République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université 8Mai 1945 – Guelma
Faculté des sciences et de la Technologie
Département d'Electronique et Télécommunications

Mémoire de fin d'étude
Pour l'obtention du diplôme de Master Académique

Domaine : Sciences et Technologie
Filière : Télécommunications
Spécialité : Réseaux et Télécommunications

CATEGORISATION AUTOMATIQUE DU CONTENU OFFENSIF

Présenté par :

NASSIROU DAOUDA Aminou

Sous la direction de :

Dr. KHEIREDDINE Abainia

Juin 2022

Remerciements

Je remercie infiniment ALLAH le tout puissant pour la santé, la force et le courage qu'il m'a donnés tout au long de notre parcours.

A la fin d'une formation, il est de tradition d'exprimer ses reconnaissances à l'égard de ceux qui, par leurs apports multiformes ont contribué à l'aboutissement et à la réussite de celle-ci. C'est ainsi qu'à travers ce mémoire, je tiens tout d'abord à remercier du fond de mon cœur mon encadreur Dr. KHEIREDDINE Abainia grâce à sa simplicité, son entière disponibilité et ses conseils dans l'acheminement du projet.

J'exprime également ma gratitude envers mes parents qui m'ont beaucoup aidé.

Je ne saurais terminer sans adresser mes sincères remerciements aux membres de jury d'avoir accepté d'examiner et d'évaluer mon travail.

Pour finir, je remercie tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Je dédie ce travail

A mes très chers parents Mr. Nassirou Daouda et Mme. Nana fatimatou dont je suis redevable de tant de choses, qui m'ont soutenu et encouragé durant mes études et toute ma vie, et qui ont veillé sur moi avec beaucoup d'amour.

Que Dieu me donne l'opportunité de leur rendre un peu de cette dette.

A mes frères et sœurs
A mes ami(e)s
A mes cousins et cousines
A toute ma famille
A toute la communauté nigérienne de Guelma

Résumé

Ce travail est basé sur la détection et la catégorisation du contenu offensif et abusif dans les commentaires arabes dialectal (dialect algérien en particulier). On se base sur Facebook, qui est la plateforme des réseaux sociaux la plus utilisée en Algérie. De ce fait, nous avons utilisé une base de données de plus 8,4k textes annotés en trois catégories telle qu'offensif, abusif et normal. Cependant, nous avons utilisé plusieurs algorithmes d'apprentissage automatique comme le *SVM*, le *NB* et le *SGD*. Cependant, nous avons obtenus des résultats remarquables qui peuvent être plus amélioré par des recherches supplémentaires.

Abstract

This work is based on the detection and categorization of offensive and abusive content in Arabic dialect comments (Algerian dialect in particular). It is based on Facebook, which is the most used social networking platform in Algeria. Therefore, we used a database of more than 8.4k annotated texts in three categories such as offensive, abusive and normal. However, we used several machine learning algorithms such as SVM, NB and SGD. However, we obtained remarkable results that can be further improved by additional research.

Résumé en arabe

وجه على الجزائرية اللهجة) العربية باللهجة التعليقات في والمسيء المسيء المحتوى وتصنيف كشف على العمل هذا يعتمد قاعدة استخدمنا ، لذلك .الجزائر في استخدامًا الاجتماعي التواصل منصات أكثر وهو ، Facebook على نعتمد .(الخصوص العديد استخدمنا فقد ، ذلك ومع .والعاديّة والهجومية الهجومية مثل فئات ثلاث في شرحه تم نص 8400 من أكثر تضم بيانات من بمزيد تحسينها يمكن رائعة نتائج على حصلنا فقد ، ذلك ومع .SGD و NB و SVM مثل الآلي التعلم خوارزميات من البحث.

BA	Bat Algorithm
CA	Classique arabe
C-LSTM	Convolutional-Long-Short Term Memory
CNN	Convolutional Neural Networks
DL	Deep Learning
FS	Feature Selection
GWO	Grey Wolf Optimization
LSTM	Long-Short Term Memory
ML	Machine Learning
MSA	Modern Standard Arabic
NB	Naïve Bayes
NLP	Natural Langage Processing
PSO	Particle Swarm Optimization
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine

Introduction Générale

De nos jours, les plateformes des réseaux sociaux sont devenues un endroit où les gens peuvent communiquer entre eux et partager leur opinion ou avis. Cependant, certaines personnes profitent de cette facilité de communication en ligne pour envoyer certaines formes d'intimidation en ligne, de discours haineux, de contenu violent tout en utilisant un langage abusif dans le but d'offenser, de menacer un individu ou groupe de personne. En effet, ces genres de message offensif ont des conséquences désastreuses (voire fatales) envers les victimes, ce qui provoque la dépression, les troubles mentaux ou même conduire au suicide.

Toutefois, pour contrer et éviter ces risques, plusieurs approches sur la détection automatique des langages offensifs ont été faites dans divers langues. De ce fait, notre travail se concentre spécialement sur la langue arabe et en particulier le dialecte algérien, car notre base de données utilisée pour la catégorisation des textes a été recensée dans les commentaires arabes, plus spécifiquement algériens.

Le but de notre travail est de développer un modèle fondé sur l'apprentissage automatique pour la catégorisation des textes afin de détecter le langage offensif dans les commentaires des medias sociaux pour ainsi prévenir de tels comportements et de protéger le bien-être des utilisateurs. De ce fait, nous nous sommes concentrés sur la classification des commentaires en trois catégories : offensifs, abusifs et normaux, qui sont dans la base de données qu'on a utilisé. Notre mémoire est scindé en trois chapitres principaux.

Dans un premier chapitre intitulé langage offensif sur les réseaux sociaux, nous parlons en premier lieu sur le langage et le dialecte arabe, ensuite les différents types du langage offensif en ligne, ainsi que leurs effets négatifs sur les utilisateurs. En outre, nous détaillons la modélisation du contenu abusif, et enfin les perspectives contre les messages offensifs.

Dans le deuxième chapitre intitulé méthodologie, nous donnons un aperçu général sur la méthodologie pour la classification des textes, ainsi détailler notre approche générale sur le travail.

Enfin, dans le dernier chapitre intitulé résultats et expériences, nous présentons les résultats finaux suite à notre application sur la classification des textes.

Chapitre I :
Langage offensif sur les réseaux sociaux

I.1. Introduction

Les plateformes de médias sociaux sont vues comme des lieux (virtuels) où de nombreuses personnes de différentes parties du monde peuvent discuter et partager leurs vies quotidiennes. De ce fait, le langage offensant sur les médias sociaux est un problème majeur qui touche de nombreux individus et groupes. Dans ce chapitre, nous allons donner un aperçu du langage formel et dialecte arabe, ainsi nous définissons le langage offensant avec ses catégories et les résultats de la dépression.

I.2. Langage et dialecte

I.2.1. Langage formel

Le langage formel est un ensemble de mots qui possède un alphabet, dans lequel contiennent des symboles et des lexèmes dont on se sert pour construire les mots du langage.

I.2.2. Langage Arabe

Environ 420 millions de personnes dans le monde utilisent l'arabe comme langue maternelle. En arabe, les scripts se lisent et s'écrivent de droite à gauche. Il comporte 28 lettres, dont chacune peut être écrite différemment selon sa position dans le mot et aussi les voyelles sont représentées par des signes diacritiques. L'arabe a deux formes qui sont le CA et MSA.

L'arabe classique ou classique arabe (CA) : souvent connu sous le nom d'arabe coranique, et est la langue écrite du Saint Coran (le texte sacré et spirituel de l'Islam). L'arabe classique n'est plus une langue parlée et est principalement utilisé à des fins religieuses en raison de son âge [1].

L'Arabe Standard Moderne ou Modern Standard Arabic (MSA) : est la langue officielle du monde arabe, et est la langue prédominante des médias et de la culture. Il est utilisé dans les réunions formelles, la politique, les actualités, les journaux, etc. MSA est basé sur CA en termes syntaxiques, morphologiques et phonologiques, mais lexicalement moderne. Ce n'est pas la langue maternelle d'un Arabe, mais c'est la langue d'enseignement dans tout le monde arabe, mais c'est une langue écrite plutôt que parlée [1].

I.2.3. Dialecte Arabe

L'arabe dialectal est dérivé du système linguistique arabe originel : l'arabe littéraire. Pour reprendre notre métaphore, les dialectes arabes sont comme les olives d'un olivier, ils sont tous différents, mais issus d'un seul noyau : l'arabe littéraire. L'arabe dialectale est généralement désigné de deux manières en arabe, et ce, en fonction des régions :

- dans les pays de l'ouest (Maroc, Algérie, Tunisie, Libye), on utilise le terme " darija " (الدَّارِجَة) pour désigner l'arabe maghrébin ou l'arabe occidentale ;
- à l'est, dans la péninsule arabique, on utilise plutôt le terme " 3ammiya " (العَامِيَّة).

Le darija et le 3amiyya sont donc deux termes regroupant l'ensemble des dialectes arabes.

Les dialectes arabes actuels proviennent de deux phénomènes majeurs. Le premier est l'évolution naturelle de l'arabe, langue née dès avant l'arrivée de l'islam dans la péninsule Arabique. L'arabe est devenu à partir du VIIe siècle, une langue religieuse, celle du Coran, ainsi que politique et administrative. Le second phénomène est l'islamisation et l'arabisation de populations non musulmanes et non arabophones qui adoptent la langue arabe au rythme des conquêtes musulmanes à partir du VIIe siècle [2].

Le dialecte arabe est une langue de communication directe et immédiate. C'est un arabe parlé utilisé par les résidents des pays arabophones pour communiquer entre eux au quotidien, il n'est donc pas destiné à être écrit [3]. Cependant, de nos jours, il est largement utilisé dans les messages texte ou les communications sur les réseaux sociaux, et même dans la publicité. Pour cela, certaines personnes utilisent l'alphabet arabe classique, tandis que d'autres utilisent phonétiquement l'alphabet latin. Ce qui est intéressant ici, c'est l'origine de ces dialectes et comment tous ces fruits découlent d'un noyau unique [3].

Avec la propagation de l'islam, ces dialectes sont apparus pour la première fois après la conversion de ces peuples à l'islam. C'était donc au départ un mélange d'arabe littéraire et de langues locales. Ce mélange fut alors à nouveau soumis à la colonisation, aux flux migratoires ou aux échanges commerciaux. Il existe donc de nombreux dialectes plus ou moins éloignés les uns des autres, d'où chaque pays ou même chaque région d'un même pays a son propre dialecte.

I.2.4. Dialecte Algérien

L'Algérie est le plus grand pays du Maghreb (Afrique du Nord), avec une superficie d'environ 2,4 millions de km2. L'arabe algérien, ou le dialecte parlé en Algérie comme langue maternelle de 70 à 80% et maitrisée par 95 à 99% de la population, elle est localement appelé darija avec une structure linguistique compliquée. En particulier, le processus d'arabisation a été causé par l'adoption de l'arabe par les indigènes berbères (parlant berbère ou tamazight) et les 130 années de profonde colonisation par la France [1]. De plus, d'autres langues comme le turc, l'italien et l'espagnol ont également influencé le dialecte arabe algérien. Cependant, Le dialecte algérien est fréquemment utilisé par les utilisateurs sur les plateformes de médias sociaux telles que Twitter, Facebook, Instagram, etc.

I.3. Langage offensif

Un langage offensit est un comportement conçu pour blesser les sentiments, susciter la colère, le ressentiment, le dégoût ou l'indignation dans l'esprit d'une personne raisonnable [4]. Le langage offensif peut être divisé en plusieurs catégories selon le degré de l'infraction. Ainsi, il peut prendre différentes formes comme le discours de haine ou hate speech, le racisme, le sexisme, la cyberintimidation, les commentaires toxiques, etc.

I.3.1. Le discours haineux ou hate speech

Le discours de haine désigne de manière générale un discours offensif envers un groupe de personnes avec l'intention de causer du tort, de la violence ou de l'instabilité sociale. Il est aussi défini comme un langage destiné à dénigrer, humilier ou insulter les membres d'un groupe particulier ou à manifester de l'hostilité à leur égard [1]. Cela peut également être défini comme

l'utilisation de stéréotypes pour désigner des personnes en fonction de leur appartenance à certains groupes, faire des déclarations négatives sur les groupes minoritaires, utiliser des termes raciaux et désobligeants pour causer du tort [1].

Le concept de discours de haine touche au choc de la liberté d'expression et des droits individuels, collectifs et des minorités, ainsi qu'aux concepts de dignité, de liberté et d'égalité. Il n'est pas facile à définir mais peut être reconnu par sa fonction [5].

Dans les législations nationales et internationales, le discours de haine désigne les expressions prônant l'incitation au mal, notamment la discrimination, l'hostilité, la radicalisation, la violence verbale et/ou physique, fondée sur l'identité sociale des cibles. Le discours de haine peut inclure, mais sans s'y limiter, un discours qui préconise, menace ou encourage des actes violents [5]. D'une façon brève, tout type de communication par la parole, l'écrit ou le comportement, qui attaque ou utilise un langage péjoratif ou discriminatoire à l'égard d'une personne ou d'un groupe en raison de ce qu'ils sont, en d'autres termes, en raison de leur religion, de leur ethnicité, de leur nationalité, de leur race, de leur couleur, de leur ascendance, de leur sexe ou de tout autre facteur d'identité, est conçu comme un discours haineux.

I.3.2. La Cyberintimidation

La cyberintimidation, est un geste d'intimidation réalisé dans le cyberespace c'est-à-dire en ligne, c'est lorsqu'une personne est méchante avec une autre personne, qu'elle la menace, la blesse, l'humilie ou l'intimide en utilisant un moyen technologique: réseaux sociaux, sites Web, messageries (courriels, textos), etc [6]. Elle peut prendre les formes suivantes : se moquer de quelqu'un ou de son apparence, mettre en colère ou faire honte les personnes ciblées. Si la cyberintimidation évolue constamment, la forme la plus courante consiste à insulter quelqu'un ou à lui faire des commentaires méchants en ligne [6].

