an, ce qui représente un total de 53 % de la population mondiale ou 453 millions de nouveaux usagers qui ont été enregistrés entre octobre 2019 et 2020 [1].

Parmi les réseaux sociaux les plus célèbres on trouve Twitter qui est un réseau social de micro-blocage géré par l'entreprise Twitter Inc., créé en mars 2006. Ce service permet aux internautes d'envoyer gratuitement de brefs messages « tweets »limités à 140 caractères, il compte 330 millions d'utilisateurs actifs chaque mois dans différents pays et avec différentes langues.

Dans ce chapitre, nous allons survoler brièvement et généralement le monde des réseaux sociaux ainsi que leurs différents types et leurs différentes utilisations tout en se focalisant à la fin du chapitre sur le réseau social Twitter.

1.2 Notions sur les réseaux

1.2.1 Définition d'un réseau

Un réseau est un ensemble d'éléments reliés entre eux et réglés de manière qu'ils puissent communiquer [2]. Les réseaux existent partout et dans différents domaines, on peut citer :

- Les réseaux Internet.
- Les réseaux sociaux.
- Les réseaux de neurones et de protéines. (Dans la biologie)
- Les réseaux d'aéroport, de bus. (Dans les transports)

1.3 Définition des réseaux sociaux

Les réseaux sociaux sur Internet sont des applications ayant comme objectif de relier des amis, des connaissances ou des associés.

Les réseaux présentent des orientations plus ou moins personnelles ou professionnelles, c'est-à-dire que l'objectif des utilisateurs peut être de retrouver des amis et de partager des outils avec eux (photos, messages, commentaires, applications ludiques...) ou de tisser un réseau professionnel (rencontrer des partenaires potentiels, trouver un nouvel emploi, trouver des collaborateurs, annoncer des événements ou des activités professionnelles...).

Le principe d'un réseau social est de retrouver des personnes que vous connaissez, qui à leur tour, vous permettront de rentrer en contact avec d'autres personnes. De fil en aiguille, votre réseau peut très vite devenir considérable. La communication est évidemment un élément central des réseaux sociaux qui proposent tous des outils de communication synchrones (chat ou vidéoconférence) et asynchrones (commentaires, forum) [3].

1.4 Origines des réseaux sociaux

La première personne à avoir représenté un réseau social est Jacob Levy Moreno au début des années 1930 [Moeno, 1933]. Son objectif étant de visualiser graphiquement un réseau social, il a représenté les personnes par des points et une relation entre deux personnes par des flèches. Au milieu du 20ième siècle, Cartwright et Harary sont les premiers à avoir appliqué la théorie des graphes dans l'analyse des réseaux sociaux. Le graphe est devenu par la suite la représentation adoptée par toutes les sciences manipulant l'analyse des réseaux sociaux, dont la sociologie, les mathématiques et l'informatique. [5]

La figure ci-dessous représente l’histoire des réseaux sociaux :

Figure 1 : Histoire des réseaux sociaux [6].

1.5 L'intérêt de l'utilisation des réseaux sociaux

Les réseaux sociaux sont des moyens indispensables dans notre société actuelle, et dans nos relations avec les autres. Ils visent à créer un tissu relationnel, et permettent de rester disponible avec la famille, les amis (qu'ils soient anciens ou nouveaux), les clients (dans le milieu professionnel ou même au sein d'une même entreprise) ainsi que les fans (pour les différentes célébrités). Ce sont des outils idéaux pour envoyer des messages, partager des idées avec une grande communauté de personnes appelée parfois « amis » par exemple sur Facebook. Il est possible aussi d'utiliser les réseaux sociaux dans le domaine commercial ou en marketing pour vendre des produits ou des services et bien évidemment, augmenter les ventes est sûrement un objectif prioritaire [4].

1.6 Types des réseaux sociaux

Les réseaux sociaux peuvent être classés selon différents types,

1.6.1 Les réseaux personnels et généralistes

Ce type est tourné vers des sujets généraux et divers (sport, musique, politique, etc.). L'objectif est de faire partager des passions et des idées avec le reste de la communauté. Exemple : Facebook, Twitter ...

1.6.2 Les réseaux de partage

Ce sont des plateformes dédiées au partage des multimédias (photos, sons, vidéos, etc.) entre internautes. L'objectif est de faciliter l'accessibilité aux sources multimédias pour les internautes d'une communauté. Parmi ces plateformes les plus populaires on peut citer YouTube.

1.6.3 Les réseaux professionnels

Ce sont les réseaux les plus performants au sens propre du terme. Ils offrent la possibilité de se connecter et de partager des informations en mode professionnelle. Parmi ces réseaux en peut citer LinkedIn qui permet de publier et partager des (CV) et de chercher d'embauche dans les entreprises et les organisations qui publient leurs annonces de travail dans le réseau.

1.6.4 Les réseaux personnels thématiques

Les réseaux personnels thématiques peuvent être vue comme des réseaux généralistes mais sont orientés autour d'une thématique (voiture, musique, etc.) [4].

1.7 La plateforme « Twitter »

Twitter est un système de «microblogging» qui vous permet d'envoyer et de recevoir de courts messages appelés tweets. Les tweets peuvent comporter jusqu'à 140 caractères et peuvent inclure des liens vers des sites Web et des ressources pertinents.

Les utilisateurs de Twitter suivent d'autres utilisateurs. Si vous suivez quelqu'un, vous pouvez voir ses tweets dans votre «chronologie» Twitter. Vous pouvez choisir de suivre des personnes et des organisations ayant des intérêts académiques et personnels similaires à vous. Vous pouvez

créer vos propres tweets ou vous pouvez retweeter des informations qui ont été tweetées par d'autres. Le retweet signifie que les informations peuvent être partagées rapidement et efficacement avec un grand nombre de personnes [7].

1.7.1 Pourquoi utiliser Twitter ?

Twitter est un système de «microblogging» qui vous permet d'envoyer et de recevoir de courts messages appelés tweets. Les tweets peuvent comporter jusqu'à 140 caractères et peuvent inclure des liens vers des sites Web et des ressources pertinentes.

Les utilisateurs de Twitter suivent d'autres utilisateurs. Si vous suivez quelqu'un, vous pouvez voir ses tweets dans votre «chronologie» Twitter. Vous pouvez choisir de suivre des personnes et des organisations ayant des intérêts académiques et personnels similaires à vous. Vous pouvez créer vos propres tweets ou vous pouvez retweeter des informations qui ont été tweetées par d'autres. Le retweet signifie que les informations peuvent être partagées rapidement et efficacement avec un grand nombre de personnes [7].

Twitter est devenu de plus en plus populaire auprès des universitaires ainsi que des étudiants, des décideurs, des politiciens et du grand public. Ce qui le rend particulièrement sous la recherche et l'analyse fréquente auprès des chercheurs et doctorants. Twitter vous permet de [7]:

- Promouvoir facilement votre recherche, par exemple en fournissant des liens vers vos articles de blog, articles de journaux et actualités.
- Atteindre rapidement un grand nombre de personnes via des tweets et des retweets
- Suivez les travaux d'autres experts dans votre domaine.
- Etablir des relations avec des experts et d'autres adeptes.
- Tenez-vous au courant des dernières nouvelles et développements et partagez-les instantanément avec les autres.
- Atteindre de nouveaux publics.
- Demander des commentaires sur votre travail et donner des commentaires aux autres.
- Suivre et contribuer aux discussions sur des événements, par exemple des conférences auxquelles vous ne pouvez pas assister en personne.
- Exprimez qui vous êtes en tant que personne.

1.7.2 Les caractéristiques d'un Tweet

Un tweet est un message informatif court déposé sur le réseau social Twitter. Il est caractérisé par :

- La date de sa publication.
- Le nom ou le pseudo de son rédacteur ainsi que sa photo de profil.
- Les nombres de réaction sur ce tweet.

1.8 Conclusion

Notre époque a été celle des progrès scientifiques les plus remarquables. L'avènement des réseaux sociaux confirme bien ce niveau de développement fulgurant. Aujourd'hui, les réseaux sociaux constituent les systèmes de communication les plus rapides et les plus fiables pour la vie quotidienne de l'homme ce qui les rend des outils de communication incontournables.

Dans ce chapitre nous avons introduits brièvement le concept des réseaux sociaux en général, passant par les différents types des réseaux sociaux en se basant sur le réseau social twitter.

Le monde des réseaux sociaux propose une toute collection d'étude et de recherche dans plusieurs domaines de l'informatique qui ne cesse de s'étendre selon le besoin, pour cela nous allons introduire dans le prochain chapitre le domaine de l'intelligence artificielle avec les différents sous domaine ce qui nous aidera à analyser le contenu des réseaux sociaux.

Chapitre 2 : Généralités sur le Machine Learning « ML »

2.1 Introduction

Depuis plus d'une décennie l'intelligence artificielle (IA) vit une accélération dans son développement et son adoption est tout autour de nous. Elle intervient chaque fois que nous cherchons un mot dans Google, une série sur Netflix, une vidéo sur YouTube,....

Parmi les meilleures révolutions dans ce domaine ces dernières années est le Deep Learning qui a bouleversé le monde et spécialement le monde de l'IA.

Dans ce chapitre, nous présenterons l'apprentissage automatique, ses méthodes, le Deep Learning. Nous expliquerons en détail les algorithmes de classification, quelques algorithmes profonds et nous terminerons par une brève explication sur les différentes mesures de performances.

2.2 Intelligence Artificielle

L'intelligence artificielle correspond à un ensemble de technologies qui permet de simuler l'intelligence et accomplir automatiquement des tâches de perception, de compréhension et de prise de décision. Ces techniques font particulièrement appel à l'utilisation de l'informatique, de l'électronique, des mathématiques (notamment statistiques), des neurosciences et des sciences cognitives. Historiquement, les travaux en IA démarraient dans les années 1950 avec les travaux d'Alan TURING. L'IA est devenue un domaine de recherche à l'été 1956. Avant 2000, les limites imposées par les capacités de calculs et de stockage n'ont pas permis de réaliser des avancées significatives dans le domaine de l'IA. En conséquence, l'intelligence artificielle s'est développée très fortement depuis plus de 10 ans avec une accélération dans les 5 dernières années. L'IA nécessite pour le moment des ressources considérables en données et en puissance de calcul pour apprendre efficacement. La recherche développe maintenant des techniques pour réduire la consommation d'énergie et limiter le besoin de données, et d'autres techniques pour permettre de généraliser une solution à plusieurs usages [8].

2.3 Apprentissage Automatique

L'apprentissage automatique(ML) est un sous-domaine de l'intelligence artificielle (IA). Il englobe la conception, l'analyse, le développement et l'implémentation des méthodes permettant à une machine de progresser par un processus systématique, afin de remplir des tâches difficiles par des moyens algorithmiques. Il adapte ses analyses et ses comportements en solution définie, en se basant sur l'analyse de données empiriques[3] provenant d'une base de données ou de capteurs [9]. En général, l'apprentissage automatique est la capacité de la machine à apprendre à faire mieux à l'avenir sur la base de ce qui a été connu dans le passé.

2.3.1 Méthodes d'apprentissage automatique

Dans cette partie, nous allons détailler les différentes méthodes de l'apprentissage automatique telles que : l'apprentissage supervisé, l'apprentissage non-supervisé, apprentissage semi supervisé et l'apprentissage par renforcement (voir Figure 2).

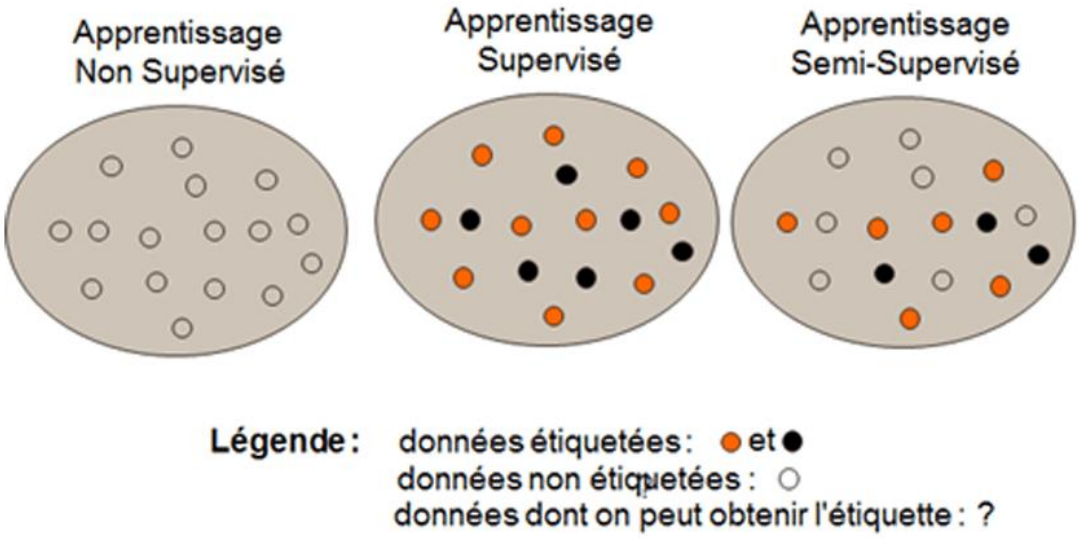


Figure 2 - les différentes méthodes de l'apprentissage automatique ML [9].

[3] C'est la recherche basée sur l'expérimentation dont l'objectif est de tester une hypothèse.

2.3.1.a Apprentissage supervisé

Dans cette méthode, le système apprend à partir d'un modèle connu et un échantillon d'apprentissage Dn = (x1, y1), . . ., (xn, yn) où xi = feature (propriété mesurable de l'objet que vous souhaitez l'analyser) et yi = target (est la variable dont les valeurs de cette variable doivent être prédites par d'autres variables features).

