République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université de Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj
Faculté des Mathématiques et d'Informatique
Département d'informatique

MEMOIRE

Présenté en vue de l'obtention du diplôme

Master en Informatique

Spécialité : Ingénierie de l'Informatique Décisionnelle

THEME

La Reconnaissance du Langage Offensant dans le Contenu Arabe en Ligne

Présenté par :

Boussouf Silia

Soutenu publiquement le : jj/mm/aaaa

Devant le jury composé de:

Président : ……..

Examinateur : ……

Encadreur : ……

2023/2024

الإهداء

بسم الله الرحمن الرحيم الحمد لله الذي ما نجحنا وما علونا ولا تفوقنا إلا برضاه

الحمد لله الذي ما اجتزنا دربا ولا تخطينا جهدا إلا بفضله واليه ينسب الفضل والكمال والاكمال

(وَآخِر دَعْوَاهُمْ أَنِ الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِين)

بعد مسيرة دراسية دامت سنوات وحملت في طياتها الكثير من الصعوبات والتجارب والتعب، ها أنا ذا اليوم، أنا أقف على عتبة تخرجي اقطف ثمار تعبي وأرفع قبعتي بكل فخر

الحمد لله حباً وشكراً وامتناناً. ما كنت لأفعل هذا لولا فضل الله، فالحمد الله على البدء وعلى الختام

.أهدي هذا النجاح لنفسي أولاً، ثم لكل من سعى معي لإنهاء هذه الرحلة بالنجاح دمتم لي سندا لا عمر له

إلى نبراس أيامي ووهج حياتي إلى التي ظلت دعواتها تضم اسمي دائماً إلى من أفنت عمرها في سبيل أن أحقق طموحي قدوتي ومعلمتي الأولى التي منها تعرفت على القوة والثقة بالنفس لمن رضاها يخلق لي التوفيق (أمي) أطال الله في عمرك بالصحة والعافية

إلى من لا ينفصل أسمي عن أسمه ذلك الرجل العظيم، رَجُلٍ عَلمني الحياة بأجمل شكل وبذل كل ما بوسعه ولم يبخل، مأمني الوحيد وفرحتي الدائمة (أبي) أدامك الله لنا

الى ملهمي نجاحي صُناع قوتي صفوة ايامي وسلوة أوقاتي الى الشموع التي تنير لي الطريق الى قرة عيني إخوتي

إلى براعم العائلة مريم،امير وعبد الرحيم

لم تكن الرحلة قصيرة والأمور لم تكن سهلة ولكن بعون الله اجتزتها

Remerciement

Tout d’abord, je tiens à remercier Dieu de m'avoir donné la force et la capacité de terminer ce projet. Je suis profondément reconnaissant envers mes parents pour m’avoir facilité ma vie en me fournissant tout ce dont j'avais besoin pour réussir mes études.

Un grand merci à mon encadrant, Dr Mohdeb Djamila, pour avoir accepté de me superviser, pour ses conseils avisés et sa disponibilité. Je lui suis extrêmement reconnaissant d'avoir partagé ses connaissances et sa méthode de travail avec moi.

J’adresse mes remerciements à l'ensemble du jury pour avoir accepté de participer à ma soutenance et pour avoir partagé leurs connaissances et leur expertise.

Je n'oublie pas toutes les personnes qui, de près ou de loin, ont contribué à l'élaboration de ce travail et que je n'ai pas pu citer.

Enfin, je tiens également à remercier tous ceux qui liront ce mémoire, que ce soit pour s'informer ou pour approfondir leur connaissance du sujet. Je vous souhaite une excellente continuation dans vos travaux futurs.

Résumé

Dans cette étude, nous avons abordé la problématique de la détection du langage offensant sur les médias sociaux en arabe, une langue souvent sous-représentée dans les recherches en traitement automatique du langage naturel (TALN). En nous appuyant sur une base de données publique récemment publiée, nous avons entraîné plusieurs modèles de machine learning et de deep learning pour accomplir cette tâche.

Les modèles de machine learning utilisés incluent le Naive Bayes, le SVM, les Arbres de décision et les Forêts aléatoires. Parallèlement, nous avons exploré des architectures de deep learning telles que les réseaux de neurones convolutionnels (CNN) et les réseaux de neurones récurrents (RNN). Nos expériences ont montré des résultats remarquables, démontrant l'efficacité de ces approches dans la détection du langage offensif en arabe.

Pour améliorer l'expérience utilisateur et faciliter l'application de notre travail, nous avons également développé une interface utilisateur complète en Python. Cette interface permet une utilisation intuitive de nos modèles de détection, rendant la technologie accessible à un public non technique.

Les résultats obtenus sont prometteurs et ouvrent la voie à des améliorations futures, notamment par l'optimisation des modèles actuels et l'exploration de nouvelles techniques d'apprentissage automatique et de deep learning.

Mots-clés : Détection du langage offensif, contenus offensifs, réseaux sociaux, discours haineux, traitement automatique du langage, apprentissage automatique, apprentissage profond, classification de texte.