« La cyberintimidation implique l'utilisation des technologies de l'information et de la communication comme le courriel, les messages textuels envoyés par téléphone cellulaire ou par téléavertisseur, la messagerie instantanée, les sites Web personnels diffamatoires et les sondages diffamatoires sur sites Web personnels dans le but de renforcer un comportement hostile, délibéré et répétitif d'un individu ou d'un groupe qui cherche à blesser les autres. » (Traduction libre – Bill Belsey) [7].

Comme variante de cyberintimidation, il y'a aussi le cyberharcèlemment qui est une sorte d'acte agressif, intentionnel perpétré par un individu ou un groupe d'individus au moyen de formes de communication électroniques, de façon répétée à l'encontre d'une victime qui ne peut facilement se défendre seule [8]. Cependant, il y'a une légère différence avec la cyberintimidation qui consiste généralement à faire des messages électroniques qui intimident ou menacent leur destinataire alors que le cyberharcèlement désigne l'utilisation répétée d'un moyen de communication électronique afin de harceler ou d'effrayer une autre personne [9]. Il existe plusieurs formes de cyberintimidation ou cyberharcèlement, voici quelques exemples :

- Créer un site diffamatoire où on insulte et humilie une organisation ou quelqu'un en particulier
- Faire circuler des propos haineux liés aux orientations sexuelles, à la religion ou au racisme à propos de quelqu'un par l'entremise des messageries instantanées, des sites Web, des messages textes, des courriels
- Écrire des commentaires désobligeants, haineux sur le blogue de quelqu'un;
- Rendre accessibles ou diffuser des photos embarrassantes dans Internet
- Encourager l'envoi de messages électroniques hostiles à une personne (ex. : une personne distribue l'adresse courriel de quelqu'un et demande qu'on lui envoie des insultes)
- Subtiliser l'identité de quelqu'un pour inscrire de faux messages sur des sites particuliers (ex. : annoncer l'homosexualité de quelqu'un sur des forums de discussion)
- Mettre en ligne des photos ou vidéos de nature privée après une rupture amoureuse;
- Envoyer des insultes ou menaces directement à la personne par courriel, messagerie instantanée ou messagerie texte. Ces insultes ou menaces peuvent être envoyées en utilisant une fausse identité ou le mot de passe de quelqu'un d'autre.

Il en résulte que les victimes de la cyberintimidation peuvent se sentir encore plus accablées et impuissantes que dans les cas d'intimidation traditionnelle. La cyberintimidation peut donc atteindre de nouveaux sommets sur le plan de l'intimidation et de la détresse. Elle engendre également des conséquences qui lui sont propres, comme le bris de l'intimité, l'atteinte à la vie privée et un sentiment d'impuissance face à un agresseur parfois anonyme [10]. Enfin, la nature même des moyens de communication utilisés amplifie les conséquences de la cyberintimidation.

I.3.3. Le sexisme

Le sexisme est une discrimination fondée sur le sexe ou sur le genre d'une personne. Le sexisme est lié aux préjugés et au concept de stéréotype et de rôle de genre, pouvant comprendre la croyance qu'un sexe ou qu'un genre serait intrinsèquement supérieur à l'autre [11]. De façon extrême, il peut encourager le harcèlement sexuel, le viol ou toute autre forme de violence sexuelle. Le sexisme évoque également la discrimination de genre sous la forme des inégalités hommes-femmes (cible principale est la femme).

Le sexisme c'est une façon de se basée sur le fait que certaines personnes, le plus souvent des femmes, sont inférieures en raison de leur sexe. Il est à la base des inégalités entre les femmes et les hommes, il affecte les femmes et les filles de manière disproportionnée. Les actes individuels de sexisme peuvent sembler bénins, mais ils créent un climat d'intimidation, de peur et d'insécurité. Cela conduit à l'acceptation de la violence, principalement envers les femmes et les filles [12]. Le sexisme est dangereux et engendre des sentiments de dévalorisation, d'autocensure, des changements de comportement et une détérioration de la santé.

I.3.4. Le racisme

Le racisme est une attitude d'hostilité ou de mépris systématique à l'égard de certaines personnes ou groupes de personnes à cause de leur nationalité, leur couleur de peau, leur ascendance, leur origine nationale ou leur origine ethnique [13].

Les réseaux sociaux sont un endroit où les gens du monde entier se rencontre, il est donc évident qu'ils soient devenus un espace pour la création de mouvements et le partage d'idées. Toutefois, les médias sociaux sont de plus en plus souvent utilisés pour amplifier les profondes fissures historiques qui divisent les communautés et servent d'outil pour diffuser le racisme.

Les personnes de couleur sont ciblées par des efforts organisés de désinformation utilisant les technologies numériques. Nous avons identifié quatre principaux discours racistes qui opèrent sur les réseaux sociaux : les stéréotypes, les boucs émissaires, les allégations de racisme à l'envers et les chambres d'écho [14]. Par exemple, le tweet de Trump de mars 2020 implique un bouc émissaire en ce sens qu'il blâme les Chinois et la Chine pour la propagation du coronavirus aux États-Unis, exonérant ainsi son gouvernement de toute responsabilité. Pour lutter contre le racisme sur les réseaux sociaux, il faut comprendre que les utilisateurs qui diffusent de la désinformation raciste le font différemment, aggravant parfois plusieurs formes de racisme dans un seul message [14].

I.3.5. Contenu abusif et violent

Il existe plusieurs types de violence sur les réseaux sociaux, comme : le crime, le terrorisme, la violation des droits de l'homme, l'opinion politique, la crise, les accidents et les conflits. L'utilisation d'un langage manipulateur qui génère la peur, la culpabilité, l'humiliation, l'éloge, le blâme, le devoir, l'obligation ou la punition est souvent la cause d'une communication violente [1].

Un langage abusif est extrêmement grossier, violent et insultant. Il est utilisé la plupart des temps pour offenser ou mettre mal à l'aise un individu ou un groupe de personnes. De nos jours, plus de la moitié des cas de harcèlement en ligne se produisent sur les plateformes des réseaux sociaux. Une populaire forme spécifique de harcèlement en ligne est l'utilisation du langage abusive, ainsi une déclaration abusive ou violente est envoyée toutes les 30 secondes à travers le monde. De ce fait, l'utilisation du langage abusif sur les réseaux sociaux contribue à un stress mental ou émotionnel [15].

Le langage abusif peut être synthétisé dans une typologie à deux volets qui considère si le contenu abusive est dirigé contre une cible, et la mesure dans laquelle elle est explicite. En commençant par les cibles, un contenu abusive peut soit être dirigé vers un individu ou une entité spécifique soit il peut être utilisé vers un autre généralisé [16].

L'autre dimension est la mesure dans laquelle le langage abusif est explicite ou implicite. Un langage abusif explicite est celui qui est sans ambiguïté dans son potentiel d'être abusif, par exemple un langage qui contient des insultes raciales ou homophobes. Un langage abusif implicite est celui qui n'implique pas ou ne dénote pas immédiatement un abus. Ici, la vraie nature est souvent obscurcie par l'utilisation de termes ambigus, le sarcasme, l'absence de

Blasphèmes ou de termes haineux, et d'autres moyens, ce qui rend généralement plus difficile à détecter à la fois par les annotateurs et les approches d'apprentissage automatique [16].

I.4. Effet négatif du langage offensif

De nos jours, la plupart des gens se connecte sur les réseaux sociaux, tout le monde peut être victime des messages offensants. D'autre parvient à surmonter et passer le cap vis-à-vis des commentaires offensifs par contre d'autre, ça les conduit aux désastres. Les effets négatifs du langage offensif sont courant dans la société telle que la dépression, le suicide, le trouble mental, etc.

I.4.1. La dépression

La santé mentale d'une personne peut être affectée par les réseaux sociaux surtout que les gens ont tendance à passer plus de temps sur les réseaux sociaux comme Facebook, Twitter, Instagram et autres. Ainsi, certains internautes sont souvent intimidés par d'autres utilisateurs des réseaux sociaux avec des insultes, menaces, désaccords et harcèlement dans les forums de discussion et les sections de commentaires des médias sociaux. Cependant, la lecture des commentaires offensifs en ligne provoque la dépression chez les gens.

La dépression est un trouble du comportement dans lequel l'humeur est pathologiquement figée dans la tristesse ou la douleur, elle se caractérise aussi par un sentiment de désespoir, une perte de motivation sur les activités, une diminution du sentiment de plaisir, des troubles alimentaires et du sommeil, des pensées morbides et l'impression de ne pas avoir de valeur en tant qu'individu. La tristesse d'une personne dépressive est intense et n'est pas diminuée par des circonstances extérieures [17]. La dépression survient généralement sous forme de périodes dépressives qui peuvent durer des semaines, des mois voire des années. Selon l'intensité des symptômes, la dépression sera qualifiée de légère, modérée ou majeur. Dans les cas les plus graves, la dépression peut conduire au suicide. La dépression affecte l'humeur, les pensées et le comportement, mais aussi le corps [18].

La dépression est une maladie qui peut toucher chacun d'entre nous (quels que soient son âge, son sexe, son niveau social…). Contrairement à certaines idées reçues, elle ne relève ni d'une « fatalité », ni d'une faiblesse de caractère. La volonté seule ne suffit pas pour en sortir, notamment parce que la maladie provoque un sentiment de dévalorisation de soi et des pensées négatives. La dépression entraîne aussi un ralentissement dans tous les registres de la vie quotidienne : vie affective, fonctionnement intellectuel, forme physique, mécanismes vitaux et corporels [19].

I.4.1.1. Complications liées à la dépression

Les relations peuvent être tendues par la dépression, entraînant la perte d'amitiés, la rupture de liens et des divorces. En effet, la dépression peut entraîner des sentiments de solitude et d'aliénation extrêmes, ce qui peut rendre toute relation difficile. La fatigue débilitante et le désespoir sont des symptômes courants de la dépression, qui peuvent être très pénibles pour deux personnes dans une relation [1]. Il existe plusieurs complications possibles liées à la dépression :

- la récidive de dépression ;

Elle est fréquente puisqu'elle concerne 50 % des personnes ayant vécu une dépression. La prise en charge diminue considérablement ce risque de récidive.

- la persistance de symptôme résiduel ;

Il s'agit de cas où la dépression ne se guérit pas entièrement et où même après l'épisode dépressif, persistent des signes de dépression.

- le passage à la dépression chronique ;
- le risque suicidaire [18].

La dépression est aussi la première cause de suicide : près de 70 % des personnes qui décèdent par suicide souffraient d'une dépression, le plus souvent non diagnostiquée ou non traitée [19].

I.4.1.2. Les troubles associés à la dépression

La dépression a des liens physiques ou psychologiques avec d'autres problèmes de santé :

- anxiété ;
- dépendance ;

Alcoolisme, abus de substances telles que le cannabis, l'ecstasy, la cocaïne, dépendance à certains médicaments comme les somnifères ou les tranquillisants.

- augmentation du risque de certaines maladies.

Notamment maladies cardiovasculaires et de diabète. En effet, la dépression est associée à un risque plus élevé de problèmes cardiaques ou d'accidents vasculaires cérébraux. Par ailleurs, le fait de souffrir de dépression pourrait accélérer légèrement l'apparition du diabète chez les personnes déjà à risque [18].

I.4.2. Le suicide

La crise suicidaire est une période critique, marquée par un envahissement des émotions, par de grandes difficultés pour se concentrer, par le sentiment profond d'avoir tout essayé et que rien ne marche pour être soulagé. Le vécu d'impuissance est majeur. Cette crise suit souvent un processus qui comporte plusieurs étapes : la personne a d'abord des « flashs » (visions brèves qui donnent l'impression de devenir fou), puis des idées de suicide plus ou moins fréquentes et intenses contre lesquelles elle va lutter mais qui peuvent éventuellement l'envahir ; elle risque alors de passer à l'étape de l'intention (prise de décision), de la planification (recherche du moyen, du lieu, des circonstances et du moment) et de la mise en œuvre de son suicide [19].

Le suicide peut être considéré comme l'un des problèmes de santé sociale les plus graves de la société moderne. De nombreux facteurs peuvent conduire au suicide par exemple :

- Des problèmes personnels, tels que le désespoir, une anxiété sévère, l'alcoolisme ou l'impulsivité ;
- Des facteurs sociaux, comme l'isolement social et la surexposition aux décès ;
- Des événements négatifs de la vie, y compris des événements traumatisants, une maladie physique, des troubles affectifs et des tentatives de suicide antérieure [20].

Les idées de suicide sont fréquentes dans la dépression (elles font d'ailleurs partie des symptômes de la maladie), elles méritent dans tous les cas d'être signalées à un professionnel de santé afin d'en parler et de les désamorcer. Il est important de savoir que :

- les personnes suicidaires ne veulent pas nécessairement mourir mais souhaitent plutôt mettre fin à une souffrance devenue insupportable ;
- l'immense majorité des personnes en proie à des idées de suicide ne feront pas de tentative.

I.4.3. Les troubles relationnels

Le trouble relationnel est défini comme [21]:

- Une difficulté à développer et maintenir des relations ;
- Une impression de ne pas avoir de confident ou un ami de confiance ;
- Une difficulté récurrente à partager son avis ou recevoir celui des autres.

Ainsi, les commentaires offensant ou abusif sur les plateformes des réseaux sociaux produisent ce genre de sentiment d'être détester par autrui, cela est dû à l'attachement que les victimes des messages offensant ont envers les avis d'autrui et ceci peut provoquer un cas de trouble relationnel, sentir qu'on est pas aimé, qu'on a pas d'ami jusqu'à abandonner toute sorte de relation dans la société et manquer toute sorte de confiance.

I.5. Modélisation hiérarchique du contenu offensif

I.5.1. Détection du langage offensive

Comme le contenu offensant est devenu omniprésent dans médias sociaux, il y a eu beaucoup de recherches pour identifier les messages potentiellement offensifs. Une des stratégies les plus courantes pour résoudre le problème est de former des systèmes capables de reconnaître le contenu offensif, qui peut alors être supprimé ou mis de côté pour la modération humaine.