Exemple

Imaginez que nous avons une image de différentes catégories d'animaux, La tâche de notre modèle d'apprentissage supervisé est d'identifier les animaux et de les classer. Donc, pour identifier l'image dans l'apprentissage supervisé, nous allons donner les données d'entrée ainsi que la sortie pour cela, ce qui signifie que nous allons former le modèle par la forme, la taille, la couleur chaque animal. Une fois l'entraînement terminé, nous testerons le modèle en donnant le nouvel ensemble d'animaux. Le modèle identifiera l'animal et prédit le résultat à l'aide d'un algorithme approprié. Comme montre la figure 3 :

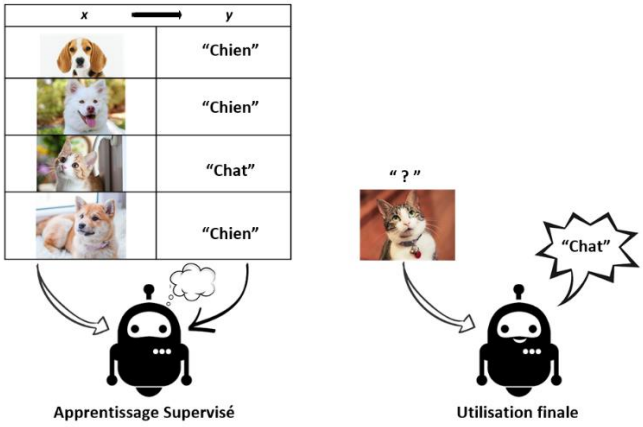


Figure 3 - Exemple sur l'apprentissage supervisé [10]

On parle d'apprentissage supervisé lorsque l'on fournit à une machine beaucoup d'exemples ($\boldsymbol{x}$, $\boldsymbol{y}$) dans le but de lui faire apprendre la relation qui relie $\boldsymbol{x}$ à $\boldsymbol{y}$.

Il existe deux types d'apprentissage supervisé qui sont :

2.3.1.a.1 Classification

Classer un objet dans différentes classes, c'est-à-dire que l'on cherche à prédire la valeur d'une variable discrète.

2.3.1.a.2 Régression

Prédire la valeur d'une variable continue, c'est-à-dire une variable qui peut prendre une infinité de valeurs.

2.3.1.b Apprentissage non supervisé

L'apprentissage dans cette méthode se fait à partir d'un échantillon d'apprentissage Dn = x1, . . ., Xn (les modèles sont déduits des données d'entrée non étiquetées), donc si seuls des exemples sans étiquette sont disponibles ou si les classes et leur nombre sont inconnus, on parle d'apprentissage non supervisé.

2.3.1.c Apprentissage par renforcement

C'est apprendre à agir par essai et erreur. Dans ce paradigme, un agent peut percevoir sont état et effectuer des actions. Après chaque action, une récompense numérique est donnée. Le but de l'agent est de maximiser la récompense totale qu'il reçoit au cours du temps.. Cette méthode étant mathématiquement plus avancée que les deux premières [11].

2.3.1.d Apprentissage semi-supervisé

Effectué de manière probabiliste ou non, il vise à faire apparaitre la distribution sous-jacente des « exemples » dans leur espace de description. Il est mis en œuvre quand des données (ou « étiquettes ») manquent... Le modèle doit utiliser des exemples non-étiquetés pouvant néanmoins renseigner.

Exemple : En médecine, il peut constituer une aide au diagnostic ou au choix des moyens les moins onéreux de tests de diagnostics [12].

2.3.2 Les algorithmes utilisés dans le domaine

- Les machines à vecteurs support
- Le boosting
- Les réseaux de neurones pour un apprentissage supervisé ou non-supervisé

- La méthode des k plus proches voisins pour un apprentissage supervisé
- Les arbres de décision
- Les méthodes statistiques comme par exemple le modèle de mixture gaussienne
- La régression logistique

Ces méthodes sont souvent combinées pour obtenir diverses variantes d'apprentissage. L'utilisation de tel ou tel algorithme dépend fortement de la tâche à résoudre (classification, estimation de valeurs, etc.) [12].

2.4 Réseaux de Neurones artificiels (Artificial Neural Network)

Le ANN est constitué d'un ensemble de couches successives dont chacune prend ses entrées sur les sorties de la précédent. Chaque couche est un ensemble de neurones n'ayant pas de connexion entre eux et qui reçoivent des informations numériques en provenance de neurones voisins. L'ensemble de couches est composé d'une couche d'entrée qui lit les valeurs d'entrées, une couche de sortie qui fournit les résultats du système et entre ces deux se cache une à plusieurs couches dites cachées qui participent au transfert.

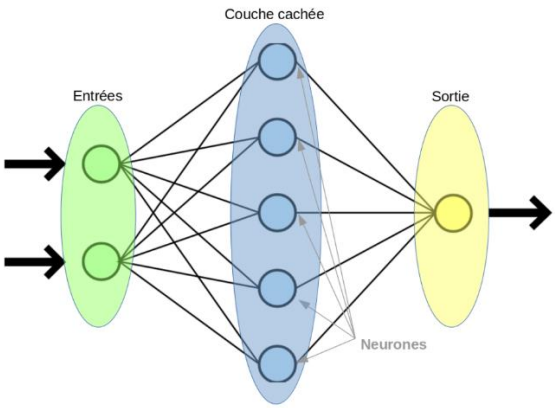


Figure 4 - Architecture de base d'un réseaude neurones artificiel [13]

2.5 Apprentissage Profond (Deep Learning)

2.5.1 Définition et origines

Le Deep Learning (DL) ou apprentissage profond est un type d'intelligence artificielle, dérivé du Machine Learning qui a été développé dans le but de créer des algorithmes capables

d’apprendre et de s'améliorer de manière autonome, contrairement à la programmation où la machine se contente d'exécuter à la lettre des règles prédéterminées [14].

L’apprentissage profond est l’une de nombreuses approches de l’apprentissage automatique(le Deep Learning est un sous domaine du Machine Learning).

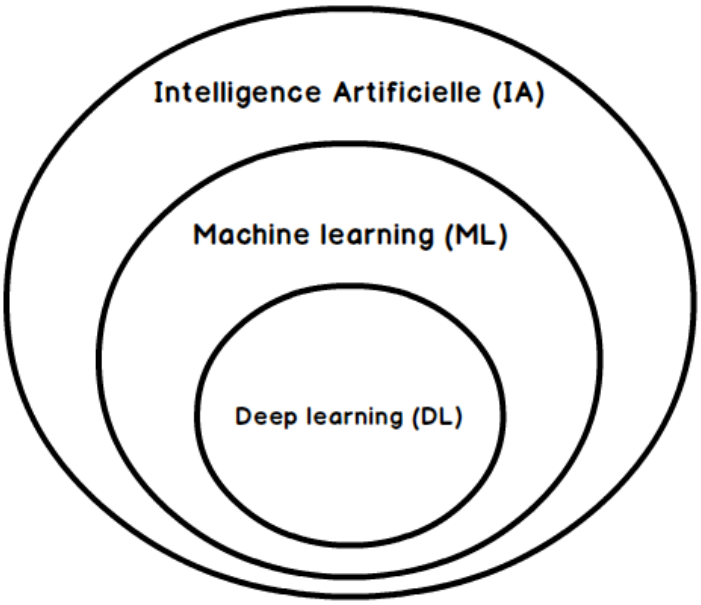


Figure 5 - Différence entre IA, ML et DL [15].

2.5.2 Réseaux de neurones et apprentissage profond

Les réseaux de neurones sont composés de nœuds interconnectés, ou neurones, chacun recevant un certain nombre d'entrées et fournissant une sortie. Chacun des nœuds dans les couches de sortie effectue un calcul de somme pondérée sur les valeurs qu'ils reçoivent des nœuds d'entrée, puis génère des sorties en utilisant de simples fonctions de transformation non linéaire sur ces sommations. Les corrections des pondérations sont effectuées en réponse aux erreurs ou aux pertes individuelles que les réseaux présentent aux nœuds de sortie. Ces corrections sont généralement effectuées dans les réseaux modernes en utilisant la descente de gradient stochastique, en considérant les dérivées des erreurs aux nœuds, une approche appelée rétro-propagation [16].

Les principaux facteurs qui distinguent les différents types de réseaux les uns des autres sont la manière dont les nœuds sont connectés et le nombre de couches Bien qu'il n'y ait pas de consensus clair sur ce qui définit exactement un DNN (réseau de neurones profonds), en général, les réseaux avec plusieurs couches cachées sont considérés comme profonds et ceux avec de nombreuses couches sont considérés comme très profonds [17].

2.5.2.a Application du Deep Learning

Le tableau suivant montre les différentes applications de l'apprentissage profond dans les domaines de la vie réel

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.

Le traitement automatique de langage naturel (NLP) : 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.

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.

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.

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 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.

Analyse des sentiments du texte : 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.

Recherche en marketing : 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.

2.6 Algorithmes de Classification

Dans cette partie nous présenterons les fameux algorithmes de classification.

2.6.1 Machine à vecteurs de support (SVM)

SVM est une méthode de classification binaire par apprentissage supervisé, elle fut introduite par Vapnik en 1995. Cette méthode est donc une alternative récente pour la classification. Cette méthode repose sur l'existence d'un classificateur linéaire dans un espace approprié. Puisque c'est un problème de classification à deux classes, cette méthode fait appel à un jeu de données d'apprentissage pour apprendre les paramètres du modèle. Elle est basée sur l'utilisation de fonctions dites noyau (kernel) qui permettent une séparation optimale des données. Le but de SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est un classificateur linéaire appelé hyperplan [19].

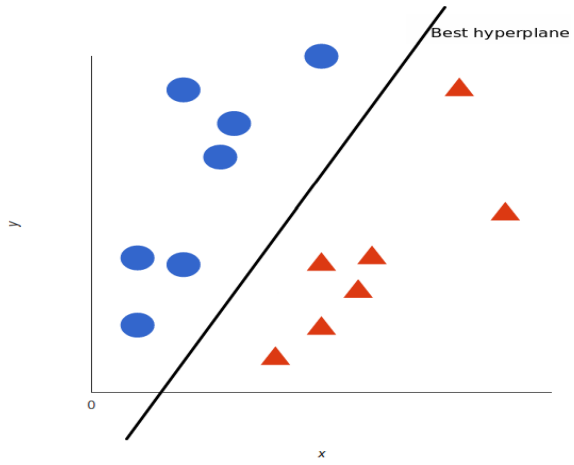


Figure 6 - Machine à vecteurs de support SVM.[20].

2.6.2 Régression logistique

C'est un algorithme de classification utilisé pour définir les frontières de décision et attribuer des observations à un ensemble discret de classes, elle transforme sa sortie à l'aide de la fonction sigmoïde, Cette fonction à la particularité d'être toujours comprise en 0 et 1.

Exemple :

Prédire si un email est un spam ($classe\ y = 1$) ou non ($classe\ y = 0$) selon le nombre de liens présent dans l'email (x)

2.6.3 Forets Aléatoires(RF)

Les forêts aléatoires est un algorithme d'apprentissage supervisé qui peut être utilisé pour la classification ou la régression aussi.

Les forêts aléatoires font partie des algorithmes qui restent efficaces lorsqu'ils sont appliqués à des grands jeux de données. La forêt sont composées de plusieurs arbres qui sont chacun construit avec une partie du jeu de données. La prédiction de la forêt est alors obtenue simplement en agrégeant les prédictions des arbres. Le fait que les forêts puissent être employées pour résoudre un grand nombre de problèmes d'apprentissage a fortement contribué à leur popularité. De plus, elles ne dépendent que d'un petit nombre de paramètres faciles à calibrer, les forêts sont également connues pour leur précision et leur capacité à traiter des jeux de données composés de peu d'observations et de nombreuses variables [21].

2.6.4 Arbres de décisions

Le principe de cet algorithme est la construction automatique d'une collection de règles mutuellement exclusive de type « si-alors» à partir de l'échantillon d'apprentissage. Ces règles sont structurées en arbre binaire et elles peuvent être facilement interprétées par un expert humain [22].

2.6.5 Bais Naïves

Les réseaux bayésiens sont un formalisme de raisonnement probabiliste de plus en plus utilisé en classification pour des problèmes de fouille de données, Un réseau bayésien B représente une distribution de probabilité sur X qui admet la loi jointe suivante : P(X1, X2, · · · , Xn) = ∏ P(Xi/XP a(Xi)) Cette décomposition de la loi jointe permet d'avoir des algorithmes d'inférence puissants qui font des réseaux bayésiens des outils de modélisation et de raisonnement très pratiques lorsque les situations sont incertaines ou les données incomplètes. Ils sont alors utiles en classification si les interactions entre les différents critères peuvent être modélisées par des relations de probabilités conditionnelles [23].

2.6.6 Auto-encodeur

Les auto-encodeurs (AE) sont des réseaux de neurones qui ont pour objectif de copier leurs entrées dans leurs sorties. Ils travaillent en comprimant l'entrée dans une représentation spatiale latente, puis en reconstruisant la sortie de cette représentation. Ce type de réseau est composé de trois parties (Voir Figure 7) : Encodeur, Milieu et Décodeur. Le milieu est une représentation compressée de l'entrée d'origine, créée par le codeur, qui peut être reconstruite par le décodeur.

- Encodeur : C'est la partie du réseau qui compresse l'entrée en une représentation en espace latent. Il peut être représenté par une fonction de codage h = f (x).
- Milieu (code) : C'est l'espace latent où l'on trouve la partie compressée des données d'entrée.
- Décodeur : Cette partie a pour objectif de reconstruire l'entrée de la représentation.

L'auto-encodeur dans son ensemble peut donc être décrit par la fonction g (f (x)) = y) où l'on veut que la sortie y soit aussi proche que l'entrée x initiale. Un auto-encodeur est un réseau de neurones capable de réaliser l'apprentissage non supervisé. Un réseau auto-encodeur tente

cependant de prédire x à partir de x, sans avoir besoin d'étiquettes. Ici, le défi consiste à recréer l'information de l'entrée originale à partir de données compressées, bruyantes ou corrompues.

L'idée derrière auto-encodeur est de construire un réseau avec une couche cachée étroite entre Encodeur et Décodeur, qui sert de représentation compressée des données d'entrée. De nos jours, le dé-bruitage des données et la réduction de la dimensionnalité pour la visualisation des données sont considérés comme deux principales applications pratiques intéressantes des auto-encodeurs [24].