Ainsi, la première approche est d'identifier si le type de message sur les réseaux sociaux est offensif ou non offensif :

- Non offensif : messages qui ne contiennent pas de contenu offensent ou de blasphème ;
- Offensif : Messages contenant n'importe quel langage inacceptable (blasphème) ou une infraction ciblée, qui peut être voilée ou directe. Cela inclut les insultes, les menaces et les messages contenant un langage profane ou des jurons.

I.5.2. Détection automatique du langage offensive

Dans cette section, nous présentons une revue de la littérature sur les travaux antérieurs réalisés en détection du langage offensant sur la langue et le dialecte arabe puisque notre travail et surtout notre base de données est basé que sur la langue arabe (en particulier l'arabe algérien).

I.5.2.1. Langage arabe

Une approche de détection de langage abusif sur les médias sociaux arabes a été proposée, où deux ensembles de données ont été introduits [22]. Le premier contient 1 100 tweets étiquetés manuellement et le second contient 32 000 commentaires d'utilisateurs que les modérateurs d'un site populaire d'actualité arabe ont jugés inappropriés. Les auteurs ont proposé une approche statistique basée sur une liste de mots abusifs, et les résultats produits étaient d'environ 60 % du score F1.

La détection de la cyberintimidation dans le contenu arabe a été réalisée dans [23], dans lequel les auteurs ont introduit une approche axée sur la prévention des attaques de la cyberintimidation. L'approche utilise le Traitement du langage naturel plutôt appelé Natural Language Processing (NLP) pour identifier et traiter les mots arabes, et les techniques d'apprentissage automatique pour classer le contenu d'intimidation.

Un ensemble de données sur le discours de haine en arabe avec 9 300 tweets annotés a été proposé par Raghad et AlKhalifa [24]. Les auteurs ont expérimenté plusieurs modèles d'apprentissage en profondeur et l'apprentissage automatique pour détecter les discours de haine dans les tweets en arabe. Les résultats ont montré que CNN GRU a produit les meilleures performances (0,79 de F1score). Une approche multitâche couvrant la détection du langage offensif arabe et du discours de haine à l'aide de DL, l'apprentissage par transfert et l'apprentissage multitâche a été proposée dans [25].

I.5.2.2. Dialecte Arabe

Husain a utilisé différentes approches ML et un classificateur d'ensemble pour traiter l'identification de la langue offensif de l'arabe dialectal [26]. L'étude menée a montré un impact intéressant du prétraitement sur une telle tâche, ainsi que les performances du classificateur d'ensemble par rapport aux algorithmes ML simples.

Un autre travail axé sur les discours de haine tunisiens et les discours abusifs a été proposé afin de créer un ensemble de données de référence (c'est-à-dire 6k de tweets) de contenus toxiques tunisiens en ligne [27]. Les auteurs ont testé deux approches ML (c'est-à-dire NB et SVM), où le classificateur NB a surpassé le SVM (92,9 % de précision dans la classification binaire).

La détection des discours de haine contre les femmes dans la communauté algérienne a été abordée dans [28], pour laquelle les auteurs ont utilisé *word2vec* et *FastText* avec des fonctionnalités différentes. D'après les résultats expérimentaux, *FastText* a produit un résultat prometteur contrairement à *word2vec*.

.

I.5.3. Catégorisation du langage offensif

- Insulte ciblée : ce sont des publications contenant des insultes/menaces à l'encontre d'un individu, d'un groupe ou d'autres personnes.
- Non ciblé : ce sont des messages contenant des grossièretés et des jurons non ciblés, messages avec les blasphèmes généraux qui sont pas ciblés, mais qui contiennent un langage inacceptable [29].

I.5.4. Identification des cibles du langage offensif

Les cibles des messages offensifs sont :

- Individu : Messages ciblant un individu, cela peut être une personne célèbre, une personne nommé ou un participant anonyme à la conversation. Insultes et menaces visant les individus sont souvent définis comme du cyber harcèlement.
- Groupe : messages ciblant un groupe de personnes considérées comme une unité en raison de la même origine ethnique, sexe ou orientation sexuelle, affiliation politique, croyance religieuse, race ou autre caractéristique commune. Beaucoup d'insultes et menaces visant un groupe est généralement compris comme un discours de haine.
- Autre : La cible des messages offensifs n'appartient à aucun des deux précédentes catégories (par exemple, une organisation, une situation, un événement ou un problème) [29].

I.6. Perspective contre le langage offensif

L'utilisation d'un langage offensif est un problème courant de comportement abusif sur les réseaux sociaux. Dans le passé, divers travaux ont attaqué ce problème en utilisant différents modèles d'apprentissage pour détecter les comportements abusifs. La plupart de ces travaux partent du fait qu'il suffit de filtrer l'ensemble des messages offensif [30]. Cependant, les utilisateurs qui prévoient de poster un message offensif, si l'on pouvait non seulement avertir qu'un contenu est offensif et qu'il sera bloqué, mais aussi proposer une version polie du message qui peut être republiée, cela pourrait encourager de nombreux utilisateurs à changer d'avis et à éviter les injures.

Une nouvelle manière de traiter le problème du langage offensif sur les médias sociaux consiste à utiliser des techniques de transfert de style pour traduire les phrases offensives en phrases non offensives. Un simple encodeur-décodeur avec attention serait suffisant pour créer un traducteur raisonnable si un grand corpus parallèle est disponible [30].

Autre perspective serait de faire un long travail de la classification des textes sur toutes les langues du monde si possible et de classifier tous les contenus offensifs ou abusifs afin de les bloquer sur toutes les plateformes des réseaux sociaux.

I.7. Conclusion

Dans ce chapitre, nous avons parlé de la langue et du dialecte arabe en particulier le dialecte algérien et nous avons défini le langage offensif sur les réseaux sociaux, présenté les différents types de langage offensif en ligne et leurs impacts négatifs sociaux sur les individus. Nous avons aussi présenté une revue de la littérature sur les travaux antérieurs de la détection automatique de la langue arabe. Dans le chapitre suivant, nous présenterons la méthodologie adoptée pour identifier automatiquement le langage offensif sur l'arabe dialectal algérien.

Chapitre II
Méthodologie

II.1. Introduction

La classification des textes devient de plus en plus importante à mesure que l'on utilise les plateformes en ligne. Une telle tâche peut être réalisée en utilisant la classification automatique ou manuelle. Dans ce chapitre, nous détaillerons la catégorisation automatique des textes et les classificateurs utilisés pour la classification automatique, ainsi que notre démarche sur la méthodologie du projet et aussi un certain nombre de technique d'optimisation pour l'obtention d'un meilleur résultat.

II.2. Approches de catégorisation des textes

La catégorisation des textes peut se faire de deux manières : manuellement ou automatiquement. Un annotateur humain maitrise le contenu du texte et le catégorise correctement et manuellement. Cette procédure peut donner d'excellents résultats, mais elle prend beaucoup de temps et est coûteuse. Cependant, les approches guidées par l'intelligence artificielle sont utilisées pour catégoriser automatiquement les textes de manière plus rapide, plus économique et plus précise.

II.3. La catégorisation automatique des textes

II.3.1. L'apprentissage automatique

L'apprentissage automatique ou machine learning (*ML*) est une sous-catégorie de l'intelligence artificielle qui permet aux machines d'apprendre sans avoir été au préalablement programmées spécifiquement pour ça. Il est explicitement lié au Big Data[1], du moment que, pour apprendre et se développer, les ordinateurs ont besoin de flux de données à analyser, sur lesquelles s'entraîner et bien fonctionner à la fin [31]. Le développement d'un modèle de *ML* repose sur quatre étapes principales :

La première étape consiste à sélectionner et à préparer un ensemble de données d'entraînement. Ces données seront utilisées pour nourrir le modèle de *ML* pour apprendre à résoudre le problème pour lequel il est conçu. Les données peuvent être étiquetées, afin d'indiquer au modèle les caractéristiques qu'il devra identifier. Elles peuvent aussi être non étiquetées, et le modèle devra repérer et extraire les caractéristiques récurrentes de lui-même. Dans les deux cas, les données doivent être soigneusement préparées et organisées. Dans le cas contraire, l'entraînement du modèle de ML risque d'être biaisé. Les résultats de ses futures prédictions seront directement impactés [32].

La deuxième étape consiste à sélectionner un algorithme à exécuter sur l'ensemble de données d'entraînement. Le type d'algorithme à utiliser dépend du type et du volume de données d'entraînement et du type de problème à résoudre [32].

[1] Données massives

La troisième étape est l'entraînement de l'algorithme. Il s'agit d'un processus itératif. Des variables sont exécutées à travers l'algorithme et les résultats sont comparés avec ceux qu'il aurait dû produire. Les " poids " et le biais peuvent ensuite être ajustés pour accroître la précision du résultat. On exécute ensuite de nouveau les variables jusqu'à ce que l'algorithme produise le résultat correct la plupart du temps. Ainsi, l'algorithme entraîné, est le modèle de *ML* [32].

La quatrième et dernière étape est l'utilisation et l'amélioration du modèle. On utilise le modèle sur de nouvelles données, dont la provenance dépend du problème à résoudre. Par exemple, un modèle de Machine Learning conçu pour détecter les spams sera utilisé sur des emails [32].

Figure 2.1. Processus de la machine learning [34]

II.3.2. L'apprentissage profond

L'apprentissage profond ou deep learning (*DL*) est une sorte d'intelligence artificielle dérivé du *ML* qui passe par le déploiement d'un réseau de neurones artificiel préalablement entraîner.

Le *DL* est un système avancé basé sur le cerveau humain, qui comporte un vaste réseau de neurones artificiels. Ces neurones sont interconnectés pour traiter et mémoriser des informations, comparer des problèmes ou situations quelconques avec des situations similaires passées, analyser les solutions et résoudre le problème de la meilleure façon possible [33].

Il s'appuie sur un réseau de neurones artificiels s'inspirant du cerveau humain. Ce réseau est composé de dizaines jusqu'à de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente. Le système apprendra par exemple à reconnaître les lettres avant de s'attaquer aux mots dans un texte, ou détermine s'il y a un visage sur une photo avant de découvrir de quelle personne il s'agit. À chaque étape, les mauvaises réponses sont éliminées et renvoyées vers les niveaux en amont pour ajuster le modèle mathématique. Au fur et à mesure, le programme réorganise les informations en blocs plus complexes et lorsque ce modèle est par la suite appliqué à d'autres cas, il est normalement capable de reconnaître un chat sans que personne ne lui ait jamais indiqué qu'il n'ait jamais appris le concept de chat. Les données de départ sont essentielles : plus le système accumule d'expériences différentes, plus il sera performant [33].

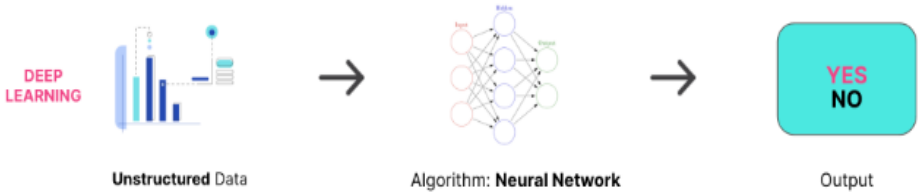

Figure 2.2. Processus de deep learning [34]

Récemment, les modèles de *DL* ont obtenu des résultats remarquables dans divers domaines du traitement du langage naturel, tels que la classification de textes. Ainsi, deux éléments sont à prendre en compte pour utiliser du *DL* dans un projet : la puissance calculatrice des machines et/ou le volume de données doit être trop élevés.

II.3.3. Traitement du langage naturel

Il est appelé natural langage processing en anglais (ou *NLP*), et est une branche de l'intelligence artificielle qui vise à permettre aux ordinateurs de comprendre et d'interpréter le langage humain. Le problème de l'interprétation du langage humain est qu'il ne s'agit pas d'un ensemble de règles ou de données binaires qui peuvent être introduites dans le système et que comprendre le contexte d'une conversation ou lire entre les lignes est un jeu tout à fait différent. Cependant, avec les progrès récents du *ML*, le *DL* avec l'aide des réseaux neuronaux et des modèles faciles à utiliser en python, nous avons ouvert les portes pour coder notre façon de faire comprendre aux ordinateurs le langage humain complexe.

II.4. La tâche de classification de texte

L'identification du langage offensif est une tâche de classification de texte par *NLP* dont l'objectif est de modérer et de réduire le contenu toxique des médias sociaux. La classification de texte permet aussi de concevoir des algorithmes appropriés pour permettre aux ordinateurs d'extraire des caractéristiques et de classer des textes automatiquement.

Les algorithmes de classification de texte sont au cœur d'une variété de systèmes logiciels qui traitent les données textuelles à grande échelle. Les logiciels de messagerie électronique utilisent la classification de texte pour déterminer si le courrier entrant doit être envoyé dans la boîte de réception ou filtré dans le dossier spam. Les forums de discussion utilisent la classification de texte pour déterminer si les commentaires doivent être signalés comme inappropriés. Il s'agit là de deux exemples de classification de sujets, qui consistent à classer un document textuel dans l'un des ensembles prédéfinis de sujets. Dans de nombreux problèmes de classification de sujets, cette catégorisation est basée principalement sur des mots-clés dans le texte [35].

Les tâches de classification des textes commencent par la fourniture d'un ensemble d'apprentissage : documents et catégories (étiquettes) à l'algorithme d'apprentissage automatique. Une fois le modèle formé, il peut être utilisé pour catégoriser de nouveaux exemples [36].