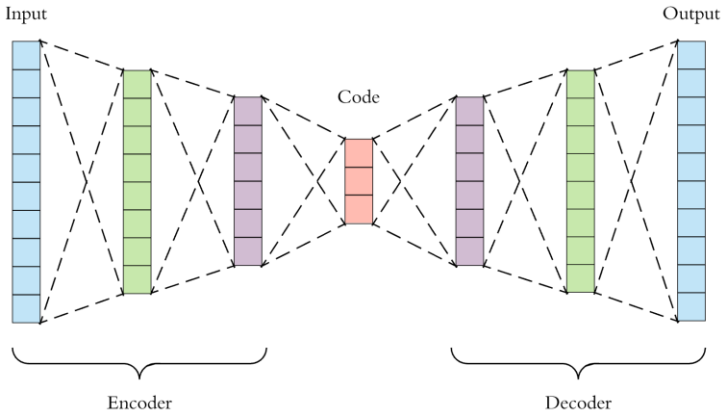


Figure 7 - Architecture Auto-encodeur.[25].

2.6.6.a Différence entre réseau de neurones et auto-encodeur

Habituellement, dans un réseau de neurones, on essaie de prédire un vecteur cible y à partir de vecteur d'entrée x. Dans un auto-encodeur, on essaie de prédire x à partir de x. Un réseau auto-encodeur est un type de réseau de neurones profonds dont l'objectif principal est d'extraire des caractéristiques qui aideront à reconstituer efficacement le signal d'entrée d'origine à partir de ces caractéristiques. Ainsi, un auto-encodeur possède un ensemble de couches cachées pour extraire successivement de telles caractéristiques à plusieurs niveaux et un autre ensemble de couches cachées qui suivent les couches précédentes et qui visent à reconstruire le signal d'entrée original. La formation d'un tel réseau a pour objectif que la sortie représente le plus fidèlement possible la contribution [24] .

2.6.7 Réseaux de neurone convolutif CNN

CNN est essentiellement un type de réseaux de neurones artificiels profonds à réaction qui est principalement utilisé dans les applications liées à la vision par ordinateur et au traitement

d’image. La puissante capacité d'apprentissage de Deep CNN est principalement due à l'utilisation de plusieurs étapes d'extraction de fonctionnalités qui peuvent automatiquement apprendre des représentations à partir des données. Pour cela, il est l'un des meilleurs algorithmes d'apprentissage pour comprendre le contenu des images. Ils consistent en un empilage multicouche de neurones, des fonctions mathématiques à plusieurs paramètres ajustables, qui prétraitent de petites quantités d’informations [26].

La figure 8 présente l’architecture du modèle CNN et ces différentes couches.

CNN est un type spécifique de réseaux de neurones qui sont généralement composés des couches suivantes :

2.6.7.a Couche convolutionnelle (CONV)

La couche convolutionnelle (en anglais convolution layer) utilise des filtres qui scannent l'entrée suivant ses dimensions en effectuant des opérations de convolution. La sortie de cette opération est appelée « Feature Map » ou aussi « Activation Map » [27].

2.6.7.b Pooling (POOL)

La couche de pooling (en anglais pooling layer) est une opération de sous-échantillonnage typiquement appliquée après une couche convolutionnelle. En particulier, les types de pooling les plus populaires sont le max et l'average pooling, où les valeurs maximales et moyennes sont prises, respectivement [27].

2.6.7.c Fully Connected (FC)

La couche de fully connected (en anglais fully connected layer) (FC) s'applique sur une entrée préalablement aplatie où chaque entrée est connectée à tous les neurones. Les couches de fully connected sont typiquement présentes à la fin des architectures de CNN et peuvent être utilisées pour optimiser des objectifs tels que les scores de classe [27].

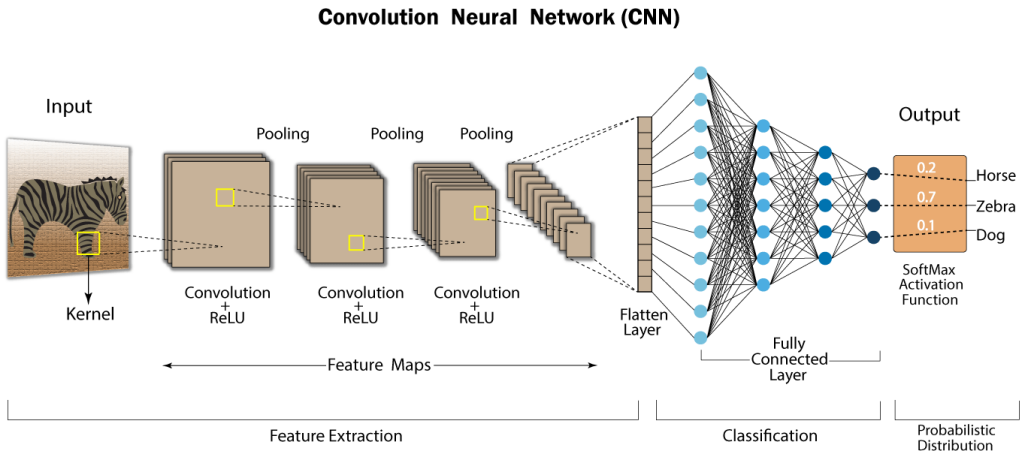


Figure 8 - Architecture du model CNN [28].

2.6.8 Réseaux de neurones récursifs (RNN)

Les réseaux de neurones récurrents est un type de réseau neuronal dans lequel la sortie de la couche précédente est alimentée en entrée de la couche en cours. Dans les réseaux de neurones traditionnels, toutes les entrées et sorties sont indépendantes les unes des autres, mais dans des cas comme lorsqu'il est nécessaire de prédire le mot suivant d'une phrase, les mots précédents sont nécessaires et il est donc nécessaire de se souvenir des mots précédents. C'est ainsi que RNN a vu le jour, ce qui a résolu ce problème à l'aide d'une couche cachée. Les RNN sont appelés récurrents car ils exécutent la même tâche pour chaque élément d'une séquence, la sortie étant dépendante des calculs précédents.

Les RNNs sont surtout utilisés dans les domaines du traitement automatique du langage naturel et de la reconnaissance vocale [29].

2.6.8 Mémoire à long/court terme (Long Short Term Memory) LSTM

Les RNN ont des problèmes de mémoire à court terme. Si une séquence est suffisamment longue, ils ont du mal à transporter les informations des étapes de temps antérieures aux étapes ultérieures. Par conséquent, cela entraîne le besoin d'une mémoire à long terme (LSTM) qui est un type spécial de RNN, capable d'apprendre les dépendances à long terme. Les LSTM ont les compétences nécessaires pour se souvenir des informations pendant de longues périodes. C'est parce que les LSTM contiennent leurs informations dans une mémoire, ce qui ressemble

beaucoup à la mémoire d'un ordinateur parce que le LSTM peut lire, écrire et supprimer des informations de sa mémoire.

2.6.9 Mémoire à long/court terme bidirectionnelle (Bidirectional Long Short Term Memory) BILSTM

Les BILSTM sont des LSTM bidirectionnelles. La différence entre elle, est que les BILSTM ne sont pas connectées juste aux étapes passés mais au futur aussi (le signal se propage aussi bien en arrière qu'en avant dans le temps).

2.7 Natural Langage processing(NLP)

Le traitement du langage naturel est un domaine de recherche et d'application qui explore comment les ordinateurs peuvent être utilisés pour comprendre et manipuler du texte ou de la parole en langage naturel pour faire des choses utiles [30].

Il est aussi définit comme une gamme théoriquement motivée de techniques de calcul pour analyser et représenter des textes naturels à un ou plusieurs niveaux d'analyse linguistique dans le but de réaliser un traitement du langage de type humain pour une gamme de tâches ou d'applications [31] .

2.8 Conclusion

Nous avons consacré ce chapitre à la présentation des notions de la machine Learning, Deep Learning, les algorithmes de classification ainsi qu'à l'utilisation de la machine Learning et du Deep Learning dans ce domaine. Nous avons également abordé la notion des CNN, RNN, LSTM, BILSTM .Nous explorerons dans le chapitre suivant certains travaux liées à la détection des hates speech dans les réseaux sociaux en détaillerons leurs modèles choisis et les algorithmes utilisés.

Chapitre 3 : Détection du Hate Speech - Etat de l’art

3.1 Introduction

On distingue que dans n'importe quel thème de recherche la partie de l'état de l'art est essentielle, d'où elle joue un rôle très important qui rends notre enquête plus approfondie et solide afin que les résultats seront basé sur diverses travaux avec des ressources fiables.

Le discours de haine est devenu un problème majeur qui est actuellement un sujet brûlant dans le domaine des médias sociaux. Par conséquent, nous avons rencontré des problèmes lors de la recherche des travaux existants car le domaine du hate speech est typiquement récent.

Dans ce chapitre, nous parlerons d'abord sur le domaine du hate speech passant par les problèmes dû à ce nouveau phénomène qui envahit les réseaux sociaux. Dans la deuxième partie de ce chapitre, nous partagerons les points de revus de littératures ainsi que les différentes approches et solutions proposées. A la fin nous débattrons les différents résultats obtenus dans les anciens travaux afin d'arriver à une conclusion qui nous aidera à implémenter et terminer les chapitres suivants.

3.2 Le discours de haine (Hate speech : HS)

Le discours de haine est un domaine de recherche actif dans la communauté sociologique. En particulier, certaines formes de discours de haine sont loin d'être résolues dans notre société, spécialement celles contre les Noirs, les immigrants et les femmes. Les discours de haine provenant de tels préjugés sont assez abondants et les autorités ont créé des politiques standards pour les contrer. Au fil du temps, cette tension a conduit à l'évolution des politiques standards pour réglementer le discours de haine pour que ça rentre dans les droits de l'homme .Malgré ça, ce problème ne cesse d'augmenter surtout avec l'évolution de la technologie et l'apparition des réseaux sociaux.

Le hate speech se produit lorsque les individus sont en désaccord et chacun cherche à exprimer et imposer son point de vue sur une variété de sujets. Le hate speech est décrit comme un acte «menaçant» qui se produit lors d'interactions interpersonnelles.

Nous allons présenter les plus importantes définitions du hate speech provenant d'une variété de sources.

D'après [32], le hate speech se situe à l'intersection de multiples tensions en tant qu'expression de conflits entre différents groupes au sein et entre les sociétés, est un phénomène qui peut facilement proliférer sur les réseaux sociaux.

Selon [33,34], Le discours de haine est généralement défini comme toute communication qui dénigre une personne ou un groupe sur la base de certaines caractéristiques telles que la race, la couleur, l'appartenance ethnique, le sexe, l'orientation sexuelle, la nationalité, la religion ou d'autres caractéristiques.

Il y a une énorme différence entre la liberté d'expression et le discours de haine, mais nous pouvons considérer le discours de haine comme un type de liberté d'expression. La liberté d'expression aide la communauté ou les individus à exprimer leurs idées et opinions sans crainte de sanction. La liberté d'expression est requise pour les droits démocratiques et elle garantit la jouissance autonome de l'individu. Contrairement, le discours de haine viole les droits fondamentaux de la communauté et des individus et réduit les limites autorisées de la liberté d'expression [35].

Le discours de haine est largement répandu sur les sites de réseaux sociaux en ligne (OSN), tels que Twitter, Facebook et Instagram, car ils offrent un espace ouvert aux utilisateurs pour exprimer leurs opinions, leurs croyances et leurs idées sans aucune limitation. De plus, les réseaux sociaux rassemblent des utilisateurs de différents pays et nationalités, chacun d'eux a des antécédents, des coutumes et des traditions différentes.

Le HS se présente sous de nombreuses formes différentes comme le discours de haine sur les femmes, les religions, les immigrants et la race etc., à cet effet ce phénomène peut causer d'immenses dégâts humaines, psychologiques et sociales.

3.3 Identification du hate speech dans les réseaux sociaux

Les plateformes de réseaux sociaux fournissent un moyen de communication peu coûteux qui permet à quiconque d'atteindre rapidement des millions d'utilisateurs, Par conséquent, sur ces

plateformes, n'importe qui peut publier du contenu et toute personne intéressée par le contenu peut l'obtenir, ce qui représente une révolution transformatrice dans notre société. Néanmoins, ce même potentiel des systèmes de médias sociaux soulève un défi important: ces systèmes offrent un espace pour des discours qui sont nuisibles à certains groupes de personnes. Ce défi se manifeste par un certain nombre de variantes, y compris l'intimidation, le contenu offensant et le discours de haine. Plus précisément, les autorités de nombreux pays reconnaissent aujourd'hui rapidement le discours de haine comme un problème sérieux, notamment parce qu'il est difficile de créer des barrières sur Internet pour empêcher la diffusion de la haine à travers les pays ou les minorités.

Les réseaux sociaux comme Facebook, Twitter et YouTube ont utilisé différentes politiques pour gérer les discours de haine. Selon [36], le problème de la détection des personnes haineux reste toujours un immense problème surtout qu'il n'existe aucune solution optimale actuelle qui pourra arrêter l'augmentation de ce phénomène malgré les différentes politiques déjà appliquées.

3.4 Travaux connexes

La détection du langage agressif et son rôle dans l'analyse de sentiments a fait l'objet de plusieurs campagnes d'évaluation ces dernières années, telles que

- ✓ la campagne SemEval[4] 2015 (International Workshop on Semantic Evaluation) Task 11 (Ghosh et al. 2015) sur des tweets en anglais.
- ✓ les campagnes SENTIPOLC@Evalita [5](SENTment POLarity Classification) dans leurs éditions de 2014 et 2016 sur des tweets en italien (Basile et al. 2014; Barbieri et al. 2016).
- ✓ DEFT [6](Défi Fouille de Textes) tweet en français.
- ✓ CLEF 2021[7](Conference and labs of the evaluation forum) cette année sur les tweets anglais et espagnols.

Les premières publications sur la classification des courriers électroniques basées sur le Machine Learning remontent à 2012. William Warner et Julia Hirschberg [37] se sont intéressés à

[4] https://semeval.github.io/
[5] http://www.di.unito.it/~tutreeb/sentipolc-evalita16/
[6] https://deft.limsi.fr/2018/
[7] http://clef2021.clef-initiative.eu/

la détection des discours de haine sur le World Wide Web et ont proposé une détection de la parole par apprentissage de tous les discours de haine. Sur la base de la signification des mots, les annotateurs peuvent utiliser la probabilité pour marquer un paragraphe, qu'ils appellent logodds après avoir capturé les paragraphes qui correspondent aux expressions régulières générales des mots liés au judaïsme et à Israël. Cela a donné lieu à environ 9 000 paragraphes. Parmi eux, ils ont rejeté ceux qui ne contenaient pas de phrases complètes. Ensuite, ils ont identifié sept catégories, à savoir l'antisémitisme, l'anti-noir, l'anti-asiatique, l'anti-féminin, l'antimusulman, l'anti-immigration ou toute autre haine.

Les auteurs de [38] ont utilisé une approche d'apprentissage automatique supervisé avec N-gramme comme fonctionnalités pour détecter les tweets racistes contre les Noirs. Ils ont classé chaque tweet comme «raciste» et «non raciste». La méthodologie du questionnaire a été utilisée dans cette recherche pour mesurer la complexité de la façon dont les gens peuvent identifier le discours de haine. Ils ont recueilli des centaines de tweets contenant des mots-clés haineux et ont demandé à trois étudiants du même sexe et du même âge mais de races différentes de classer si le tweet était offensant ou non. Le résultat a montré une concordance globale de 33%, ce qui est plus difficile pour les machines à faire avec précision. Ils ont fait la distinction entre les tweets racistes et non racistes en utilisant le classificateur Naïve Bayes, qui a obtenu une précision moyenne de 76% pour les tweets individuels et un taux d'erreur moyen de 24%.

L'approche de [39] utilise une classification basée sur les textes avec l'apprentissage automatique supervisé pour faire la distinction entre les réponses antagonistes[8] et les discours haineux, y compris la religion, l'ethnicité, la race et des réponses plus générales. Ils se basent sur l'utilisation de N-gram, Ainsi que d'autres méthodes de classification (Random Forest, Decision Tree, SVM). Ils ont obtenu une mesure « F-score » globale de 0,95.

Dans l'approche de [41], ils ont construit un dictionnaire pour détecter le racisme dans les commentaires des médias sociaux néerlandais. Ils ont créé trois dictionnaires de discours pour classer le texte en catégorie raciste ou non raciste. Le premier dictionnaire a été créé pour récupérer des termes plus neutres et peut-être des termes racistes, tandis que le second dictionnaire a été créé par expansion automatique basée sur l'utilisation d'un modèle word2vec

[8] État d'opposition entre des personnes, des nations, des classes sociales, des doctrines

formé sur un large corpus de texte général en néerlandais. Le troisième dictionnaire a été créé sur la base du filtrage manuel des extensions incorrectes. Le but de cette étude était de classer le texte en catégorie raciste ou non raciste. Ils dépendaient principalement de l'utilisation de SVM comme classificateur d'apprentissage automatique supervisé.

Le travail de [42] a mené une étude sur la détection des discours haineux contre les immigrants et les femmes spécialement. Ils ont proposé une tâche qui s'articule autour de deux sous-tâches liées. La sous-tâche A est une tâche de classification à deux classes (ou binaire) où le système doit prédire si un tweet en anglais ou en espagnol avec une cible donnée (femmes ou immigrés). Ensuite, dans la sous-tâche B, les systèmes sont invités à classer les tweets haineux (par exemple, les tweets où HS contre nos cibles a été identifié) en fonction à la fois de l'attitude agressive et de la cible harcelée. Ils ont participé avec ce model dans SemEval 2019.

Les chercheurs de [43] ont abordé le sujet de la détection automatisée des discours haineux et problème du langage offensant en prenant les méthodes de classification automatisées comme approche en utilisant d'abord une régression logistique avec régularisation L1 pour réduire la dimensionnalité des données. Ils ont testé ensuite une variété de modèles qui ont été utilisés dans des travaux antérieurs: régression logistique, bayes naïve, arbres de décision, forêts aléatoires et SVM linéaires. Ils ont aussi testé chaque modèle, ils ont découvert que la régression logistique et la SVM linéaire avaient tendance à être nettement meilleures que les autres modèles. Ils ont décidé d'utiliser une régression logistique pour le modèle final car elle les permet plus facilement d'examiner les probabilités prédites. Ils ont obtenus en final une précision globale de 0,91, un rappel de 0,90 et un score F1 de 0,90.

Dans le travail de [44], les auteurs ont proposé un modèle de traitement profond du langage naturel en combinant les deux algorithmes CNN et RNN pour la détection automatique du discours de haine dans les données des médias sociaux en utilisant l'ensembles des données HASOC2019[9] (Hate Speech and Offensive Content Identification in Indo-European Languages). Ils ont obtenu un F1-score de 0,63 dans la détection de discours de haine sur l'ensemble de test de HASOC.

[9] https://hasocfire.github.io/hasoc/2019/index.html

Le travail de [45] traite l'apprentissage par transfert afin de détecter les discours haineux dans les réseaux sociaux, en utilisant pour la partie de vectorisation (extraction des features) le Word Embeddings afin d'avoir une représentation significative pour les mots similaire. Plus spécifiquement, ils ont formé et testé une architecture de réseau neuronal profond (LSTM ET B-ILSTM) sans et avec apprentissage par transfert sur le total de 37, 520 Tweets en Anglais. Leur méthode est capable de créer des incorporations de mots et de phrases qui sont spécifiques à ces tâches de détection du racisme, du sexisme, de la haine et des offensives, tout en exploitant plusieurs ensembles de données plus petits et non liés pour intégrer le sens du discours de haine générique. Son exactitude de classification de F1-Score est de 72% à 78%.

Quatre différents modèles : CNN, GRU [10](Gated recurrent unit), CNN + GRU et BERT (Bidirectional Encoder Representations from Transformers) ont été évalué et comparé dans le papier de [46]. Les résultats obtenus de leurs expériences montrent que CNN a surpassé avec succès les autres modèles, avec un score F1 de 0,79. Ils ont également montré que le BERT n'a pas réussi à améliorer les résultats par rapport aux autres modèles évalués.

3.5 Discussion

Le tableau suivant présente un résumé de tous les travaux cités ci-dessus, ils sont organisés selon leurs séries chronologiques respectives, Toutes les approches et leurs résultats d'expériences avec les métriques: Précision (P), Rappel (R) , F1-Score (F), sont répertoriés de manière concise, sans oublié de mentionner que e lors de notre revue de la littérature nous avons remarqué que le sujet en lui-même est récent et on ne trouve pas beaucoup de travaux sur le hate speech dans les réseaux sociaux, ce qui a rendu notre enquête plus difficile .

[10] GRU est comme une longue mémoire à court terme avec une porte d'oubli, mais a moins de paramètres que LSTM

Année	But de l'approche	Langue	Représentation	Plateforme	Algorithme	Mesure d'evaluation		
						P	R	F1
[37] 2012	Detecting Hate Speech on the World Wide Web	-	Template-Based Strategy	Word Wide Web	SVM	0 ,68	0,6	0,63
[38] 2013	Detection des hate speech contre les noirs dans Twitter	-	N-gramme	Twitter	Naïve Bayes	0,76	-	-
[39] 2014	Detection, classification et statistique sur le hate speech dans le réseau Twitter	Anglais	Sac de mots(BOW)	Twitter	Decision tree, SVM , Random Forest	0,89	0,69	0,95
[40] 2014	Web content classification pour la détection des discours de haines	Anglais	TF-IDF et N-gramme	Twitter	Naïve Bayes	0,97	0,82	-
[41] 2016	Une approche basée sur un dictionnaire pour la détection du racisme dans les médias sociaux néerlandais	Anglais	Dictionnaire Word2vec	Social Media	SVM	-	-	0,46
[43] 2017	Détection automatisée des discours haineux et problème du langage offensant	-	TF-IDF et N-gramme	Twitter	Régression logistique	0,91	0,90	0,90
[42] 2019	Détection multilingue des discours de haine contre Immigrants et femmes sur Twitter	Espagnol, Anglais	Approche par mot clé	Twitter	SVM, CNN, LSTM	-	-	-

[45] 2019	Apprentissage par transfert pour la détection des discours haineux dans les médias sociaux	Anglais	Word Embeddings	Twitter	B-LSTM LSTM	-	-	0,72
[46] 2020	apprentissage en profondeur pour la détection automatique des discours de haine dans la Twittersphere saoudienne	Arabe	Approche basée sur les mots clés et les threads	Twitter	CNN,GRU, CNN+GRU BERT	-	-	0,79
[44] 2021	Défis pour la détection des discours haineux	Anglais	Word Embeddings	Réseaux sociaux	CNN+LSTM	-	-	0 ,63

Tableau 1 - comparaison entre les differents literrature recentes

3.6 Conclusion

Dans ce chapitre nous avons commencé par introduire l'hate speech, passant par son utilisation dans le monde des réseaux sociaux, nous avons fini par citer quelques approches récentes dans ce domaine avec une comparaison entre eux.

Dans le chapitre suivant nous allons voir notre solution proposée pour ce problème ainsi que la conception et les différents algorithmes utilisés dans notre travail.

Chapitre 4 : Conception d’une méthode de détection du hate speech à partir des réseaux sociaux

4.1 Introduction

Après avoir parcouru les différents travaux réalisés dans le domaine de la détection des hates speechs, maintenant c'est le moment de présenter l'architecture et la conception de notre nouvelle approche. Notre approche est composée d'une partie de chargement et prétraitement de données après la vectorisation de ces données terminant par l'apprentissage et le test du modèle. Dans ce chapitre, nous allons présenter toutes les étapes en détail et comment nous avons combiné ces les différents algorithmes de la machine et Deep Learning pour obtenir notre nouvelle solution.

4.2 Architecture de notre approche

Notre architecture est composée de 4 étapes essentielles (Voir Figure 9) :

- Etape 1 : Chargement des données.
- Etape 2 : Prétraitement.
- Etape 3 : vectorisation des données.
- Etape 4 : Apprentissage des modèles.

Lors de notre exploitation des autres travaux similaires à notre domaine nous avons constaté que la résolution de ce problème été faite par l'utilisation de différents algorithmes, parmi ces algorithmes CNN, RNN, LSTM, mais selon notre revue littérature et nos recherches aucune des autres approches ont utilisé l'algorithme auto-encodeur AE qui est utilisé généralement pour la classification des images , ce qui nous rend les premiers à utiliser cet algorithme dans le domaine du NLP ainsi que dans la détection des hates speechs en combinant cet algorithme avec un algorithme du ML qui est le Random Forest.

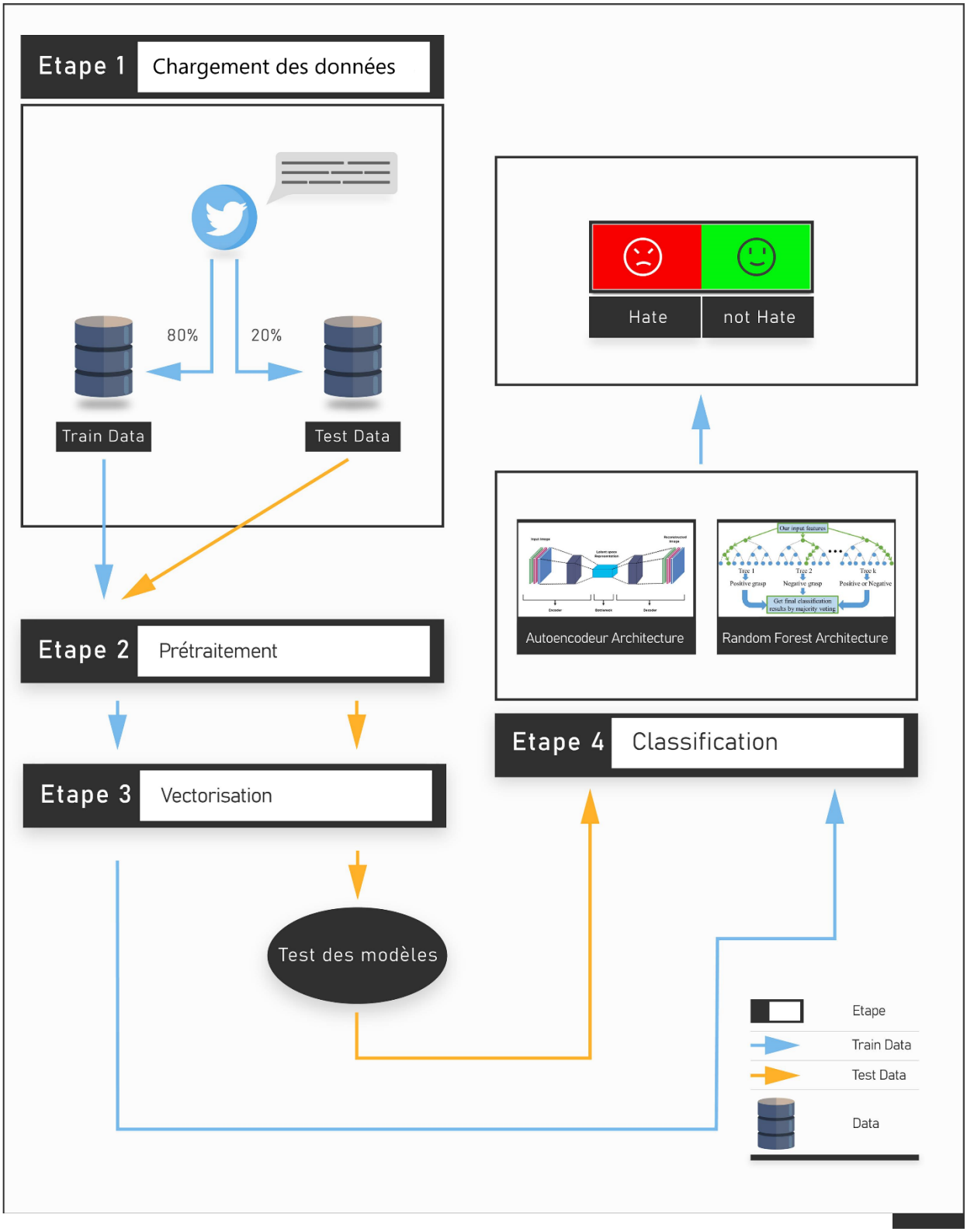


Figure 9 - L'architecture de notre modèle

4.3 Chargement des données

L'extraction des données consiste à convertir les données brutes en données préparées et ne pourra jamais être réalisée sans avoir un dataset contenant des données d'entrée. Nous avons utilisé des données d'apprentissage et des données de tests.