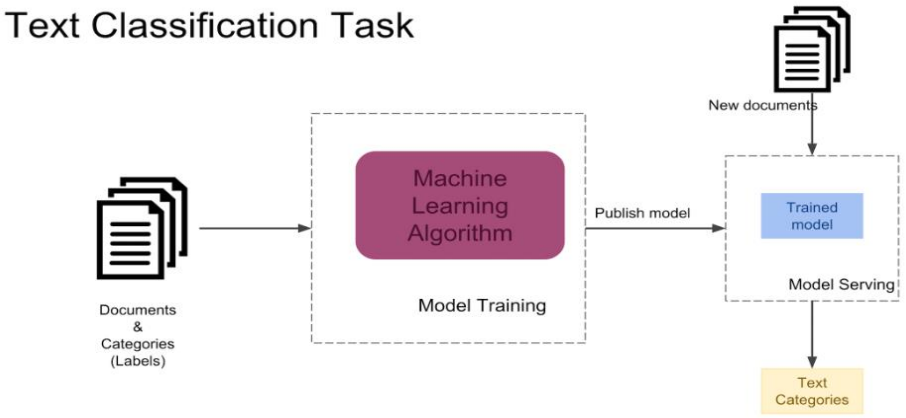

Figure 2.3. Tache de classification des textes [36]

II.5. Les algorithmes de classification

II.5.1. Réseaux neuronaux convolutifs

Dans le DL, le réseau neuronaux convolutifs (convolutional neural networks ou *CNN*) est l'algorithme le plus célèbre et le plus couramment utilisé. Le principal avantage du *CNN* par rapport à ses prédécesseurs est qu'il identifie automatiquement les caractéristiques pertinentes sans supervision humaine. Les *CNN* sont des réseaux de neurones avec une ou plusieurs couches convolutives qui sont principalement utilisés pour le traitement des images, la classification, le traitement de la parole, etc. Ils sont également été utilisés dans le traitement du langage naturel et la reconnaissance vocale. La structure des *CNN* a été inspirée par les neurones du cerveau humain et animal, à l'instar d'un réseau neuronal classique [37].

Les *CNN* sont des architectures profondes très complexes et largement utilisées, qui sont extrêmement performantes dans les domaines où les données d'apprentissage sont nombreuses avec de grandes quantités d'ensembles de données d'entraînement, et ont connu des succès intempestifs dans les tâches de classification des chiffres.

La convolution et la mise en commun sont deux procédures essentielles qui sont toujours incluses dans les *CNN*. Le processus de convolution avec plusieurs filtres est capable d'extraire des caractéristiques du jeu de données tout en préservant leurs informations spatiales. La convolution est une opération linéaire dans laquelle une collection de poids est multipliée par l'entrée. La multiplication se fait entre un tableau de données d'entrée et un tableau bidimensionnel de poids, appelé filtre ou noyau [38].

Le filtre est plus petit que les données d'entrée, et le produit scalaire est utilisé pour multiplier un morceau de taille filtre de l'entrée avec le filtre. Il est intentionnel d'utiliser un filtre qui est plus petit que l'entrée, car cela permet au même filtre d'être multiplié par le tableau d'entrée plusieurs fois à différentes positions dans l'entrée. De gauche à droite, de haut en bas, le filtre est appliqué systématiquement à chaque section qui se chevauche ou à chaque morceau de taille filtre des données d'entrée [38].

Une valeur unique est obtenue en multipliant une fois le filtre avec le tableau d'entrée. Le résultat de l'application du filtre au tableau d'entrée est un tableau bidimensionnel de valeurs de sortie qui indiquent le filtrage de l'entrée. Ce tableau de sortie bidimensionnel est appelé "*carte de caractéristiques*" ou "*feature map*".

La mise en commun ou *pooling*, également appelée sous-échantillonnage, est une technique permettant de réduire la dimensionnalité des cartes de caractéristiques créées par la procédure de convolution. La couche de mise en commun travaille sur chaque carte de caractéristiques individuellement pour construire un nouvel ensemble de cartes de caractéristiques mises en commun. La mise en commun est similaire à l'application d'un filtre aux cartes de caractéristiques dans la mesure où elle implique la sélection d'un processus de mise en commun. Le processus de mise en commun ou filtre a une dimension plus petite que la carte d'entités. Il existe plusieurs types d'opérations de regroupement : le regroupement *Max* prend l'élément le plus grand, le regroupement *Sum* extrait la somme de tous les éléments et le regroupement *Average* calcule la valeur moyenne de chaque patch[2] de la carte de caractéristiques [38].

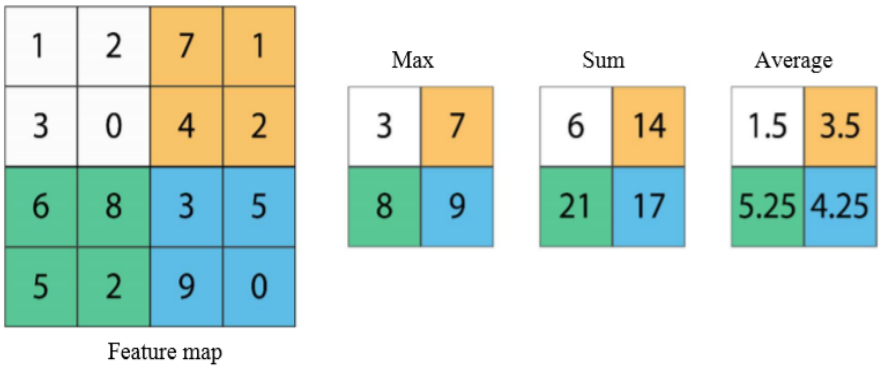

Figure 2.4. Differentes opérations de pooling [38]

II.5.2. Réseau de neurones récurrent

Le réseau de neurones récurrent ou recurrent neural network (*RNN*) est un algorithme couramment utilisé et familier dans la discipline de *DL*. Les *RNN* sont principalement appliqués dans le domaine du traitement de la parole et dans des contextes de *NLP*. Contrairement aux réseaux classiques, les *RNN* utilisent des données séquentielles dans le réseau. Puisque la structure intégrée dans la séquence des données fournit des informations précieuses. Cette caractéristique est fondamentale pour une série d'applications différentes. Par exemple, il est important de comprendre le contexte de la phrase afin de déterminer la signification d'un mot spécifique de la phrase [37].

[2] Pièce

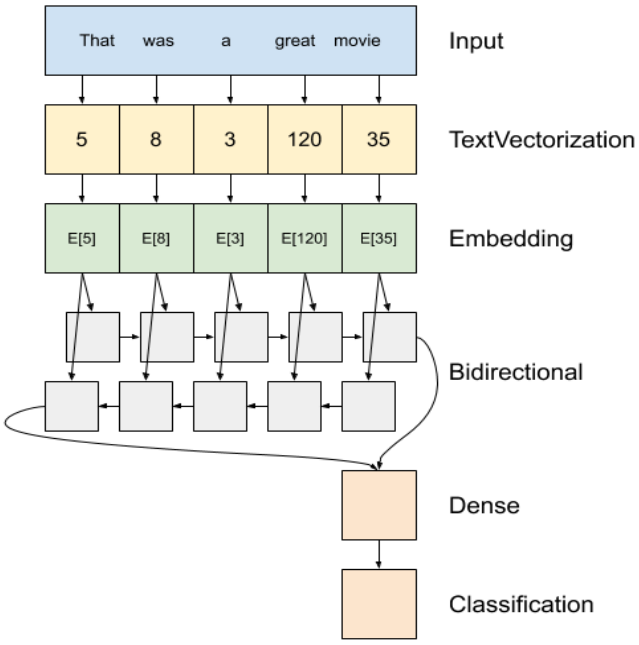

Figure 2.5. Schéma du modèle RNN [39]

La première couche est l'encoder, qui convertit le texte en une séquence d'indices de jeton. Après l'encodeur se trouve une couche d'intégration. Une couche d'incorporation stocke un vecteur par mot. Lorsqu'il est appelé, il convertit les séquences d'indices de mots en séquences de vecteurs. Ces vecteurs sont entraînables. Après apprentissage (sur suffisamment de données), les mots ayant des significations similaires ont souvent des vecteurs similaires.

Le *RNN* traite l'entrée de séquence en itérant à travers les éléments. Les RNN transmettent les sorties d'un pas de temps à leur entrée au pas de temps suivant. Ainsi,

- Le principal avantage d'un *RNN* bidirectionnel est que le signal du début de l'entrée n'a pas besoin d'être traité tout au long de chaque pas de temps pour affecter la sortie.
- Le principal inconvénient d'un *RNN* bidirectionnel est que vous ne pouvez pas diffuser efficacement les prédictions car des mots sont ajoutés à la fin.

Après, le *RNN* a converti la séquence à un seul vecteur les deux couches denses pour faire un peu de traitement final et convertir cette représentation vectorielle à un seul logit comme la sortie de classification [39].

II.5.3. Mémoire à court terme

L'algorithme de mémoire à court terme (Long-Short Term Memory ou *LSTM*). La *LSTM* est un type de réseau neuronal récurrent, mais elle est meilleure que les réseaux neuronaux récurrents traditionnels en termes de mémoire [40]. Ayant une bonne emprise sur la mémorisation de certains modèles, les *LSTM* ont des performances assez supérieures. Comme tous les autres réseaux neuronaux, les *LSTM* peuvent avoir plusieurs couches cachées et

lorsqu'ils traversent chaque couche, les informations pertinentes sont conservées et toutes les informations non pertinentes sont éliminées dans chaque cellule.

Une bonne raison d'utiliser *LSTM* est qu'il est efficace pour mémoriser les informations importantes. Si nous regardons d'autres techniques de classification de réseau non neuronal, elles sont formées sur plusieurs mots comme entrées séparées qui sont juste des mots n'ayant pas de signification réelle comme une phrase, et tout en prédisant la classe, il donnera la sortie selon les statistiques et non selon la signification. Cela signifie que chaque mot est classé dans l'une des catégories. Ce n'est pas le cas avec les *LSTM*, dans *LSTM* nous pouvons utiliser une chaîne de mots multiples pour trouver la classe à laquelle elle appartient. Ceci est très utile pour le traitement du langage naturel. Si nous utilisons des couches appropriées d'incorporation et d'encodage dans le *LSTM*, le modèle sera capable de trouver la signification réelle de la chaîne d'entrée et donnera la classe de sortie la plus précise [40].

L'architecture du modèle C-LSTM est illustrée à la figure 2.6, qui se compose de deux éléments principaux : le réseau neuronal convolutif (CNN) et le réseau de mémoire à long terme (LSTM).

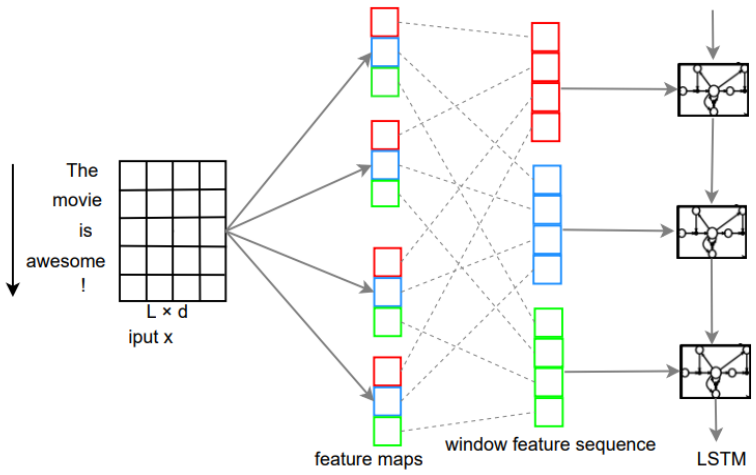

Figure 2.6. Architecture de C-LSTM pour la modélisation de phrases [41]

Les blocs de la même couleur dans la couche *feature map* et la couche *window feature sequence* correspondent aux caractéristiques de la même fenêtre. Les lignes pointillées relient la caractéristique d'une fenêtre à la carte de caractéristique source. La sortie finale de l'ensemble du modèle est la dernière unité cachée de *LSTM* [41].

II.5.4. Machines à vecteurs de support

Les machines à vecteurs de support ou (support-vector machine ou *SVM*) sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination et de régression. Les *SVM* sont une famille d'algorithmes d'apprentissage automatique qui permettent de résoudre des problèmes tant de classification que de régression ou de détection d'anomalie [42]. C'est un outil de prédiction de classification et de régression qui utilise la

théorie de *ML* pour améliorer la précision de la prédiction, tout en évitant automatiquement le sur-ajustement aux données.

C'est un modèle de *ML* supervisé qui utilise des algorithmes de classification pour les problèmes de classification à deux groupes. Après avoir donné à un modèle *SVM* des ensembles de données d'entraînement étiquetées pour chaque catégorie, il est capable de classer un nouveau texte [43].

Par rapport à des algorithmes plus récents comme les réseaux neuronaux, ils présentent deux avantages principaux : une vitesse plus élevée et de meilleures performances avec un nombre limité d'échantillons (des milliers). Cela rend l'algorithme très adapté aux problèmes de classification de textes, pour lesquels il est courant d'avoir accès à un ensemble de données comprenant au maximum quelques milliers d'échantillons étiquetés [44].

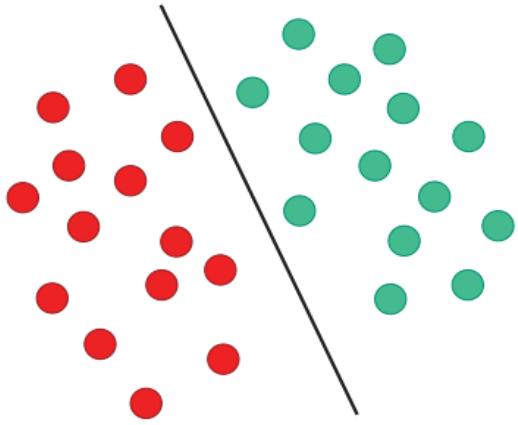

Figure 2.7. Séparation linéaire dans l'espace des données d'entrée [44]

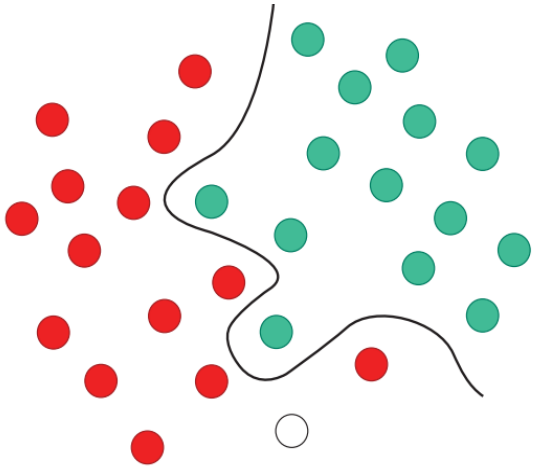

Figure 2.8. Séparation non linéaire dans l'espace des données d'entrée [44]

Comme le montre la figure 2.7, un plan de décision convient pour séparer des objets ayant des appartenances de classe différentes. La ligne de séparation crée une barrière où dans la partie droite tous les objets sont verts, et dans la partie gauche tous les objets sont rouges.