Les données d'apprentissage (training data) : les données d'apprentissage sont des données étiquetées utilisées pour entraîner nos modèles. Nous avons un ensemble de tweets où nous connaissons préalablement est ce que ce sont des hate speechs ou non. Les données d'apprentissage permettent la construction du modèle pour qu'il soit utilisé après par les données de test.

Les données de test (test data) : en utilisant le modèle préalablement appris, nous devons détecter les hate speechs à partir des données de test. Par la suite, il faut évaluer les performances du modèle à l'aide d'une ou plusieurs métriques de performance.

Notre but est de prédire les messages qui contiennent des mots de discours de haine ou pas. Comme mentionné précédemment, nous sommes intéressés par les messages du réseau social Twitter avec seulement les tweets écrits en Anglais donc nos données d'entrée sont des tweets (texte court).

4.4 Prétraitement des données

Le texte brut contient beaucoup d'aléatoire qui nuit à l'estimation des modèles : les accents, les minuscules, les majuscules, les signes de ponctuation, etc... On peut les garder mais plus de variabilité implique plus de données pour les apprendre. On préfère alors de le nettoyer avant de le découper en mots (ou caractères ou syllabes).

Les étapes que nous avons utilisées dans cette phase sont les suivantes :

- Suppression des URL
- Passage en minuscule
- Suppression des nombres et des chiffres
- Suppression de la ponctuation
- Sppression des imojis

- Elimination des « StopWords » : Ce sont les mots très courants dans la langue étudiée ("et", "à", "le"... en français / "a», "about","am","an","and" ...en anglais) qui **n'apportent pas de valeur informative** pour la compréhension du "sens" d'un document. Ils sont très fréquents et ralentissent notre travail : nous souhaitons donc les supprimer.

Tweet avec Stop words	Tweet sans Stop words
explanation\r\nwhy the edits made under my username hardcore metal	explanation edits made username hardcore metal
d'aww! he matches this background colour i'm seemingly.	d'aww! matches background colour i'm seemingly.

Figure 10 - Exemple sur le prétraitement des « Stop words ».

- La tokenisation **:** la tokenisation est un moyen de séparer un morceau de texte en unités plus petites appelées jetons. Ici, les jetons peuvent être des mots, des caractères ou des sous mots.

```
t[22]: [['explanation',
         'edits',
         'made',
         'username',
         'hardcore',
         'metallica',
         'fan',
         'reverted',
         'vandalisms',
         'closure',
         'gas',
         'voted',
         'new',
         'york',
         'dolls',
         'fac',
         'please',
         'remove',
         'template',
         'talk',
```

explanation edits made username hardcore metal

'explanation ' , 'edits ' , 'made ' , ' username' , 'hardcore ' , 'metal '

Figure 11 - Exemple sur l'étape de tokenisation.

```
Function Preparing _Data (dataset):
Begin
D = dataset ;
for each T ∈ D
    T = set_to_lower(T) ;              // Mettre le text en miniscule
    T = clean_links(T) ;               // Supprimer tous les liens dans un texte
    T = remove_stopwords(T) ;          // Eliminer les « stopwords »
    T = remove_punctuation(T) ;        // Supprimer la ponctuation
    T = remove_bad_chars(T) ;          // Suppression des caracteres indésirable au debut comme « RT »
    T = remove_numbers(T) ;            // Eliminer les nombres
    D[T] = T ;
end for ;
Adjust_targets(D) ;
End.

Function Adjust_targets (dataset):
Begin
D = dataset ;
for each label ∈ D
   if label > 1 then
      D[label] = 1 ;
end for ;
End.
```

Figure 12 - Le pseudo code de la préparation des données

4.5 Vectorisation

La phase de vectorisation est un processus du langage naturel qui utilise des modèles de langages pour mapper des mots à un espace vectoriel. Cet espace vectoriel représente chaque mot par un vecteur de nombres réels. Parmi les principaux algorithmes utilisés dans la vectorisation on site :

- **Word Embeddings :** est un type de représentation de mots qui permet aux mots ayant une signification similaire d'avoir une représentation similaire. Plus récemment, de nouvelles techniques basées sur des modèles probabilistes et des réseaux de neurones, comme Word2Vec, ont permis d'obtenir de meilleures performances. Dans notre architecture on s'intéresse au Word2vec.

Word2vec : le word2vec est basé sur le principe que les vecteurs sont appris en comprenant le contexte dans lequel les mots apparaissent. Le résultat est des vecteurs dans lesquels des mots ayant des significations similaires se retrouvent avec une représentation numérique similaire [47].

Voir la figure 13 qui represente des vecteurs de mots et la distance de chaque deux mots qui ont des significations similaires

Dans cette phase nous allons utiliser word2vec afin de générer in vecteur de mots appeler v1

Cet algorithme permet d'améliorer considérablement les modèles d'apprentissage automatique qui utilisent du texte comme entrée.

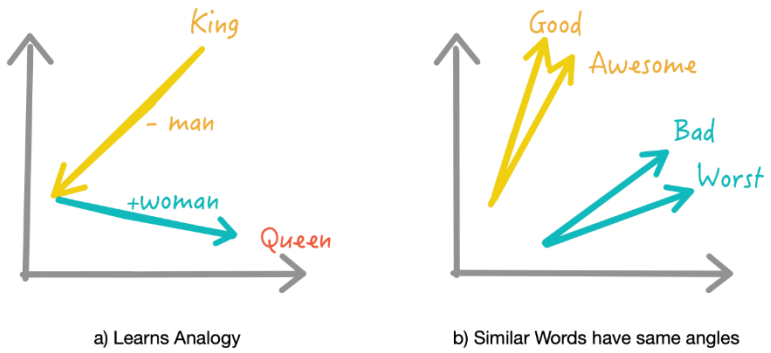


Figure 13 - Exemple sur la representation des mots similaire avec word2vec[48]

Word2vec possède deux architectures neuronales appelées CBOW et Skip-Gram :

CBOW est un modèle d'apprentissage dans lequel le réseau de neurones essaie de prédire un mot dans un contexte.

Skip-Gram est un modèle d'apprentissage dans lequel le réseau de neurones essaie de prédire un contexte sur la base d'un mot donné.

Notre Solution utilise le Word2vec google qui est basé sur l'architecture Skip-gram d'aprés plusieurs chercheurs.

La figure suivante montre les deux architectures CBOW et SKIP-Gram :

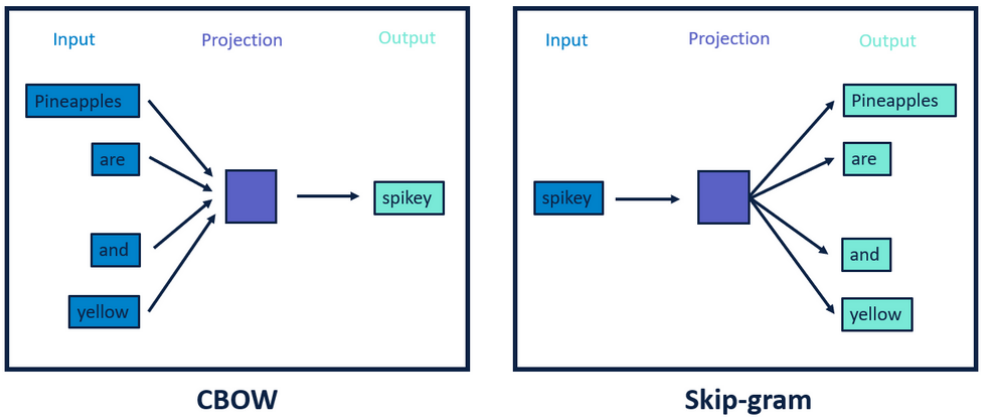


Figure 14 - Exemple comparatif des architectures CBOW et Skip-gramme [49]

4.6 Apprentissage des modèles

4.6.1 Modèle Random Forest

Nous avons choisis l'utilisation de l'algorithme Random Forest car c'est est un algorithme d'apprentissage automatique populaire qui appartient à la technique d'apprentissage supervisé et qui est utilisé pour les données dans un but de classification et aussi de régression. Il est basé sur le concept d'apprentissage d'ensemble, qui est un processus consistant à combiner plusieurs classificateurs pour résoudre un problème complexe et améliorer les performances du modèle. Il

contient un certain nombre d'arbres de décision sur divers sous-ensembles de l'ensemble de données donné et on choisit la catégorie la plus fréquente de cet ensemble de donnée.

Le schéma ci-dessus explique le fonctionnement de notre modèle Random Forest, où nous avons en entrée nos features sous forme d'un vecteur de mots qui vont être par la suite partagé sur un ensemble d'arbres et chaque arbre prend une décision : si la donnée est un hate speech ou pas, à la fin la décision va être choisis par rapport au vote majoritaire.

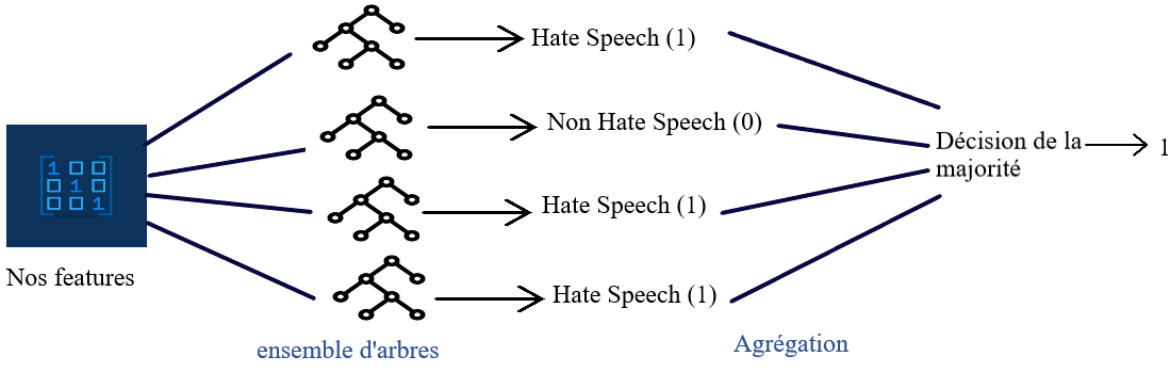


Figure 15 - Architecture de notre modèle Random Forest

4.6.2 Modèle « Autoencodeur_RandomForest »

Les auto-encodeurs sont des algorithmes d'apprentissage à base des réseaux de neurones artificiels. Un auto-encodeur est un modèle qui prend en entrée en TAL (Traitement automatique de la langue) un vecteur de mots et encode une représentation cachée qui va être utilisée comme features dans le nouvel algorithme de classification.

L'auto-encodeur est composé de deux parties : l'encodeur et le décodeur. L'encodeur est constitué d'un ensemble de couches de neurones, qui traitent les données afin de construire de nouvelles représentations dites "encodées". À leur tour, les couches de neurones du décodeur, reçoivent ces représentations et les traitent afin d'essayer de reconstruire les données de départ. Les différences entre les données reconstruites et les données initiales permettent de mesurer l'erreur commise par l'auto-encodeur.

La plupart du temps, on ne s'intéresse pas à la dernière couche du décodeur, qui contient uniquement la reconstruction des données initiales, mais plutôt à la nouvelle représentation créée par l'encodeur, généralement l'auto-encodeur est utilisée pour l'extraction des caractéristiques et la classification des images mais récemment beaucoup de recherches dans la TAL se basent sur l'auto-encodeur qui prend un vecteur de mots en entrée pour avoir des résultats performants. Cette technique a créé une nouvelle révolution dans le domaine de TAL. Les performances de certains algorithmes d'apprentissage automatique pourraient être encore améliorées grâce à l'utilisation de ces AE.

Afin d'arriver à des résultats qui montrent la possibilité de l'utilisation de l'algorithme Auto-Encodeur dans le TAL et aussi les hauts résultats de Random Forest dans le NLP, nous avons combiné ces deux algorithmes.

Dans cette partie après encodage nous allons générer un vecteur de mots appelé V2.

La figure suivante représente l'architecture de notre modèle autoencodeur_randomforest.

Dans l'implementation de notre solution nous avons utilisé 50 couches pour encoder notre vecteur de mots, le choix de nombre de couches a été par rapport au meilleur résultat .

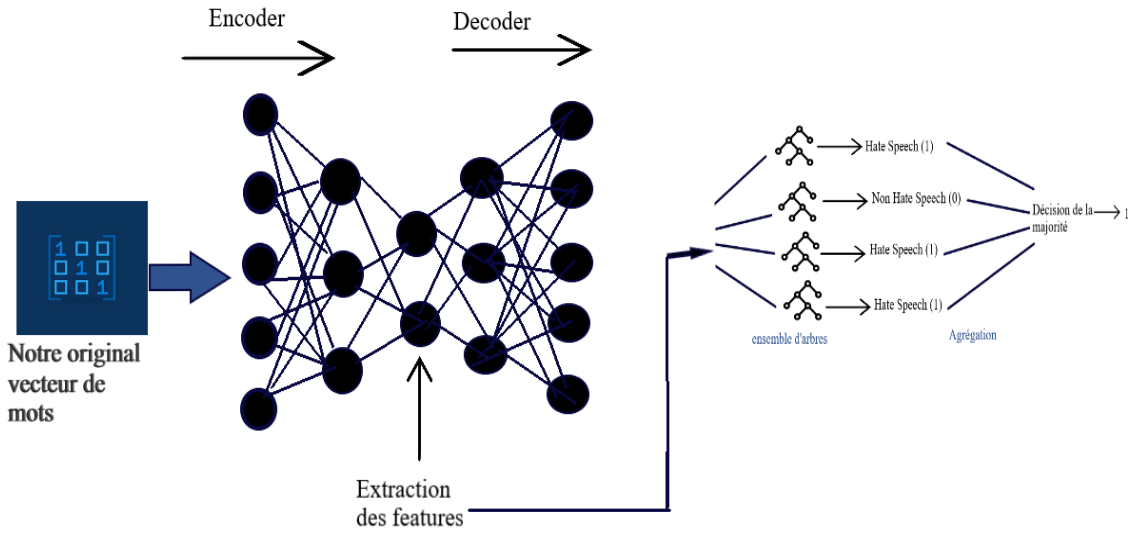


Figure 16 - Architecture de notre modèle Autoencodeurt-Randomforest

4.7 Conclusion

Dans ce chapitre, nous avons présenté profondément les différentes phases de notre approche proposée afin de mieux comprendre les bases et l’architecture choisie. Dans le chapitre suivant, nous allons décrire le processus que nous avons suivi pour mettre en œuvre ce modèle et les résultats que nous avons obtenus pour pouvoir les discuter et les comparer avec les travaux du même domaine à la fin.

Chapitre 5 : Expérimentations et résultats

5.1 Introduction

Afin de détecter le hate speech dans les tweets, un ensemble d'outils et de matériels est utilisé pour réaliser notre modèle. Dans ce chapitre, nous allons présenter l'ensemble de matériels utilisés, les différents résultats de notre approche proposée et enfin démontrer leur efficacité en les comparant avec les autres travaux déjà vus dans ce domaine.

5.2 Matériel utilisé

Le matériel utilisé est vraiment important lors de l'implémentation de la solution et surtout dans le traitement des données, qui dépend vivement sur la performance du matériel et aussi pour l'évaluation des modèles basés sur les réseaux de neurones. Dans notre travail, le matériel utilisé pour l'implémentation et le test est :

- **Processeur :** Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz.
- **Mémoire RAM installée :** 16,0 Go.
- **Type du système :** Système d'exploitation 64 bits, Windows 10 professionnel.

5.3 Outils utilisés

Dans le but d'implémenter notre solution, nous avons choisi le langage **Python version 3.8** car c'est :

- Un langage de programmation interprété [11]
- Le plus approprié au domaine de notre travail
- Extensible
- Facile, simple à utiliser et compréhensible.
- Extensible et intégrable.
- Ces diverses bibliothèques étendues et qui sont dédiées au traitement de données et au Deep Learning.

[11] Interprété : il ne nécessite pas d'être compilé pour fonctionner et le même code source pourra marcher directement sur tout ordinateur.

De plus, nous avons travaillé avec l'environnement **Jupyter**[12] qui est une application web open source dérivé du projet Ipython. **Cet environnement permet** d'éditer des visualisations, de les partager et de permettre des modifications interactives du code et il permet aussi d'apprendre facilement à analyser et à manipuler de gros volumes de données.

5.4 Bibliothèques utilisées

- **Keras :** keras [13] est une bibliothèque open source qui permet la constitution rapide de réseaux neuronaux, elle est connue comme une API de réseaux de neurones de haut niveau.
- **Pandas** : Pandas [14] est une bibliothèque qui permet l'analyse et la manipulation des données en particulier des structures de données et des opérations de manipulation de tableaux numériques.
- **NumPy :** NumPy [15]est une bibliothèque open source destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.
- **Scikit-learn :** Scikit-learn [16] est une bibliothèque libre destinée à l'apprentissage automatique, Elle propose de nombreuses bibliothèques d'algorithmes à implémenter, clé en main, Elle comprend notamment des fonctions pour estimer des forêts aléatoires, des régressions logistiques, des algorithmes de classification..
- **NLTK :** (NLTK)[17] est une bibliothèque permettant un traitement automatique des langues, elle est parmi les bibliothèques NLP les plus puissantes, qui contient des paquets permettant aux machines de comprendre le langage humain et d'y répondre de manière appropriée.
- **Gensim :** Gensim[18] est une bibliothèque Python open source gratuite permettant de représenter des documents sous forme de vecteurs sémantiques.

[12] https://jupyter.org/
[13] https://keras.io/
[14] https://pandas.pydata.org/
[15] https://numpy.org/
[16] https://scikit-learn.org/stable/
[17] https://www.nltk.org/
[18] https://radimrehurek.com/gensim/

- **Seaborn :** Seaborn[19] est une bibliothèque de visualisation qui fournit une interface de haut niveau pour dessiner des graphes statistiques.

La figure 16 montre les différentes bibliothèques que nous avons chargées

```
Entrée [1]: import pandas as pd # importing pandas
            from nltk.corpus import stopwords
            import nltk
            import seaborn as sns
            sns.countplot('hate_speech',data=hate_df)
            import numpy as np
            from sklearn.model_selection import train_test_split
            import multiprocessing
            from gensim.test.utils import common_texts
            from gensim.models import Word2Vec
            from sklearn.preprocessing import MinMaxScaler
            from sklearn.model_selection import train_test_split
            from tensorflow.keras.models import Model
            from tensorflow.keras.layers import Input
            from tensorflow.keras.layers import Dense
            from tensorflow.keras.layers import LeakyReLU
            from tensorflow.keras.layers import BatchNormalization
            from sklearn.ensemble import RandomForestClassifier
```

Figure 17 - Chargement des librairies nécessaires pour l'implémentation

5.5 Dataset

Dans cette phase nous allons parler sur l'ensemble de données que nous avons utilisé lors de notre travail. Le Premier Dataset est téléchargé à partir du site Kaggle[20]. Il est constitué de 159571 tweets, organisés selon le contenu du tweet : si il est toxique (hate speech) ou pas. A partir de ce dataset nous avons utilisé 80% pour le training et 20% pour le test.

[19] https://seaborn.pydata.org/
[20] https://www.kaggle.com/mrinaal007/hate-speech-detection

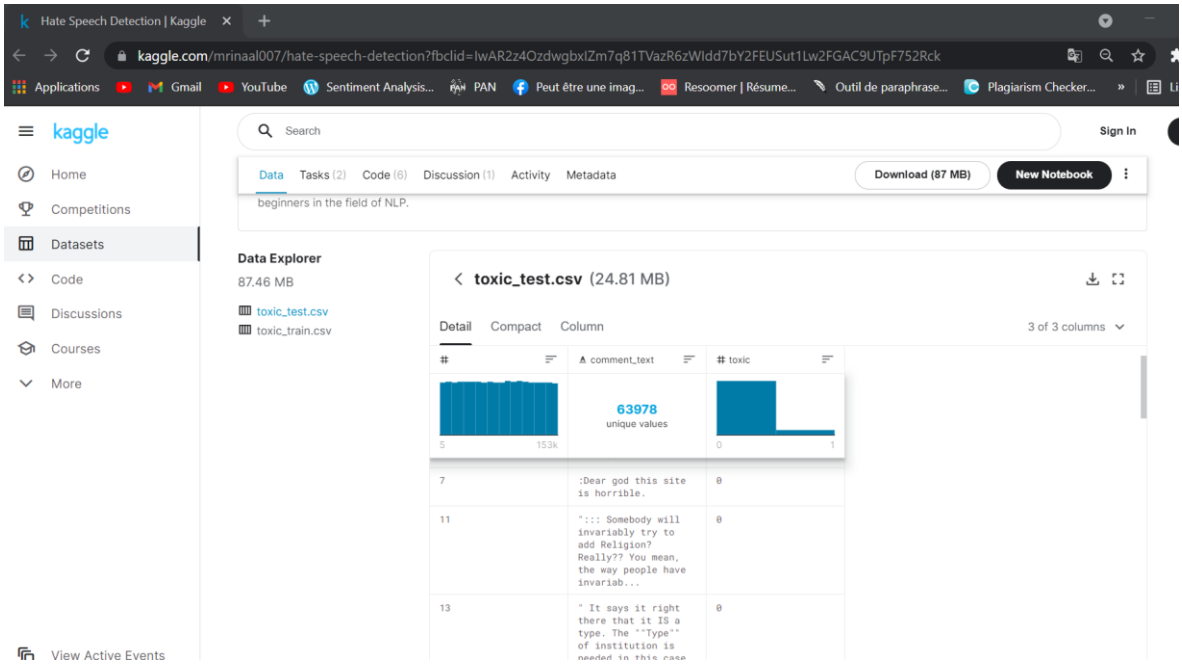


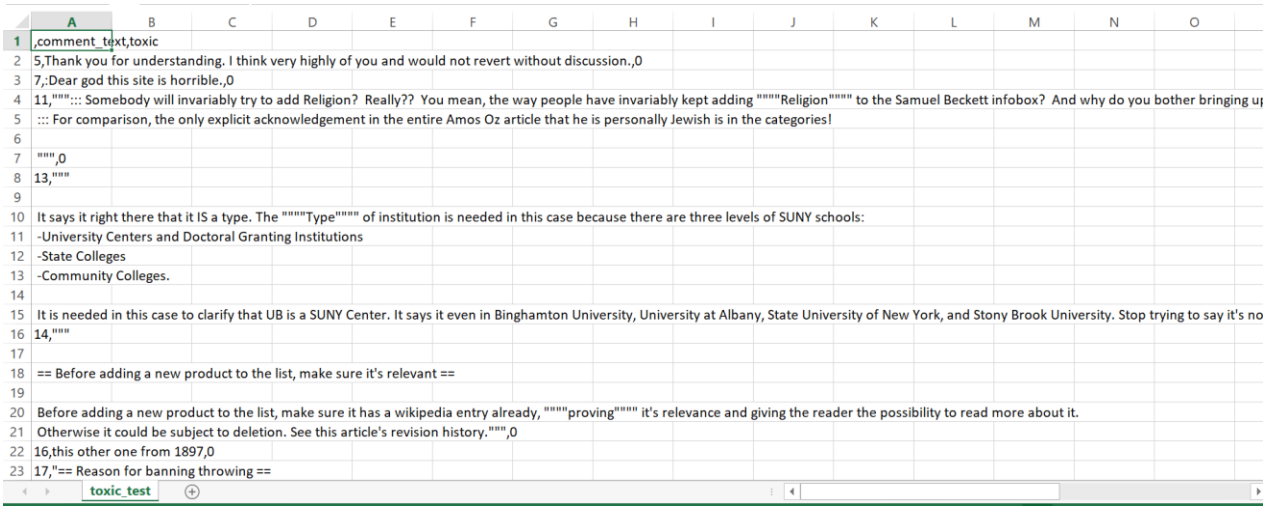
Figure 18 - Capture d'écran sur le site contenant le Dataset.

Les données sont stockées dans un fichier CSV (Excel) contenant des tweets déjà jugés (hate speech ou non hate speech),

Le fichier CSV est organisé en 3 colonnes comme suit :

- Index
- Comment_text : le texte des tweets.
- Toxic : le texte est-il un hate speech ou non (0 ou 1).

La figure ci-dessous montre une partie du fichier CSV de notre dataset :



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	,comment_text,toxic														
2	5,Thank you for understanding. I think very highly of you and would not revert without discussion.,0														
3	7,:Dear god this site is horrible.,0														
4	11,""":::: Somebody will invariably try to add Religion? Really?? You mean, the way people have invariably kept adding """"Religion"""" to the Samuel Beckett infobox? And why do you bother bringing up														
5	::: For comparison, the only explicit acknowledgement in the entire Amos Oz article that he is personally Jewish is in the categories!														
6															
7	""",0														
8	13,"""														
9															
10	It says it right there that it IS a type. The """"Type"""" of institution is needed in this case because there are three levels of SUNY schools:														
11	-University Centers and Doctoral Granting Institutions														
12	-State Colleges														
13	-Community Colleges.														
14															
15	It is needed in this case to clarify that UB is a SUNY Center. It says it even in Binghamton University, University at Albany, State University of New York, and Stony Brook University. Stop trying to say it's no														
16	14,"""														
17															
18	== Before adding a new product to the list, make sure it's relevant ==														
19															
20	Before adding a new product to the list, make sure it has a wikipedia entry already, """"proving"""" it's relevance and giving the reader the possibility to read more about it.														
21	Otherwise it could be subject to deletion. See this article's revision history.""",0														
22	16,this other one from 1897,0														
23	17,"== Reason for banning throwing ==														

toxic_test

Figure 19 - Le fichier CSV de notre dataset

Le deuxième dataset est fourni par les organisateurs du task « **Profiling Hate Speech Spreaders on Twitter »** de PAN [21] de CLEF 2021 qui est une série d'événements scientifiques et de tâches partagées sur la criminalistique des textes numériques. Le but de cette tâche est de prédire si un auteur dans la plateforme « Twitter » est toxique ou non.

Les données d'apprentissage et de test sont stockées dans des fichiers XML. Les données d'apprentissage sont constituées de 200 auteurs dont chacun a 200 tweets. Les données de tests contenaient 100 auteurs et chaque auteur contient 200 tweets.

Comme le montre la figure suivante, c'est un exemple d'un fichier XML représentant les tweets d'un seul utilisateur.

Les données du PAN n'était pas aussi grandes ce qui nous a obligé de les combiner avec les données du kaggle pour pouvoir entraîner le modèle et les insérer dans l'auto-encodeur qui nécessite une grande quantité de données.

Aussi les données de PAN n'avaient pas de fichier truth (targets), pour cela nous n'avons pas pu les insérer dans le random forest puisque cet algorithme est un algorithme supervisé et qui nécessite un target.

[21] https://pan.webis.de/

```
<?xml version="1.0"?>
<author lang="en" class="1">
  <documents>
    <document>
      <![CDATA[
        "Hey Jamal (snickering uncontrollable) You want some (PFFF) LEMONADE!" What an IDIOT! #URL#
      ]]>
    </document>
    <document>
      <![CDATA[
        RT #USER#: Cotton coming out with a banger #URL#
      ]]>
    </document>
    <document>
      <![CDATA[
        This is meant to be sarcasm but it's a good point considering how underwhelming the pandemic has been #URL#
      ]]>
    </document>
    <document>
      <![CDATA[
        Nick really just compared homosexuality to people shooting themselves in the head😂😒
      ]]>
    </document>
    <document>
      <![CDATA[
        PROTECT AMERICA FIRST! LET'S GO!!!!!!!! #URL#
```

Figure 20 - Exemple de la structure des données du PAN.

5.6 Implémentation

Afin d'implémenter notre solution proposée nous avons utilisé plusieurs techniques qui sont présentées en détail dans cette partie.