Tout nouvel objet qui tombe à droite de la ligne de séparation est classé en vert, tandis que tout objet qui tombe à gauche est classé en rouge [44].

En revanche, la plupart des travaux de classification ne sont pas aussi simples et des structures plus sophistiquées sont souvent nécessaires pour obtenir les meilleurs résultats. Comme le montre la figure 2.8, une séparation complète des éléments vert et rouge nécessiterait une courbe, qui est plus compliquée qu'une ligne. Les classificateurs hyperplans résolvent les problèmes de classification en traçant les lignes de séparation pour distinguer les éléments appartenant à diverses classes [44].

II.5.5. Classificateur naïve de Bayes

Les classificateurs naïves de Bayes (Naive Bayes ou *NB*) sont une famille d'algorithmes de classificateur linéaire basés sur le théorème de Bayes reconnus pour leur simplicité tout en étant extrêmement efficaces, qui partagent tous un principe commun, c'est à dire chaque paire de caractéristiques à classer est indépendante de l'autre.

En raison de l'hypothèse d'indépendance, le *NB* n'a pas besoin d'apprendre toutes les corrélations possibles entre les caractéristiques. Si *N* est le nombre de caractéristiques, un algorithme général nécessite d'analyser *2N* interactions possibles entre les caractéristiques, alors que le *NB* n'a besoin que de l'ordre de *N* points de données. Ainsi, les classificateurs *NB* peuvent apprendre plus facilement à partir de petits ensembles de données d'apprentissage en raison de l'hypothèse d'indépendance des classes. En même temps, le *NB* n'est pas affecté par la malédiction de la dimensionnalité [45].

NB est une méthode très rapide. Elle dépend des probabilités conditionnelles, qui sont faciles à mettre en œuvre et à évaluer. Par conséquent, elle ne nécessite pas de processus itératif. *NB* prend en charge la classification binaire ainsi que la classification multinomiale. Le *NB* suppose que les caractéristiques sont indépendantes entre elles, mais cette hypothèse ne se vérifie pas toujours. Malgré cela, le *NB* donne de bons résultats lorsqu'il est appliqué à des textes courts comme les tweets. Pour certains jeux de données, le *NB* peut battre d'autres classificateurs en utilisant la sélection de caractéristique [45]. Ainsi, il existe plusieurs variantes du classificateur *NB* :

- **NB optimal :** Ce classificateur sélectionne la classe dont la probabilité postérieure d'occurrence est la plus élevée. Il est idéal comme son nom l'indique, mais parcourir toutes les possibilités disponibles est assez lent et prend du temps.
- **NB Gaussian :** Gaussian Bayes est basé sur la distribution normale. Il accélère considérablement la recherche, et l'erreur n'est que deux fois plus élevée que dans Optimal Bayes sous certaines circonstances non strictes.
- **NB Multinomial :** Il est généralement utilisé pour résoudre les problèmes de classification de documents. Il fonde ses décisions sur des critères discrets, tels que la fréquence des mots dans le document.
- **NB Bernoulli :** Dans Bernoulli, les prédicteurs sont des variables booléennes. Par conséquent, les paramètres utilisés pour prédire la variable de classe ne peuvent avoir que des valeurs de type oui ou non, comme l'existence ou non d'un mot dans le texte.

II.5.6. Gradient descent stochastique

L'algorithme de gradient descent stochastique (Stochastic Gradient Descent ou SGD) pourrait induire en erreur beaucoup de personnes en leur faisant croire que SGD est un classificateur, alors que *SGD* est un classificateur linéaire (*SVM*) optimisé par la *SGD*. Il s'agit de deux concepts différents, c'est-à-dire que le *SGD* est une méthode d'optimisation alors que le *SVM* est un algorithme/modèle de ML. On peut considérer qu'un modèle de ML définit une fonction de perte et que la méthode d'optimisation la minimise ou la maximise [46].

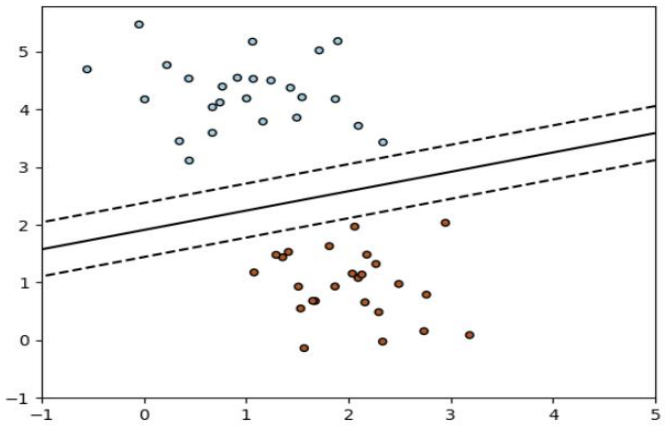

Figure 2.9. SVM linéaire optimisé avec SGD [47]

SGD est une approche simple mais très efficace pour ajuster des classificateurs sous des fonctions de pertes convexes comme le *SVM* (linéaire). Bien que la *SGD* existe depuis longtemps dans la communauté de *ML*, elle n'a reçu que récemment une attention considérable dans le contexte de l'apprentissage à grande échelle [47]. Le classificateur *SGD* met essentiellement en œuvre une routine d'apprentissage *SGD* ordinaire prenant en charge diverses fonctions de perte et pénalités pour la classification. Il a été appliqué avec succès aux problèmes de *ML* à grande échelle et à faible densité souvent rencontrés dans la classification de textes et le *NLP* [47]. Les principaux avantages du *SGD* sont :

- l'efficacité
- Facilité de mise en œuvre (nombreuses possibilités d'adaptation du code).

Tandis que les inconvénients du *SGD* sont :

- La *SGD* nécessite un certain nombre d'hyper paramètres tels que le paramètre de régularisation et le nombre d'itérations.
- La *SGD* est sensible à la mise à l'échelle des caractéristiques.

II.6. Méthodologie

Dans ce processus, un algorithme est d'abord conçu, puis il est entraîné avec un ensemble de caractéristiques spécifiques, par exemple les occurrences de mots ou encore les distributions de thèmes dans un document. Une fois entraîné, l'algorithme est utilisé pour étiqueter de nouveaux textes. Ces derniers sont différents des textes utilisés lors de l'entraînement. L'algorithme est évalué sur le nombre d'erreurs de classification obtenues lors de la phase d'apprentissage et lors de la phase de test.

II.6.1. But du projet

La classification de texte a pour fonction première de définir le contenu d'un document en seulement quelques mots, ainsi le but de notre projet est de créer des algorithmes capables de détecter et classer un texte du dialecte arabe (Dialecte Algérien) en trois catégories : Abusive, Offensive ou Normale.

II.6.2. Motivation

Récemment, la recherche sur la détection du langage offensif dans les dialectes arabes s'est beaucoup développée. Mais rares sont les approches qui ont abordé le dialecte algérien. En effet, avec la croissance du nombre d'utilisateurs sur les réseaux sociaux, et avec le contenu offensant laissé sans surveillance, la santé mentale et physique des personnes est en danger. Cependant, notre travail aidera pas mal de personnes à ne plus se faire intimider et diminuera aussi le taux de dépression au monde. Ainsi, la recherche dans ce sujet utilise généralement la classification automatique, et l'expérimentation du dialecte algérien nécessite un algorithme.

II.7. Notre approche

II.7.1. La catégorisation

Pour faire de la classification de texte, notre algorithme a besoin d'un corpus de textes d'apprentissage, c'est-à-dire un ensemble de texte possédant déjà une ou plusieurs catégories. Dans un premier temps il faut donc définir l'ensemble des catégories ou étiquettes que l'on souhaite attribuer aux documents en fonction de notre besoin et contexte. Une fois l'ensemble des catégories définies, l'ensemble des textes du corpus d'apprentissage devra avoir une ou plusieurs catégories. Ce travail est fait "manuellement". Cependant, nous avons déjà un corpus préétabli d'où la base de données que nous avons utilisée.

Ce corpus servira de base d'apprentissage à un algorithme d'apprentissage automatique. C'est-à-dire que l'algorithme va construire un modèle (dépendant de la méthode d'apprentissage choisie) pour définir des liens, des relations entre documents d'une même catégorie et ce qui différencie les catégories ente elles.

De ce fait, nous avons importé nos ensembles de données offensives, abusives et normales en utilisant la bibliothèque json car ces derniers sont en json (JavaScript Object Notation) qui est un format d'échange de données en texte lisible, et est utilisé pour représenter des structures de données et des objets simples dans un code qui repose sur un navigateur Web.

Puis, nous avons combiné ou fusionner les trois ensembles de données et afficher chaque texte avec son étiquette ou catégorie, ensuite on a divisé la base de donnée en deux, 90% pour entraîner les algorithmes et 10% pour faire le test.

II.7.2 La classification du texte

La première étape de toute tâche de traitement de texte consiste à lire les données. Ainsi, nous avons utilisé Python pour notre travail sur la classification de texte, en raison de sa syntaxe simple et du nombre de bibliothèques open-source disponibles. Nous avons aussi la bibliothèque pandas pour rendre les données uniforme.

En effet, un ordinateur ne fonctionne pas comme un cerveau humain et les algorithmes d'apprentissage automatique non plus. Un texte n'est ici qu'une suite de caractères binaires et n'a pas de sens pour un algorithme. Pour donner du sens à un texte, du point de vue de l'algorithme d'apprentissage, il faut transformer le texte en vecteur. Il s'agit de l'étape la plus importante des algorithmes de classification de texte qu'on appelle la vectorisation, ainsi python nous offre une librairie adéquate qui est sklearn.feature_extraction.text tout en important CountVectorizer pour transformer les strings en occurrence et ainsi effectuer la tâche à notre place.

II.7.3. L'extraction de caractéristiques avec scikit-learn

L'extraction de caractéristiques consiste à transformer des données arbitraires, telles que du texte ou des images, en caractéristiques numériques utilisables pour l'apprentissage automatique.

Le module sklearn.feature_extraction peut être utilisé pour extraire des fonctionnalités dans un format pris en charge par des algorithmes d'apprentissage automatique à partir d'ensembles de données constitués de formats tels que du texte.

II.7.3.1. La représentation du sac de mots

L'analyse de texte est un domaine d'application majeur pour les algorithmes d'apprentissage automatique. Cependant, les données brutes, une séquence de symboles, ne peuvent pas être transmises directement aux algorithmes eux-mêmes, car la plupart d'entre eux s'attendent à des vecteurs de caractéristiques numériques de taille fixe plutôt qu'à des documents textuels bruts de longueur variable [48]. Pour remédier à ce problème, scikit-learn fournit des utilitaires pour les méthodes les plus courantes d'extraction de caractéristiques numériques du contenu textuel, à savoir :

- La tokenisation des chaînes de caractères et l'attribution d'un identifiant entier pour chaque token possible, par exemple en utilisant les espaces blancs et la ponctuation comme séparateurs de tokens.
- Compter les occurrences des tokens dans chaque document.
- Normaliser et pondérer avec une importance décroissante les tokens qui apparaissent dans la majorité des échantillons / documents.

Dans ce schéma, les caractéristiques et les échantillons sont définis comme suit :

- Chaque fréquence d'occurrence de token individuelle (normalisée ou non) est traitée comme une caractéristique.
- Le vecteur de toutes les fréquences de tokens pour un document donné est considéré comme un échantillon multi varié.

Un corpus de documents peut donc être représenté par une matrice avec une ligne par document et une colonne par token (par exemple, un mot) apparaissant dans le corpus.

Nous appelons vectorisation le processus général consistant à transformer une collection de documents textuels en vecteurs de caractéristiques numériques. Cette stratégie spécifique (tokenisation, comptage et normalisation) est appelée représentation du sac de mot.

La représentation par sac de mots est assez simpliste mais étonnamment utile en pratique. En particulier, dans un cadre supervisé, elle peut être combinée avec succès à des modèles linéaires rapides et évolutifs pour entraîner des classificateurs de documents [48].

II.7.3.2 Décodage des fichiers texte

Le texte est constitué de caractères, mais les fichiers sont constitués d'octets. Ces octets représentent des caractères selon un certain codage. Pour travailler avec des fichiers texte dans Python, leurs octets doivent être décodés dans un jeu de caractères appelé Unicode. Les encodages courants sont ASCII, Latin-1 (Europe occidentale), KOI8-R (russe) et les encodages universels UTF-8 et UTF-16. Il en existe de nombreux autres [48].

Un codage peut également être appelé "jeu de caractères", mais ce terme est moins précis : plusieurs codages peuvent exister pour un seul jeu de caractères. Les extracteurs de caractéristiques textuelles de scikit-learn savent comment décoder les fichiers texte, mais seulement si vous leur indiquez l'encodage des fichiers. Le CountVectorizer prend un paramètre d'encodage à cette fin. Pour les fichiers texte modernes, l'encodage correct est probablement UTF-8, qui est donc la valeur par défaut (encoding="utf-8") [48].