5.6.1 Chargement du Dataset

La figure suivante présente le code du chargement du premier dataset de Kaggle:

```
Entrée [2]: df = pd.read_csv('dataset2.csv') #krina csv

Entrée [3]: df.head(10) #affichina 10 lowline
```

Out[3]:

	Unnamed: 0	tweet	hate_speech
0	0	Explanation\r\nWhy the edits made under my use...	0
1	1	D'aww! He matches this background colour I'm s...	0
2	2	Hey man, I'm really not trying to edit war. It...	0
3	3	"\r\nMore\r\nI can't make any real suggestions...	0
4	4	You, sir, are my hero. Any chance you remember...	0
5	5	"\r\n\r\nCongratulations from me as well, use ...	0
6	6	COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK	1
7	7	Your vandalism to the Matt Shirvington article...	0
8	8	Sorry if the word 'nonsense' was offensive to ...	0
9	9	alignment on this subject and which are contra...	0

Figure 21 – Le code du chargement des données

Pour le 2ème dataset de PAN, il est nécessaire d'extraire seulement les tweets texte par texte pour chaque utilisateur et les organiser dans un tableau « Data Frame » avant la vectorisation et l'application du modèle sur ces derniers. La figure suivante **(Figure 27)** montre le script de traitement :

```
Entrée [46]: import xml.etree.ElementTree as ET
             import pandas as pd

             import xml.etree.ElementTree as ET
             mytree = ET.parse('0a3ce42bea89e2a92a28f685735e605e.xml')
             myroot = mytree.getroot()

             tweets = []
             cpt = 0
             for x in myroot[0]:
                     tweets.append(x.text)
             df_test = pd.DataFrame(tweets,columns=['tweet'])
```

5.6.2 Prétraitement des données

Le script de traitement des données de Kaggle est le suivant :

```
hate_df = df[['hate_speech','tweet']] # On a spécifier seulement les colonnes importante pour notre entrainement
```

Figure 21 - Chargement fichier XML PAN

```
hate_df['tweet'] = hate_df['tweet'].replace(r'http\S+', '', regex=True).replace(r'www\S+', '', regex=True) # eliminer les l

hate_df.loc[:,'tweet'] = hate_df['tweet'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop)]))
# eliminer les "stop words"

import string
import re

def remove_punctuations(text):
    for punctuation in string.punctuation:
        text = text.replace(punctuation, '')
    return text


def remove_rt(text):
    text = re.sub(r'\brt\b\s+',"",text).lstrip()
    return text

hate_df = df[['hate_speech','tweet']] # On a spécifier seulement les colonnes importante pour notre entrainement

hate_df.loc[:,"tweet"] = hate_df.tweet.apply(lambda x : str.lower(x))  # mettre tout les tweets en miniscule

hate_df['tweet'] = hate_df['tweet'].replace(r'http\S+', '', regex=True).replace(r'www\S+', '', regex=True) # eliminer les liens
```

Figure 22 - Partie prétraitement des données de Kaggle.

Pour, le prétraitement des données de PAN a nécessitait une étape supplémentaire de traitement puisqu'il contenait pleins d'emojis.

```
def remove_emojis(data):
    emoj = re.compile("["
        u"\U0001F600-\U0001F64F"
        u"\U0001F300-\U0001F5FF"
        u"\U0001F680-\U0001F6FF"
        u"\U0001F1E0-\U0001F1FF"
        u"\U00002500-\U00002BEF"
        u"\U00002702-\U000027B0"
        u"\U00002702-\U000027B0"
        u"\U000024C2-\U0001F251"
        u"\U0001f926-\U0001f937"
        u"\U00010000-\U0010ffff"
        u"\u2640-\u2642"
        u"\u2600-\u2B55"
        u"\u200d"
        u"\u23cf"
        u"\u23e9"
        u"\u231a"
        u"\ufe0f"
        u"\u3030"
                          "]+", re.UNICODE)
    return re.sub(emoj, '', data)

df_test.loc[:,"tweet"] = df_test['tweet'].apply(remove_emojis)
```

Figure 23 - La fonction qui supprime les emojis d'un tweet.

Les résultats obtenus après le prétraitement sont affichés comme suit

Entrée [12]:
```
df_test
```

Out[12]:

	tweet
0	rt for those asking no we will not carry joe b...
1	just another swamp rat being revealed i guess
2	we all should let the gop die the slow death i...
3	sounds good it may be a while i'm over all thi...
4	no i'm not at least not yet twitter needs to d...
...	...
195	lou dobbs is good too he s a huge trump supporter
196	yeah he showed his true colors during the deba...
197	he seems like such a little weasel
198	rt nikki haley takes a jab at elizabeth warren...
199	rt hunt his ass down name him and make his nas...

200 rows × 1 columns

Figure 24 – Résulatst des données de PAN aprés le prétraitement.

5.6.3 Modèle word2Vec

Dans le but d'apprendre la représentation vectorielle des mots, le word2vec prend un corpus de texte en entrée et produit les vecteurs de mots en sortie.

Pour cela, nous avons opté pour l'utilisation de word2vec Google gratuit qu'on peut trouver sur le web.

Word2Vec Google [22]: Est un outil qui fournit une implémentation efficace des architectures de sac de mots continus et de saut de gramme pour le calcul des représentations vectorielles des mots. Ces représentations peuvent ensuite être utilisées dans de nombreuses applications de traitement du langage naturel et pour des recherches ultérieures.

Le modèle de Google est formé de 100 milliards de mots, il comprend des vecteurs de mots pour un vocabulaire de 3 milliards de mots et d'expressions. Lorsqu'on applique ce modèle il prend l'intersection entre les mots de nos données et l'ensemble des mots du modèle de Google donc à la fin de l'intersection on aura juste un vecteur des mots pour nos données d'entrée.

La figure suivante présente le code source de l'importation du modèle word2 avec Google sur nos données.

```
Entrée [23]: from gensim.models import KeyedVectors
             model = KeyedVectors.load_word2vec_format('Models/GoogleNews-vectors-negative300.bin', binary=True)
```

Figure 25 - Code source word2vec Google.

Le vecteur de mots obtenu après l'utilisation du modèle word2vec Google est présenté dans la figureci-dessous :

```
Entrée [27]: train_vectors

Out[27]: array([[ 0.00212397,  0.00611471,  0.00336629, ..., -0.00745605,
                 -0.00091614, -0.00179647],
                [-0.00024287,  0.00042765, -0.00126668, ..., -0.00040527,
                  0.00201416,  0.00042643],
                [ 0.00536504,  0.00427938, -0.00025907, ..., -0.00863627,
                  0.00037343,  0.00276638],
                ...,
                [ 0.00158468, -0.00187744, -0.00125651, ..., -0.00435465,
                 -0.00273885, -0.000413  ],
                [ 0.00079447,  0.00378789,  0.00191282, ..., -0.00229126,
                 -0.00064657, -0.00113688],
                [ 0.00344808,  0.00214539,  0.00236626, ..., -0.00274974,
                  0.00385885,  0.00050171]], dtype=float32)
```

Figure 26 - Vecteur de mots après l'utilisation de word2vec.

[22] https://code.google.com/archive/p/word2vec/

```
Entrée [487]: test_vectors = make_vectors(word2vec_pan, model_google, 300)

Entrée [488]: x = test_vectors
              x_pan_test = encoder2.predict(x)

              x_pan_test
```

Figure 27 - Code de l'application de word2vec sur les données du PAN.

Les résultats obtenus sont présentés dans la figure suivante.

```
Out[488]: array([[ 173.26428, -384.0465 ,  821.12897, ...,  635.1248 ,  253.32887,
                   387.04257],
                 [ 173.88191, -383.22687,  821.2591 , ...,  634.78485,  251.72008,
                   386.7838 ],
                 [ 174.6537 , -383.0139 ,  823.0549 , ...,  634.82294,  252.29277,
                   386.30966],
                 ...,
                 [ 174.29419, -382.4304 ,  821.03265, ...,  634.0331 ,  251.52489,
                   385.91132],
                 [ 170.83243, -386.8434 ,  821.20374, ...,  637.52094,  256.8112 ,
                   390.1465 ],
                 [ 175.32965, -384.06796,  825.6049 , ...,  636.2456 ,  252.58849,
                   387.4446 ]], dtype=float32)
```

Figure 28 - Résultats de l'application de word2vec sur les données du PAN.

5.6.4 Modèle Random Forest

```
Entrée [33]: from sklearn.ensemble import RandomForestClassifier

Entrée [34]: model_forest = RandomForestClassifier(n_estimators = 100)

Entrée [35]: model_forest.fit(x_train, y_train)
    Out[35]: RandomForestClassifier()

Entrée [36]: predict = model_forest.predict(x_train)
```

Figure 29 - Code source « Random Forest ».

5.6.5 Modèle Randomforest_autoencodeur

Dans cette partie nous allons présenter le code de notre 2éme modèle qui est la combinaison entre le « Random Forest » et « l'Auto-encodeur ».

```
n_inputs = x_train.shape[1]

t = MinMaxScaler()
t.fit(x_train)
x_train = t.transform(x_train)
x_test = t.transform(x_test)

visible = Input(shape=(n_inputs,))

e = Dense(n_inputs*2)(visible)
e = BatchNormalization()(e)
e = LeakyReLU()(e)

e = Dense(n_inputs)(e)
e = BatchNormalization()(e)
e = LeakyReLU()(e)

n_bottleneck = round(float(n_inputs) / 2.0)
bottleneck = Dense(n_bottleneck)(e)

d = Dense(n_inputs)(bottleneck)
d = BatchNormalization()(d)
d = LeakyReLU()(d)

d = Dense(n_inputs*2)(d)
d = BatchNormalization()(d)
d = LeakyReLU()(d)

output = Dense(n_inputs, activation='linear')(d)

model = Model(inputs=visible, outputs=output)

model.compile(optimizer='adam', loss='binary_crossentropy')



history = model.fit(x_train, x_train, epochs=50, batch_size=16, verbose=2, validation_data=(x_test,x_test))


encoder = Model(inputs=visible, outputs=bottleneck)
```

Figure 30 - Code source de « autoencodeur_randomforest »

5.7 Mesures d'évaluation

Après que nous avons terminé l'apprentissage de notre modèle, il est temps pour appliquer notre modèle final sur de nouveaux exemples de données afin de pouvoir estimer et trancher la qualité et la fiabilité de notre nouvelle solution.

L'évaluation de notre modèle qui appartient aux modèles de classification et retourne des valeurs binaires soit 1 ou 0, consiste à savoir si le nouveau message appartient à la classe 0 (non hate speech) ou à la classe 1 (hate speech).

Pour cela nous avons utilisé la matrice de confusion en insérons de nouvelles données (un nouveau dataset qui ne contient pas de réponse sur la catégorie du tweet hate speech ou non).

La Matrice de confusion :

Appelons "**positive**" la classe correspondant à un hate speech et "**négative**" l'autre. Si on prédit un hate speech quand il y en a bien un, on fait une prédiction "positif" qui est correcte, c'est un **vrai positif.**

Si par contre cette prédiction est incorrecte, il s'agit d'un **faux positif.** Et ainsi de suite. On appelle aussi parfois "**erreur de type I**" les faux positifs, et "**erreur de type II**" les faux négatifs.

Le rappel : est le **taux de vrais positifs,** C'est la capacité de notre modèle à détecter tous les Hates speechs.

$$Rappel = \frac{\mathrm{TP}}{TP + FN}$$

Précision : C'est la capacité de notre modèle de prédire de bons résultats hate speechs ou non par rapport aux données totales.

$$Précision = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$$

Accuracy :est le niveau d'exactitude d'une mesure par rapport à sa valeur réelle.

$$Accuracy = \frac{\text{nombre d'instances correctement prédites}}{nombre\ total\ d'instances}$$

F-Score : évaluer un compromis entre rappel et précision (moyenne harmonique)

$$F - Score = 2 * \frac{\mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

Donc à la fin de cette partie on pourra savoir si notre modèle pourra prédire correctement la cible dans le cas de nouvelles données à venir sachant que les données futures seront des données sans des cibles (on n'aura pas les informations sur les tweets si sont haineux ou pas).

5.7.2 Résultats

Résultats de classification du dataset de Kaggle

Le tableau 3 est une représentation de la matrice de confusion du problème du hate speech.

	Non Hate speech	Hate speech
Résultats du test positif	Faux positif	Vrai positif
Résultats du test négatif	Vrai négatif	Faux négatif

Tableau 2 - Représentation de la matrice de confusion de notre problème.

La figure ci-dessus représente le résultat de la matrice de confusion des données de Kaggle ou on distingue 28470 vrais positifs, 1531 faux négatifs, 1528 pour les vrais négatifs et 380 pour les Faux positifs.

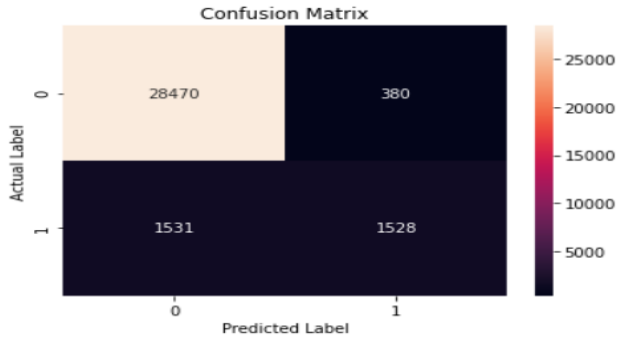


Figure 31 - Résultat matrice de confusion de nos données de Kaggle.

Le tableau suivant présente les résultats obtenus à partir du dataset de Kaggle en utilisant nos deux modèles Random Forest et Auto-encodeur+Random Forest.

En utilisant le modèle d'Autoencodeur+Random Forest, nous avons obtenu une haute précision de 0,94 par rapport au modèle Random Forest.

Les résultats de rappel et F1-score sont meilleurs aussi dans les modèles Autoencodeur+Random Forest.

	Précision	Rappel	F1-Score
Random Forest	93.64% (0,93)	0,68	0,78
Auto-encodeur +Random Forest	94 .01% (0,94)	0,72	0 ,81

Tableau 3 - Tableau comparatif sur les mesures de performances sur le dataset de Kaggle.