II.7.4. Algorithme à base d'apprentissage

Une fois les caractéristiques extraites, plusieurs algorithmes d'apprentissage peuvent être utilisés pour la classification automatique de textes. Dans notre cas, nous utiliserons comme algorithme d'apprentissage automatique :

- Le modèle multinomial naïve de bayes qui est proposé dans scikit-learn par la classe *MultinomialNB*
- Le *SVM* qui est quant à lui proposé par la classe *SVC* dans scikit-learn
- Le modèle *SGD* ou *SGDClassifier*

Le naïve de bayes multinomial est souvent utilisées dans la classification de textes, où les caractéristiques sont liées au nombre de mots ou aux fréquences dans les documents à classer et sont supposées être générées à partir d'une distribution multinomiale simple. La distribution multinomiale décrit la probabilité d'observer des comptages parmi un certain nombre de

catégories, et donc le modèle Bayes naïf multinomial est le plus approprié pour les caractéristiques qui représentent des comptages ou des taux de comptage. L'idée est de modéliser la distribution de données une distribution multinomiale la mieux ajustée [49].

Les *SVM* sont très efficaces dans les espaces de grande dimension et sont généralement utilisés dans les problèmes de classification. Les *SVM* sont populaires et efficaces en termes de mémoire car ils utilisent un sous-ensemble de points d'apprentissage dans la fonction de décision. Ainsi, *Scikit-learn* fournit la classe *SVC*, il s'agit d'une classification vectorielle de support en C dont l'implémentation est basée sur *libsvm*. Le module utilisé par *scikit-learn* est *sklearn.svm.SVC*, cette classe gère le support multi classe selon le schéma un contre un [50].

SGDClassifier peut optimiser la même fonction de coût que *LinearSVC* en ajustant les paramètres de pénalité et de perte. En outre, il nécessite moins de mémoire, permet un apprentissage incrémental (en ligne) et met en œuvre diverses fonctions de perte et régimes de régularisation [51].

Il s'agit d'évaluer l'algorithme d'apprentissage qui mettra en place le meilleur modèle pour répondre à un besoin, cette phase est de plus en plus simple grâce aux librairies de *ML* comme *Scikit-learn*. Enfin, vient l'étape de prédiction qui consiste à appliquer le modèle d'apprentissage construit, sur les textes ou documents que l'on souhaite classer.

II.8. Optimisation par méthode bayésienne

L'introduction aux stratégies d'optimisation bayésienne un sous-domaine regroupant des techniques très puissantes pour converger efficacement vers des valeurs optimales lorsqu'on fait face à une situation où le nombre d'observations est limité par des contraintes de temps ou de matériel [52].

Pour déterminer automatiquement les valeurs de paramètres qui nous permettront d'obtenir les meilleurs résultats, il existe des règles métier fournissant des pistes pour fixer ces paramètres mais ces règles ne sont pas forcément adaptées à notre charge de travail, elles sont rarement optimales et ne s'appuient pas nécessairement sur des théories. En effet, il est en réalité très difficile de déterminer des règles précises régissant ces paramètres pour rendre le choix d'une bonne configuration plus automatisé et plus efficace possible [52]. Il s'agit finalement d'un problème classique d'optimisation pour lequel de nombreuses solutions existent comme trouver rapidement une configuration maximisant ou minimisant une métrique de performance.

L'optimisation bayésienne est une approche probabiliste basée sur l'inférence bayésienne. En somme, cela veut dire qu'on va chercher à exploiter ce qu'on connaît déjà, donc l'ensemble des événements précédemment observés, pour inférer la probabilité des événements que nous n'avons pas encore observés. Dans le cadre de l'optimisation bayésienne, nous partons d'un ensemble d'observations dont nous connaissons le résultat et nous déterminons pour chaque valeur. En effet, lorsqu'on utilise un modèle de *ML*, il est difficile d'évaluer directement la valeur optimale de certains paramètres. De ce fait, on utilise l'optimisation bayésienne pour résoudre le problème à notre place : l'entrée de notre problème correspond aux valeurs des hyper-paramètres et la performance à optimiser est une métrique au choix de l'utilisateur.

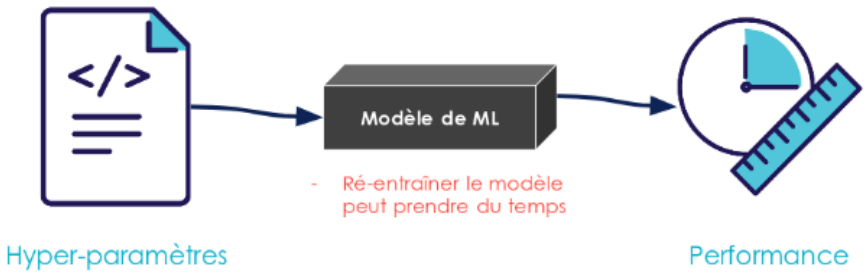

Figure 2.10. Détermination rapide des valeurs des hyper-paramètres maximisant la métrique de performance [52]

L'optimisation des hyper-paramètres est l'une des étapes cruciales de la formation des modèles d'apprentissage automatique. Avec de nombreux paramètres à optimiser, un temps d'apprentissage long et des plis multiples pour limiter les fuites d'informations, cette tâche peut s'avérer fastidieuse. Il existe quelques méthodes pour résoudre ce problème comme les méthodes bayésiennes [54]. Ainsi, Optuna est une implémentation de cette dernière.

II.8.1. Optuna

II.8.1.1. Définition

L'optimisation des hyper paramètres est l'un des processus les plus importants pour qu'un modèle d'apprentissage automatique soit très performant. Optuna est une bibliothèque Python populaire pour l'optimisation d'hyper paramètres. Il s'agit d'un logiciel facile à utiliser et bien conçu qui supporte une variété d'algorithmes d'optimisation [53]. Optuna devient alors le cadre d'optimisation de premier choix. Il est facile à utiliser, permet de définir le délai de l'étude, de poursuivre l'étude après une pause et d'accéder facilement aux données.

II.8.1.2. Étude d'optimisation

L'étude créée optimise à la fois l'étape de vectorisation et les hyper paramètres du modèle. Il est possible de choisir parmi 5 distributions [54] :

- Uniforme - valeurs flottantes
- log-uniforme - valeurs flottantes
- Discrète uniforme - valeurs flottantes avec intervalles
- Entier - valeurs entières
- Catégorique - valeurs catégoriques d'une liste

Les valeurs sont transmises au dictionnaire des paramètres, puis définies dans le modèle optimisé. Les valeurs pourraient être suggérées à l'intérieur du dictionnaire, mais cela rendrait les lignes de code très longues et difficiles à lire. La dernière étape de la définition de la fonction consiste en fait à définir l'objectif. Il doit retourner une seule valeur [54].

Cependant, pour créer l'instance d'une étude, il faut soit en créer une nouvelle, soit la charger à partir du fichier pickle pour poursuivre des expériences précédentes. Il faut spécifier la durée de l'étude en nombre d'essais (n_trials) ou en temps en secondes (timeout). Dans ce dernier cas, le dernier essai commence avant le timeout, et l'étude entière dure un peu plus longtemps que spécifié.

Notez que les meilleurs hyper paramètres jusqu'à présents sont affichés. Il faut accéder à la valeur de la meilleure métrique et au dictionnaire des meilleurs paramètres avec les attributs best_value et best_params respectivement. Vous pouvez accéder aux essais avec l'attribut trials, mais les créateurs d'Optuna ont préparé quelque chose de mieux, le fait d'utiliser la méthode trials_dataframe() pour créer un DataFrame Pandas avec les détails des essais [54].

En résumé les étapes pour utiliser et travailler avec optuna sont [55] :

- Définissez la fonction objective à optimiser.
- Suggérer des valeurs d'hyperparamètres en utilisant l'objet d'essai (n_trials).
- Créer un objet d'étude et invoquer la méthode d'optimisation sur 100 essais

Une fois l'étude terminée, on peut définir les meilleurs paramètres pour le modèle et l'entraîner sur l'ensemble complet de données. Après avoir fait l'étude de N_trials, nous utiliserons les meilleurs paramètres **C** qui est le paramètre de régularisation et **gamma** qui est le paramètre d'échelle du noyau pour l'optimisation du modèle de classificateur.

II.9. La sélection des caractéristiques

La sélection des caractéristiques ou feature selection (FS) est l'étape qui consiste à sélectionner certaines caractéristiques (par exemple, des mots ou des termes) à utiliser lors de la construction d'un classificateur automatique pour la catégorisation de textes. Au lieu de représenter un document avec toutes ses caractéristiques, nous pouvons le représenter en utilisant seulement les sélectionnées. De cette façon, le classificateur doit traiter moins de données [56].

Il y a deux raisons principales pour sélectionner certaines caractéristiques plutôt que d'autres [56]:

La précision ou accuracy : Tout d'abord, des études ont montré que les algorithmes d'apprentissage automatique peuvent produire de meilleurs résultats lorsqu'ils ne tiennent pas compte de toutes les caractéristiques. Il serait raisonnable de penser que plus il y a de caractéristiques prises en compte, plus le classificateur sera précis. Cependant, certaines caractéristiques n'ajoutent pas d'informations (elles ne sont que du bruit), et leur suppression peut rendre le classificateur plus précis.

La scalabilité ou scalability : Deuxièmement, comme les algorithmes d'apprentissage automatique sont gourmands en ressources (puissance de calcul, besoin mémoire, bande passante réseau, stockage, etc.), les exécuter sur un sous-ensemble de fonctionnalités génère généralement un temps considérable des économies. La possibilité de travailler avec un petit sous-ensemble de fonctionnalités garantit également la scalabilité ou l'évolutivité.

En combinant ces raisons, nous pouvons dire que la sélection de caractéristiques est la tâche de sélectionner le sous-ensemble de caractéristiques avec le meilleur rapport signal/bruit.

Ainsi, pour la tâche de classification des documents, nous avons besoin de caractéristiques de bonne qualité et les caractéristiques de bonne qualité contiennent beaucoup d'informations que le classificateur peut utiliser pour décider à quelle catégorie appartient un document [56]. Les différents objectifs de la sélection de caractéristiques sont l'amélioration de la précision de la prédiction, la compréhension des données pour différentes applications d'apprentissage automatique telles que le regroupement, la classification et la régression [57]. Par conséquent, la *FS* permet de minimiser l'ajustement excessif, de réduire la dimensionnalité des données, d'améliorer la précision, d'éliminer les données non pertinentes, d'accélérer la formation afin d'améliorer la compréhension et d'élucider les subtilités des données, parmi de nombreux autres avantages.

II.9.1. Méthode d'application

La sélection de caractéristiques vise à améliorer l'efficacité de la classification en ne sélectionnant qu'un minuscule sous-ensemble de caractéristiques appropriées parmi le large éventail initial de caractéristiques. La *FS* tente de trouver un ensemble optimal de caractéristiques en éliminant les caractéristiques redondantes et sans importance de l'ensemble de données. La suppression des caractéristiques non pertinentes et redondantes permet d'obtenir une bonne représentation du texte, de réduire la dimension des données, d'accélérer le cycle d'apprentissage du modèle et d'améliorer les performances du modèle prédictif [58].

Le principal défi de la *FS* consiste à choisir le plus petit nombre de caractéristiques à partir de l'ensemble de données primaire qui se compose parfois d'un grand nombre de caractéristiques. Il est assez difficile de trouver des relations spécifiques et de parvenir à une conclusion lorsqu'on a affaire à un grand ensemble de données, car certaines caractéristiques sont très liées au problème à résoudre, tandis que d'autres ne le sont pas. Si toutes les caractéristiques étaient sélectionnées, le résultat de la sélection s'en trouverait affecté. Par conséquent, pour trouver la meilleure solution, il est essentiel de sélectionner les caractéristiques qui sont les plus liées uniquement au problème donné. En outre, il convient d'éviter toute caractéristique susceptible d'affecter le résultat, de conduire à des résultats inexacts ou de faire perdre du temps au processus d'analyse [58]. L'idéologie consistant à minimiser les attributs dans le grand ensemble de données pendant la sélection des caractéristiques est représentée à la figure 2.11.

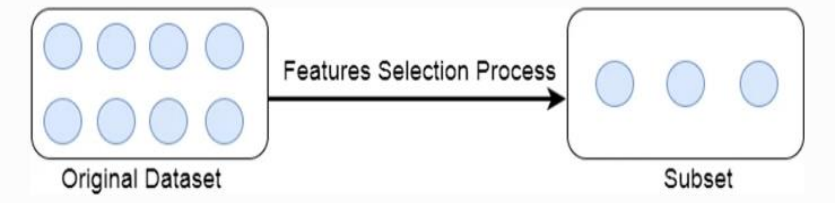

Figure 2.11. Concept de sélection des caractéristiques [58]

La figure 2.11 décrit le processus dans lequel on peut sélectionner manuellement ou automatiquement les caractéristiques de l'ensemble de données d'origine qui contribuent le plus à la variable de prédiction ou à la sortie qui nous intéresse. La présence de caractéristiques non pertinentes dans les données peut diminuer la précision des modèles et faire en sorte qu'un modèle apprenne sur la base de caractéristiques non pertinentes. Ainsi, à partir de l'ensemble de données original, un sous-ensemble de données est créé pour éliminer les caractéristiques non pertinentes.

Les quatre principales méthodes de *FS* pour la classification des textes sont les suivantes : la méthode par filtre (Filter), la méthode par enveloppe (Wrapper), la méthode intégrée (Embedded) et la méthode hybride (Hybrid) qui est la combinaison de deux ou plusieurs méthodes afin de produire une méthode de sélection de caractéristiques pour une meilleure classification des textes.

La littérature classe le processus de la *FS* en quatre catégories, à savoir les méthodes de filtrage, d'enveloppement, d'intégration et les méthodes hybrides.

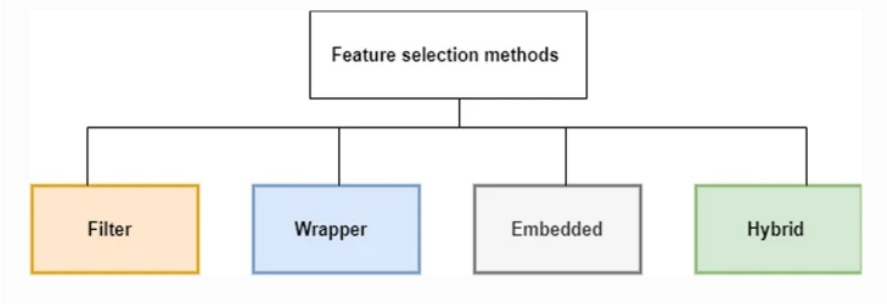

Figure 2.12. Classification des méthodes de sélection des caractéristiques [58]

Ainsi, les techniques de *FS* conventionnelles sont incapables de traiter certains problèmes. Par exemple, les méthodes basées sur le filtrage présentent des problèmes critiques tels que l'impossibilité d'augmenter le temps de consommation, de fournir des performances satisfaisantes, la complexité, etc. Ces défis, et bien d'autres encore, ont poussé les chercheurs à explorer diverses autres méthodes pour obtenir des options plus performantes lors de la tâche de classification. Ainsi, la recherche de meilleures techniques offrant des performances optimales a conduit à la découverte de méthodes de *FS* basées sur des métaheuristiques pour la classification de textes [58].

II.10. Méthodes métaheuristiques

Les algorithmes basés sur les métaheuristiques sont utilisés pour résoudre de nombreux types de problèmes d'optimisation en utilisant des opérateurs d'auto-apprentissage configurés avec des acteurs pour étudier et manœuvrer efficacement les solutions probables dans l'espoir d'arriver à la meilleure solution [58]. Ce sont des algorithmes inspirés de la nature et basés sur des principes scientifiques issus de la biologie, de l'éthologie, des mathématiques, de la physique, entre autres. En outre, ils sont identifiés comme des schémas algorithmiques de haut niveau qui fournissent une ruche de stratégies, de règles ou de directives pour concevoir des algorithmes d'optimisation heuristiques.

De nos jours, les méthodes de *FS* basées sur des métaheuristiques sont de plus en plus étudiées et appliquées en raison de l'importance et de la nécessité de la sélection de caractéristiques. Elles sont principalement classées en intelligence en essaim, algorithme évolutionnaire et algorithme basé sur la trajectoire.

L'intelligence en essaim est une technique d'optimisation stochastique basée sur la population qui est apparue comme une famille d'algorithmes inspirés par la nature. Elle décrit le comportement agrégé de cadres décentralisés, coordonnés et auto-organisés qui peuvent se déplacer rapidement de manière planifiée [58]. Citons quelques exemples de la méthode métaheuristique de sélection de caractéristiques basée sur l'intelligence en essaim : l'optimisation par essaims de particules ou particle swarm optimization (PSO), optimisation du loup gris ou grey wolf optimization (GWO), algorithme de la chauve-souris ou bat algorithm (BA), etc.

II.10.1. Algorithme de Bat

L'algorithme de Bat (BA) est motivé par le comportement d'écholocation des chauves-souris, et en raison de leur remarquable capacité d'écholocation a attiré l'attention de nombreux chercheurs. Ainsi, dans le contexte de la *FS*, cela donne à l'algorithme la flexibilité de s'adapter aux changements dans l'espace des caractéristiques et d'explorer de meilleures solutions [57]. Il peut être appliqué dans de nombreux domaines tels que la classification, la sélection de caractéristiques, l'exploration de données, etc.

BA est l'une des nouvelles méthodes métaheuristiques qui a été largement appliqué en tant que technique optimale de *FS* pour résoudre une grande variété de problèmes d'optimisation en raison de ses caractéristiques impressionnantes par rapport aux autres méthodes d'intelligence en essaim [59].

La sélection des caractéristiques joue un rôle important dans la classification et l'apprentissage automatique, puisqu'elle vise à éliminer les caractéristiques non pertinentes ou redondantes des données pour obtenir de meilleures performances de classification. La *FS* dépend de la représentation de toute solution comme une variable binaire d'une N- où N représente le nombre total de caractéristiques dans l'ensemble de données. Les techniques méta-heuristiques sont utilisées dans les problèmes de sélection de caractéristiques pour surmonter le problème de la génération de toutes les possibilités de combinaisons de caractéristiques [60].

II.11. Chi Square dans la sélection des caractéristiques

Dans la sélection des caractéristiques le Chi Square mesure l'indépendance d'une caractéristique et d'une catégorie. L'hypothèse nulle ici est que la caractéristique et la catégorie sont complètement indépendantes, c'est-à-dire que la caractéristique est inutile pour catégoriser les documents [56].

Nous avons utilisé SelectKBest pour sélectionner les caractéristiques avec le meilleur chi-carré, nous avons passé deux paramètres : l'un est la métrique de notation qui est le chi2 et l'autre est la valeur de K qui signifie le nombre de caractéristiques que nous voulons dans le jeu de données final.

Nous avons utilisé fit_transform pour adapter et transformer l'ensemble de données actuel en l'ensemble de données souhaité. Enfin, nous avons imprimé le jeu de données final et la forme du jeu de données initial et final.

II.12. Conclusion

Dans ce chapitre, nous avons présenté l'approche de la catégorisation automatique de texte à savoir le ML et le DL, le schéma général de certains classificateurs comme le SVM, NB, SGD, CNN, RNN et LSTM. Ces algorithmes sont couramment utilisés dans différentes tâches de catégorisation de textes et ont donné des résultats prometteurs. Nous avons aussi présenté notre approche méthodologique de façon détaillée.

Chapitre III
Résultats et expérimentations

I.1. Introduction

Comme déjà vu précédemment, la classification des textes consiste à attribuer des catégories prédéfinies à des documents. Dans ce chapitre, nous détaillons et discutons les différents résultats obtenus par les différents modèles de classificateur automatique en trois catégories : offensif, abusif et normal.

I.2. Revue générale

Nous avons mené une expérience propre à nous pour la catégorisation des textes. Avec la base de données (DziriOFN) de plus de 8,4k textes divisé en trois catégories qui sont : offensif, abusif et normal. Nous avons fusionné ses trois catégories pour ainsi faire notre classification automatique des textes à l'aide des modèles de l'apprentissage automatique afin de tester leur fiabilité. Le corpus a été divisé en deux ensembles, à savoir l'ensemble d'entrainement et l'ensemble de test. Le premier constitue 90% de la taille totale, tandis que le second est de 10%.

I.3. Résultats et discussions

I.3.1. Naïves Bayes

Le *NB* est une méthode très rapide qui dépend des probabilités conditionnelles et qui sont faciles à mettre en œuvre et à évaluer. Ainsi, le tableau 3.1 présente le résultat de la classification des trois catégories obtenue par le Multinomial NB.

Tableau 3.1. Résultat des tests de NBMultinomial

Catégorie	Précision	Recall	F1- Score	Nombre d'erreur de prédiction	Accuracy
Abusif	0,80	0,50	0,61	256	0,70
Normal	0,74	0,79	0,77		
Offensif	0,63	0,68	0,66		

Le *NBMultinomial* qui est un modèle de la *ML* nous donne un résultat assez remarquable avec un F1-Score de 0,61 pour la catégorie abusif, 0,77 pour normal et 0,66 pour offensif, avec aussi une précision totale (accuracy) de 0,70 et ceci sans l'intervention des méthodes d'optimisation. Ainsi, pour l'ensemble de test qui est de 10%, avec notre corpus de 8,4k textes nous nous retrouvons avec 840 textes pour faire le test. Cependant, après avoir entrainé le modèle et faire le test nous nous sommes retrouvés avec un nombre d'erreur de prédiction de 256 qui est plutôt abordable, soit 584 textes compris par le modèle.

I.3.2. SGD

Le *SGD* plutôt connu sous le nom *SGDClassifer* en python est une méthode d'optimisation qui est utilisé pour la classification des textes. De ce fait, les résultats obtenus par le classificateur linéaire optimisé par la *SGD* présenté dans le tableau 3.2 sont acceptables pour la catégorisation des textes.

Tableau 3.2. Résultat des tests de SGDClassifier

Catégorie	Précision	Recall	F1- Score	Nombre d'erreur de prédiction	Accuracy
Abusif	0,93	0,59	0,72	249	0,71
Normal	0,70	0,85	0,76		
Offensif	0,68	0,58	0,63		

Avec SGDClassifier, nous avons obtenues une très bonne précision dans la catégorie abusive avec 0,93 et aussi un F1-Score de 0,72. Cependant, pour la normale et l'offensive, nous avons un F1-Score de 0,76 et 0,63 respectivement. Le nombre d'erreur a aussi diminué par rapport au *NB* qui donne un taux de 249 sur 870 textes pour le test. A la fin, nous avons obtenus une précision totale de 0,71 qui est abordable.

I.3.3. SVM

Le *SVM* est le modèle de l'apprentissage automatique le plus utilisé dans la classification automatique des textes car il a une vitesse plus élevée et de meilleures performances avec un nombre limité d'échantillons par rapport aux algorithmes de *DL*. Ainsi, les résultats expérimentaux peuvent en témoigner. Le tableau 3.3 présente les résultats de la classification à trois catégories obtenues par le modèle d'apprentissage automatique *SVM*.

Tableau 3.3. Résultats des tests de SVM

Catégorie	F1- Score	Temps pour l'entrainement	Temps pour le test	Accuracy
Abusif	0,77	6,66 secs	0,18 sec	0,72
Normal	0,63			
Offensif	0,74			

Les SVM ont produit les meilleurs résultats avec une précision totale de 0,72 pour un temps d'entrainement de 6,66 secondes et un temps pour le test de 0,18 secondes, ce qui est fortement rapide avec une vitesse élevée pour la classification automatique des textes. Ils ont aussi un F1-Score de 0,74 pour la catégorie offensive qui est supérieur à celui de *NB* et *SGD* qui ont 0,66 et 0,63 respectivement.

I.4. Expérimentations

Les résultats sont moins performants que pour les autres chercheurs, car les algorithmes ont du mal à faire la différence entre la catégorie offensive et abusive qui ont un trait peu similaire, cependant si nous pouvons par exemple les fusionner en une seule catégorie « offensive » et avoir au final une classe de deux catégories seulement : offensive et normale. Cette approche améliora largement la performance des modèles et aussi les résultats.

Dans notre travail, nous nous sommes concentrés que sur les modèles de *ML* pour la classification automatique des textes car suite à notre base de données qui n'est pas assez vaste les modèles de la *DL* auront du mal à donner une bonne performance sur la catégorisation des textes. Ainsi, nous avons aussi fait recours à des méthodes d'optimisation pour améliorer les résultats de certains de nos classificateurs et aussi avoir une distinction entre les catégories.

I.5. Optimisation par méthode bayésienne avec Optuna

Pour améliorer notre résultat, plusieurs approches sont enviables. Cependant, nous nous somme tourné vers optuna pour l'optimisation de l'algorithme *SVM*. Le but est de chercher certains paramètres afin de les modéliser et les intégrer dans le code. Comme paramètre, nous faisons références à **c** et **gamma** où nous essayons de faire un certain nombre d'essai ou trial afin de trouver les meilleurs paramètres apte à améliorer le résultat final du modèle ainsi avoir un meilleur F1-Score. Nous avons pris un N_trials=300 pour faire notre test. Cependant, après avoir eu le meilleur résultat du test nous avons pris les meilleures valeurs de c et gamma, pour les insérer dans le code *SVM* pour une meilleure optimisation.

Nous avons tracé une courbe dans la figure 3.1 pour décrire le fonctionnement de l'optimisation du modèle avec optuna.

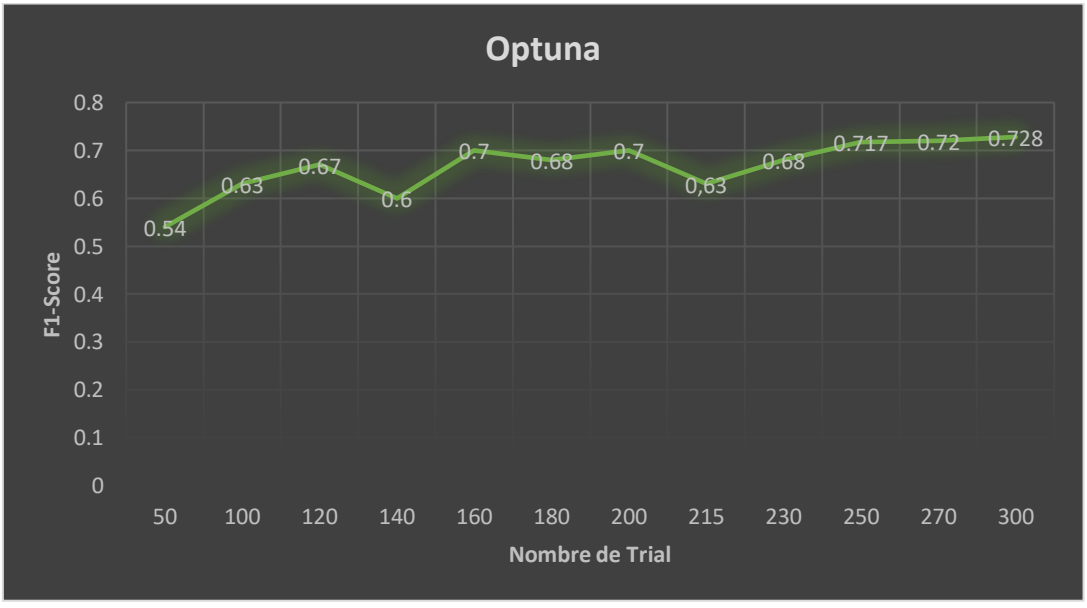

Figure 3.1. Optimisation de SVM pour trouver c et gamma

Les valeurs de ce graphe nous montrent que la courbe varie au cours de chaque essaie. Au bout d'un moment les valeurs stagnent et restent à peu près les mêmes et aussi plus on augmente le nombre d'essai plus les valeurs du F1-Score de l'algorithme change, soit ça augmente ou ça diminue. Cependant, l'optimisation de ce dernier n'est pas satisfaisant, pour ce faire nous devons nous pencher vers une autre méthode d'optimisation.