Dans la figure 32, nous avons comparé nos résultats avec ceux des autres travaux qui ont utilisé le même dataset. Nous avons pris les résultats du site de Kaggle.

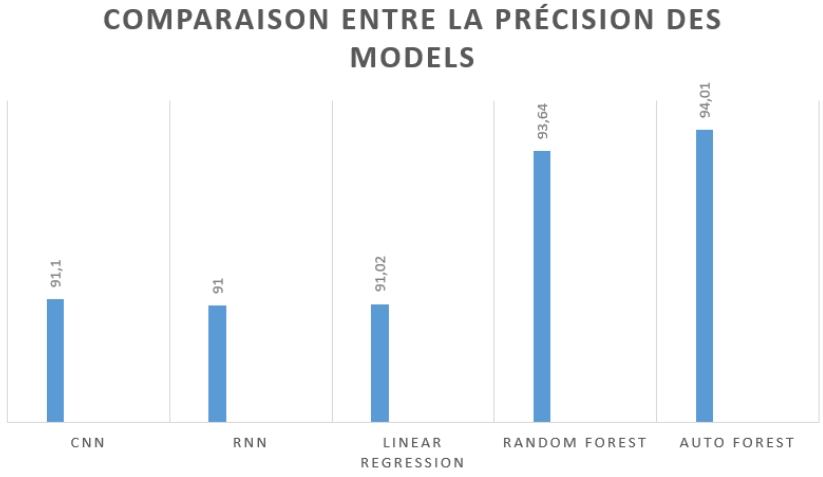


Figure 32 - Comparaison entre la précision des différents modèles.

Résultats de classification du dataset PAN

Figure 33 - Les résultats du dataset PAN fournit par notre

Cette figure représente les résultats que nous avons eu après l'application du modèle Auto-encodeur Random Forest, les 0 et 1 (hate speech ou pas) le format hexadécimal est pour identifier chaque auteur.

Afin de faciliter la tâche et que les résultats soient bien lisibles nous avons à la fin les présenter sous un dossier contenant pour chaque auteur un fichier XML qui définit si l'auteur est toxique ou pas.

```
<author id="8e5a604d6328d4b15d119b9601f5d3c2"
lang="en"
type="1"/>
```

Figure 34 - Code XML qui représente un auteur toxique.

Nous avons envoyé les 100 fichiers XML de test aux responsables du challenge de Hate speech de PAN afin de nous renvoyer les résultats de notre modèle. Comme le monte la figure suivante nous avons atteint 63% d'accuracy.

	XLMR-LSTM	62.0	73.0	67.5
47	ipek	58.0	77.0	67.5
47	schlicht21	58.0	77.0	67.5
47	peirano	59.0	76.0	67.5
47	russo	55.0	80.0	67.5
	MBERT-LSTM	59.0	75.0	67.0
51	kazzaz	55.0	77.0	66.0
52	dorado	60.0	71.0	65.5
53	kobby	53.0	77.0	65.0
53	kern	54.0	76.0	65.0
53	espinosa	64.0	66.0	65.0
56	labadie	51.0	78.0	64.5
57	silva	56.0	69.0	62.5
57	garibo	57.0	68.0	62.5
59	estepicursor	51.0	72.0	61.5
60	spears	52.0	68.0	60.0
	TFIDF-LSTM	61.0	51.0	56.0
61	barbas	46.0	50.0	48.0
62	dukic	75.0	-	-
63	tosev	70.0	-	-
64	amir	68.0	-	-
65	siebert	68.0	-	-
66	iteam	65.0	-	-
67	amina*	63.0	-	-

* Result sent beyond the deadline.

Figure 35:résultats de notre modèle sur le site PAN .

5.8 Conclusion

Dans ce dernier chapitre, nous avons présenté notre solution pour détecter le hate speech dans les réseaux Twitter, en utilisant la combinaison de « Random Forest » et « Auto-encodeur ». Nous avons listé les différents outils utilisés, nous sommes allez jusqu'à les mesure de performances et le test et les résultats de nos modèles pour pouvoir enfin passer à la conclusion générale.

Conclusion et Perspectives

L'objectif de notre recherche était de détecter les hates speechs (discours de haine) dans les réseaux sociaux vus la propagation rapide de son utilisation et qui touche toutes les catégories de la société en effectuant des pertes physiques, sociales et surtout psychologiques.

Pour cela nous avons voulu concevoir et implémenter une solution à ce majeur problème qui ne cesse d'augmenter et qui atteint toutes les catégories des gens.

Pour pallier à ce problème nous avons proposé une nouvelle méthode en utilisant l'intelligence artificielle plus précisément le Machine Learning et le Deep Learning

Nous avons pensé à utiliser l'algorithme auto-encodeur qui a fait une révolution remarquable dans le domaine des images et qui a été utilisé auparavant juste pour ce domaine. En essayant de l'adapter à notre problème, et l'utiliser dans le texte, afin de donner une nouvelle solution qui aidera les chercheurs à savoir si l'utilisation de cette nouvelle technique va bouleverser le domaine du NLP tout comme les images. Donc, nous avons proposé un premier modèle qui se base seulement sur l'algorithme du Machine Learning Random Forest. Nous avons proposé aussi un autre modèle en combinant l'algorithme auto-encodeur avec l'algorithme Random Forest. La comparaison des deux modèles permet de conclure si l'auto-encodeur a ajouté sa touche dans la précision de la détection du HS ou pas.

Nos résultats finaux ont démontré une haute précision (0,94) pour la combinaison du modèle Random Forest et auto-encodeur qui surclasse les autres modèles existant dans ce domaine (CNN, RNN, Logistic régression, Random Forest) modèles existants et qui ont utilisé le même dataset.Nous avons aussi eu une Accuracy de 63% dans notre participation au challenge du site PANen utilisant la combinaison des deux datasets(Kaggle+PAN). Ce qui nous a confirmé que l'utilisation de l'auto-encodeur était meilleure que les autres algorithmes de classification déjà utilisés dans ce domaine et que l'auto-encodeur peut être utilisé et peut faire la différence dans le domaine du texte.

Au cours de ce travail nous avons rencontré beaucoup de difficultés que nous avons pu les surmonter. Concernant le domaine de l'IA, nous n'avions pas une bonne base ce qui nous a empêché d'avancer facilement et rapidement dans nos recherches. Ajoutant un grand problème que nous avons rencontré lors de nos recherches qui est le nombre limité et restreint des articles dans le domaine du hate speech. De plus, la puissance de nos machines qui ne supportait pas une très grande quantité de données.

Comme perspective, nous voulons réaliser un script qui pourra extraire les données directement de la plateforme pour construire notre propre dataset, commençant par la plateforme Twitter en se prolongeant vers les autres réseaux sociaux.

Notre deuxième perspective la plus optimiste et futuriste est de fournir un modèle qui pourra différencier entre les types de discours de haine et le taux de gravité de ce discours et prédire aussi les conséquences qu'il pourra effectuer ce speech sur les différentes catégories sociales.

Pour conclure, ce projet était une expérience très enrichissante, il nous a permis d'acquérir des compétences et maitriser celles existantes. Nous avons également appris à maitriser le langage python ainsi que la plateforme Google colab et Jupyter que nous continuerons de les utiliser dans nos travaux futurs.

Bibliographie

[1][Enligne]Available:https://www.blogdumoderateur.com/internet-reseaux-sociaux-mobile-octobre

[2][En ligne].Available:https://spip.teluq.ca/inf1160/IMG/pdf/inf1160-notionsfondamentales.pdf:

[3][Enligne].Available:https://www.pmtic.net/sites/default/files/filemanager/memos/pmtic_com_media_reseaux_sociaux.pd

[4]]ELAGGOUNE_ACHRAF, Détection des communautés dans les réseaux sociaux :UniversitéUniversité de 8 Mai 1945 – Guelma -.2020.

[5] NEDIOUI, MED ABDELHAMID. *Fouille et apprentissage automatique dans les réseaux sociaux dynamique*. Diss. Université Mohamed Khider-Biskra, 2015.

[6][En ligne].Available:http://socialonline.over-blog.com/2016/01/les-reseaux-sociaux-et-son-histoire.html

[7][Enligne].Available:https://esrc.ukri.org/research/impact-toolkit/social-media/twitter/what-is-twitter/

[8][Enligne].Available:2019-02-intelligence-artificielle-etat-de-l'art-et-perspectives.pour la France

[9] Rajkomar, Alvin, Jeffrey Dean, and Isaac Kohane. "Machine learning in medicine." New England Journal of Medicine 380.14 (2019): 1347-1358.

[10] [En ligne]. Available : https://machinelearnia.com/apprentissage-supervise-4-etapes/

[11] 2019 Guillaume Saint-Cirgue Apprendre le Machine Learning en une semaine

[12] Marref, Nadia. Apprentissage Incrémental & Machines à Vecteurs Supports. Diss. Université de Batna 2, 2013.

[13] [En ligne]. Available : https://www.natural-solutions.eu/blog/histoire-du-deep-learning

[14] Lounis, Katia, and Dahbia Moussi. La Classification d'images d'insectes ravageurs en utilisant le Deep Learning. Diss. Université Mouloud Mammeri, 2020.

[15] [En ligne]. Available : https://waytolearnx.com/2018/11/difference-entre-machine-learning-et-deep-learning.html

[16] D. Rumelhart, G. Hinton et R. Williams, «Apprentissage des représentations internes par propagation d'erreur», UCSD, La Jolla, CA, USA, Tech. Rép. ICS-8506, 1985.

[17] Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." Neural networks 61 (2015): 85-117.

[18] [En ligne]. Available: Top 15 Deep Learning applications that will rule the world in 2018 and beyond, Vartul Mittal, 3 Oct 2017.

[19] Hasan, Mohamadally, and Fomani Boris. "Svm: Machines à vecteurs de support ou séparateurs à vastes marges." Rapport technique, Versailles St Quentin, France. Cité 64 (2006).

[20] [En ligne]. Available: https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/

[21] Scornet, Erwan. Apprentissage et forêts aléatoires. Diss. Paris 6, 2015.

[22] Marée, Raphaël. Classification automatique d'images par arbres de décision. Diss. University of Lìege-Electrical Engineering and Computer Science, 2005.

[23] Francois, Olivier, and Philippe Leray. "Evaluation d'algorithmes d'apprentissage de structure pour les réseaux bayésiens." Proceedings of 14ème congrès francophone reconnaissance des formes et intelligence artificielle, RFIA. 2004.

[24] YACINE, BENATIA. "Deep Auto-Encodeur pour la Reconnaissance de Visage."

[25][En ligne]. Available: https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

[26] Khan, Asifullah, et al. "A survey of the recent architectures of deep convolutional neural networks." Artificial Intelligence Review 53.8 (2020): 5455-5516.

[27] Kritli M C, BOUKENAOUI A, Mesure de la gravité des signes de dépression à partir des réseaux sociaux.Blida 10 septembre 2020

[28] [En ligne]. AVAILABLE: https://developersbreach.com/convolution-neural-network-deep-learning/

[29] L.DEKKICHE, Classification des arythmies ECG avec des méthodes de Machine Learning et de Deep Learning, Tizi-Ouzou 2020.

[30] Chowdhury, Gobinda G. "Natural language processing." Annual review of information science and technology 37.1 (2003): 51-89.

[31] Liddy, Elizabeth D. "Natural language processing." (2001).

[32] Poletto, Fabio, et al. "Resources and benchmark corpora for hate speech detection: a systematic review." Language Resources and Evaluation (2020): 1-47.

[33] Davidson, Thomas, et al. "Automated hate speech detection and the problem of offensive language." Proceedings of the International AAAI Conference on Web and Social Media. Vol. 11. No. 1. 2017.

[34] Poletto, Fabio, et al. "Resources and benchmark corpora for hate speech detection: a systematic review." Language Resources and Evaluation (2020): 1-47.

[35] N. Chetty et S. Alathur, «Revue des discours de haine dans le contexte des réseaux sociaux en ligne», Agressivité. Comportement violent., vol. 40, non. Avril, p. 108-118, 2018

[36] Ruwandika, N. D. T., and A. R. Weerasinghe. "Identification of hate speech in social media." 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer). IEEE, 2018.

[37] Warner, William, and Julia Hirschberg. "Detecting hate speech on the world wide web." Proceedings of the second workshop on language in social media. 2012.

[38] Kwok, Irene, and Yuzhou Wang. "Locate the hate: Detecting tweets against blacks." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 27. No. 1. 2013.

[39] Burnap, Peter, and Matthew Leighton Williams. "Hate speech, machine classification and statistical modelling of information flows on twitter: Interpretation and communication for policy decision making." (2014).

[40] Liu, Shuhua, and Thomas Forss. "Combining N-gram based Similarity Analysis with Sentiment Analysis in Web Content Classification." KDIR. 2014.

[41] Tulkens, Stéphan, et al. "A dictionary-based approach to racism detection in dutch social media." arXiv preprint arXiv: 1608.08738 (2016).

[42] Bethard, Steven, et al. "Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)."

[43] Davidson, Thomas, et al. "Automated hate speech detection and the problem of offensive language." Proceedings of the International AAAI Conference on Web and Social Media. Vol. 11. No. 1. 2017.

[44] Kovács, G., Alonso, P. & Saini, R. Les défis de la détection des discours de haine dans les médias sociaux. SN COMPUT. SCI. 2, 95 (2021).

[45] Rizoiu, Marian-Andrei, et al. "Transfer learning for hate speech detection in social media." arXiv preprint arXiv: 1906.03829 (2019).

[46] Alshaalan, Raghad, and Hend Al-Khalifa. "Hate Speech Detection in Saudi Twittersphere: A Deep Learning Approach." Proceedings of the Fifth Arabic Natural Language Processing Workshop. 2020.

[47][Enligne].Available:http://vision.gel.ulaval.ca/~jflalonde/cours/4105/h15/tps/results/projet/OLGAG28/index.html

[48][Enligne].Available:https://mlwhiz.com/blog/2019/01/17/deeplearning_nlp_preprocess/

[49][En ligne]. Available: https://community.alteryx.com/t5/Data-Science/Word2vec-for-the-Alteryx-Community/ba-p/305285