I.6. Optimisation avec Chi Square de la FS

Les méthodes de sélection de caractéristiques créent généralement une liste classée de caractéristiques qui sont bonnes pour entraîner un classificateur pour une catégorie spécifique. En d'autres termes, ils sont généralement spécifiques à une catégorie et le classificateur qui utilise les caractéristiques sélectionnées est généralement formé pour décider si les documents appartiennent à cette catégorie spécifique.

Ainsi, pour optimiser ou améliorer les résultats, notre approche s'est portée sur le Chi-Square ou Chi2 qui est une méthode d'optimisation de *FS*. Le but de Chi2 est de sélectionner les caractéristiques avec le meilleur chi-carré. Nous avons passé deux paramètres : l'un est la métrique de notation qui est le chi2 et l'autre est la valeur de K qui signifie le nombre de caractéristiques dont nous avons besoin.

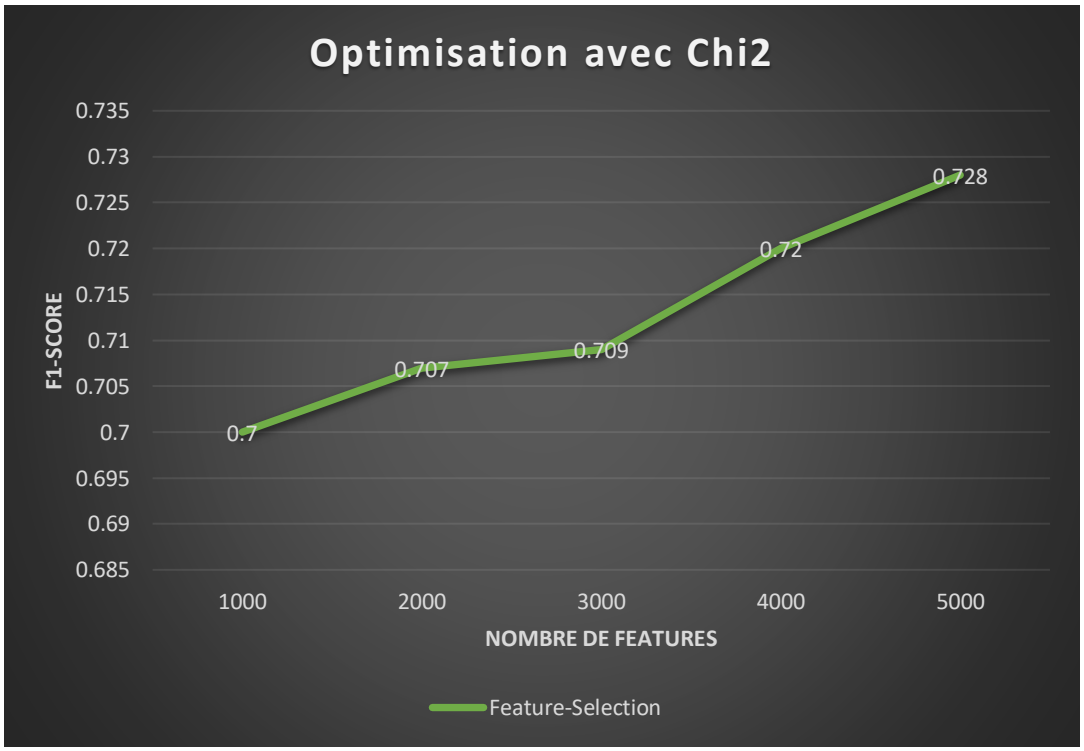

Figure 3.2. Optimisation avec Chi2 de la sélection des caractéristiques

Dans la figure 3.2, nous représentons un ensemble de caractéristique qui est varié pour catégoriser le texte et avoir un bon résultat de classification automatique. On remarque que plus le nombre de caractéristique (nombre de features) augmente plus le F1-Score augmente aussi. D'une façon générale, nous pouvons dire que la courbe est ascendante. Ainsi, pour améliorer le résultat des algorithmes, nous prenons le plus grand nombre de caractéristique pour ainsi l'utiliser le code.

I.7. Conclusion

Dans ce chapitre, nous avons évalué et discuté les résultats des algorithmes de l'apprentissage automatique. Ainsi, les résultats expérimentaux réalisés sur la classification des trois catégories ont montré que notre proposition produisait des résultats acceptables. En outre, SVM a surpassé tous les autres algorithmes en raison des techniques d'optimisation utilisé. Enfin, notre travail pourrait être étendu en incorporant une énorme base de données afin d'utiliser plus de modèle pour la classification des textes.

Conclusion Générale

Dans ce travail, nous avons présenté notre approche pour résoudre le problème de détection du langage offensif, où nous nous sommes concentrés sur la langue dialectale algérienne afin de protéger les utilisateurs (ou groupe d'individus). En particulier, la protection contre les dommages et les dangers infligés par le contenu offensif dans les médias sociaux. Cette approche consiste à construire des modèles à base d'apprentissage capables de classer le texte en tant que offensif, abusif ou normal.

Nous avons parlé sur le langage arabe, ainsi que le dialecte arabe en particulier celui de l'Algérie. Ensuite, nous avons défini le langage offensif sur les réseaux sociaux avec ses catégories, ses effets négatifs, sa modélisation et ses perspectives. Nous avons aussi brièvement discuté les différents algorithmes d'apprentissage automatique et d'apprentissage profond qui sont couramment utilisés dans la catégorisation des textes.

Notre méthodologie de travail consiste à diviser la base de données pour la préparer à l'entrainement et au test (90% et 10%, respectivement). Nous avons utilisé plusieurs algorithmes de classification pour entraîner nos modèles avec la base de données. Ainsi, nous avons utilisé trois modèles à base d'apprentissage pour la classification automatique des textes, afin d'évaluer les performances des algorithmes sur les données textuelles nous avons fait appel à d'autres méthodes d'optimisations comme la sélection des caractéristiques pertinentes parmi tant d'autres.

Nous avons obtenus de bons résultats pour la détection du langage offensif, cependant dans un travail futur ces résultats peuvent être encore plus performants en utilisant une grande base de données avec des textes supérieurs que ce qu'on avait.

Bibliographies

1. Mémoire_PFE_Boucherit.Pdf 2021 [1]
2. https://vous-avez-dit-arabe.webdoc.imarabe.org/langue-ecriture/quel-arabe-parles-tu/quelles-differences-fait-on-entre-arabe-standard-et-dialecte
3. https://apprendre-larabe-facilement.com/arabe-litteraire-dialectal/
4. https://aclawgroup.com.au/criminal-law/offences/offensive-language/
5. https://en.wikipedia.org/wiki/Online_hate_speech#Hate_speech
6. https://www.canada.ca/fr/securite-publique-canada/campagnes/cyberintimidation/cyberintimidation-jeunes/qu-est-ce-que-la-cyberintimidation.html
7. https://www.arrondissement.com/tout-get-document/u4389-cyberintimidation-nouvelle-realite-pour-jeunes
8. https://www.education.gouv.fr/non-au-harcelement/qu-est-ce-que-le-cyberharcelement-325358
9. https://www.securitepublique.gouv.qc.ca/fileadmin/Documents/police/statistiques/criminalite/cyberintimidation/Bulletin_statistique_cyberintimidation_cyberharcelement.pdf
10. https://www.quebec.ca/famille-et-soutien-aux-personnes/violences/intimidation/cyberintimidation
11. https://fr.wikipedia.org/wiki/Sexisme
12. https://human-rights-channel.coe.int/stop-sexism-fr.html
13. https://www.unia.be/fr/criteres-de-discrimination/racisme
14. https://www.brookings.edu/blog/how-we-rise/2021/12/01/combating-racism-on-social-media-5-key-insights-on-bystander-intervention/
15. Understanding Abuse: A Typology of Abusive Language Detection Subtasks Zeerak Waseem, Thomas Davidson, Dana Warmsley and Ingmar Weber https://arxiv.org/pdf/1705.09899.pdf, 30 mai 2017
16. Abusive Langage on Social Media Through the Legal Looking Glass, page (10) https://aclanthology.org/2021.woah-1.20.pdf
17. https://institutducerveau-icm.org/fr/depression/
18. https://www.passeportsante.net/fr/Maux/Problemes/Fiche.aspx?doc=depression_pm
19. La dépression https://solidarites-sante.gouv.fr/IMG/pdf/guide-8.pdf 14 mai 2022
20. https://www.hindawi.com/journals/complexity/2018/6157249/ 14 mai 2022
21. https://www.chezmonpsy.ca/evenements-de-la-vie/difficultes-relationnelles/#:~:text=Difficult%C3%A9s%20%C3%A0%20d%C3%A9velopper%20et%20maintenir,ou%20recevoir%20celui%20des%20autres
22. Hamdy Mubarak, Kareem Darwish, Walid Magdy.: Abusive language detection on Arabic social media. Proceedings of the first workshop on abusive language online. 52-56 (2017)
23. Batoul Haidar, Maroun Chamoun, Ahmed Serhrouchni.: Offensive A Multilingual System for Cyberbullying Detection: Arabic Content Detection using Machine Learning. Adv. Sci. Technol. Eng. Syst. J. 2(6). 275-284 (2017)++

24. Raghad Alshaalan, Hend Al-Khalifa.: Hate Speech Detection in Saudi Twittersphere: A Deep Learning Approach. Applied Sciences. Multidisciplinary Digital Publishing Institute. 8614 (2020)
25. Ibrahim Abu Farha, Walid Magdy.: Multitask Learning for Arabic Offensive Language and Hate-Speech Detection. Proceedings of the 4th Workshop on OpenSource Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection. 86-90 (2020)
26. Fatemah Husain.: Arabic Offensive Language Detection Using Machine Learning and Ensemble Machine Learning Approaches. arXiv preprint arXiv:2005.08946. (2020)
27. Hatem Haddad, Hala Mulki, Asma Oueslati.: T-hsab: A tunisian hate speech and abusive dataset. International Conference on Arabic Language Processing. 251-263 (2019)
28. Imane Guellil, Ahsan Adeel, Faical Azouaou, Mohamed Boubred, Yousra Houichi, Akram Abdelhaq Moumna: Sexism detection, ArXiv:2104.01443, (2021)
29. Predicting the Type and Target of Offensive Posts in Social Media Marcos Zampieri, Shervin Malmasi2, Preslav Nakov3, Sara Rosenthal, Noura Farra, Ritesh Kumar6 file:///C:/Users/hp/Downloads/N19-1144.pdf page 06 (2019)
30. Fighting Offensive Language on Social Media with Unsupervised Text Style Transfer https://arxiv.org/pdf/1805.07685.pdf (20 mai 2018) page 06
31. https://ia-data-analytics.fr/machine-learning/#definition
32. https://datascientest.com/machine-learning-tout-savoir
33. https://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/
34. Figure Deep learning &Machine learning https://www.jedha.co/formation-ia/vraie-difference-machine-learning-deep-learning
35. https://developers.google.com/machine-learning/guides/text-classification
36. https://www.katacoda.com/basiafusinska/courses/nlp-with-python/text-classification
37. CNN & RNN https://link.springer.com/content/pdf/10.1186/s40537-021-00444-8.pdf
38. Fundamentals of deep learning and computer vision : a complete guide to become an expert in deep learning and computer vision
39. https://www.tensorflow.org/text/tutorials/text_classification_rnn#create_the_text_encoder
40. https://www.analyticsvidhya.com/blog/2021/06/lstm-for-text-classification/#h2_1
41. C-LSTM Neural Network for Text Classification Chunting Zhou, Chonglin Sun, Zhiyuan Liu, Francis C.M. Lau https://arxiv.org/pdf/1511.08630.pdf 2015 page 10
42. https://dataanalyticspost.com/Lexique/svm/#:~:text=SVM%20(Support%20Vector%20Machine%20ou,ou%20de%20d%C3%A9tection%20d'anomalie
43. https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/
44. https://doc.lagout.org/Others/Data%20Mining/Handbook%20of%20Statistical%20Analysis%20%26%20Data%20Mining%20Applications%20%5BNisbet%2C%20Elder%20%26%20Miner%202009-06-05%5D.pdf
45. https://www.baeldung.com/cs/naive-bayes-vs-svm

46. SGD Explication https://michael-fuchs-python.netlify.app/2019/11/11/introduction-to-sgd-classifier/
47. https://scikit-learn.org/stable/modules/sgd.html
48. https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction
49. https://jakevdp.github.io/PythonDataScienceHandbook/05.05-naive-bayes.html
50. https://www.tutorialspoint.com/scikit_learn/scikit_learn_support_vector_machines.htm
51. https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
52. Optimisation Baysienne https://blog.octo.com/loptimisation-bayesienne-par-lexemple-a-quoi-ca-sert-et-comment-ca-marche/#:~:text=L'optimisation%20bay%C3%A9sienne%20est%20une,n'avons%20pas%20encore%20observ%C3%A9s
53. https://medium.com/optuna/an-introduction-to-the-implementation-of-optuna-a-hyperparameter-optimization-framework-33995d9ec354
54. https://towardsdatascience.com/how-to-make-your-model-awesome-with-optuna-b56d490368af (30 mai 2022)
55. https://optuna.org/ (30 mai 2022)
56. https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/250768/347827_FULLTEXT01.pdf?sequence=1 feat_select & Chi2 5 juin 22
57. https://www.emerald.com/insight/content/doi/10.1016/j.aci.2018.04.001/full/pdf?title=hybrid-binary-bat-enhanced-particle-swarm-optimization-algorithm-for-solving-feature-selection-problems FS & BAT 6juin22
58. https://link.springer.com/article/10.1007/s00521-021-06406-8#Sec13 (6juin22)
59. https://dl.acm.org/doi/abs/10.3233/IDA-205455#d62701942e1
60. https://pdfs.semanticscholar.org/14ee/2d759939827adc175b79f6bef39944f35371.pdf