Abstract

In this study, we addressed the issue of detecting offensive language on social media in Arabic, a language often underrepresented in natural language processing (NLP) research. By leveraging a recently published public dataset, we trained several machine learning and deep learning models to accomplish this task.

The machine learning models used include Naive Bayes, SVM, Decision Tree, and Random Forest. In parallel, we explored deep learning architectures such as convolutional neural networks (CNN) and recurrent neural networks (RNN). Our experiments yielded remarkable results, demonstrating the effectiveness of these approaches in detecting offensive language in Arabic.

To enhance user experience and facilitate the application of our work, we also developed a comprehensive user interface in Python. This interface allows for intuitive use of our detection models, making the technology accessible to a non-technical audience.

The results obtained are promising and pave the way for future improvements, particularly through the optimization of current models and the exploration of new machine learning and deep learning techniques.

Keywords: Offensive language detection, offensive content, social media, hate speech, natural language processing, machine learning, deep learning, text classification.

ملخص

في هذه الدراسة، تناولنا مشكلة اكتشاف اللغة المسيئة على وسائل التواصل الاجتماعي باللغة العربية، وهي لغة غالبًا ما تكون ممثلة تمثيلاً ناقصًا في أبحاث معالجة اللغة الطبيعية (NLP). من خلال الاستفادة من قاعدة بيانات عامة نُشرت مؤخرًا، قمنا بتدريب العديد من نماذج التعلم الآلي والتعلم العميق لتحقيق هذه المهمة.

تشمل نماذج التعلم الآلي المستخدمة نموذج نايف بايز، SVM، شجرة القرار، وغابة العشوائية (Random Forest) . وبالتوازي، استكشفنا هياكل التعلم العميق مثل الشبكات العصبية الالتفافية (CNN) والشبكات العصبية المتكررة (RNN). أظهرت تجاربنا نتائج رائعة، مما يدل على فعالية هذه النهج في اكتشاف اللغة المسيئة باللغة العربية.

لتحسين تجربة المستخدم وتسهيل تطبيق عملنا، قمنا أيضًا بتطوير واجهة مستخدم شاملة بلغة بايثون. تتيح هذه الواجهة الاستخدام السهل لنماذج الكشف الخاصة بنا، مما يجعل التكنولوجيا في متناول الجمهور غير الفني.

النتائج التي تم الحصول عليها واعدة وتمهد الطريق لتحسينات مستقبلية، لا سيما من خلال تحسين النماذج الحالية واستكشاف تقنيات جديدة في التعلم الآلي والتعلم العميق.

الكلمات المفتاحية: اكتشاف اللغة المسيئة، المحتوى المسيء، وسائل التواصل الاجتماعي، خطاب الكراهية، معالجة اللغة الطبيعية، التعلم الآلي، التعلم العميق، تصنيف النصوص.

Table des matières

Liste des figures

Liste des tableaux

Introduction Générale

1. Contexte

Avec l'expansion rapide des plateformes de médias sociaux, la communication en ligne est devenue un aspect incontournable de la vie quotidienne. Cependant, cette croissance s'accompagne de l'augmentation des contenus offensifs et haineux, qui peuvent avoir des impacts négatifs significatifs sur les individus et les communautés. La détection et la gestion de ces contenus sont donc devenues des préoccupations majeures pour les chercheurs en traitement automatique du langage naturel (TALN). Malgré les avancées significatives dans ce domaine, la langue arabe reste sous-représentée dans les recherches actuelles, ce qui crée un besoin urgent de développer des outils et des techniques adaptés pour cette langue spécifique.

2. Problématique

La langue arabe, avec sa richesse morphologique et ses variétés dialectales, pose des défis uniques pour la détection automatique de la langue offensante. La complexité de la langue, combinée à un manque de ressources linguistiques adaptées, rend difficile l'application des techniques de TALN développées principalement pour les langues occidentales. Par conséquent, il est essentiel de développer et d'adapter des modèles capables de détecter efficacement le langage offensif en arabe sur les plateformes de médias sociaux.

3. Objectifs de la recherche

L'objectif principal de cette étude est de concevoir et d'évaluer des modèles de machine learning et de deep learning pour la détection du langage offensif en arabe sur les médias sociaux. En utilisant une base de données publique récemment publiée, nous visons à entraîner et à tester divers modèles afin de déterminer les approches les plus efficaces. De plus, nous souhaitons rendre ces modèles accessibles et utilisables par un public non technique en développant une interface utilisateur conviviale en Python.

4. Structure du rapport

Ce mémoire est structuré en quatre chapitres principaux comme suit :

- **Chapitre 1** : Dans ce chapitre, nous introduisons le concept de langage offensif, explorons les différents types de langage offensif en ligne et discutons de leurs effets négatifs sur les utilisateurs.

- **Chapitre 2** : Nous nous concentrons dans ce chapitre sur la langue arabe et présentons quelques travaux connexes sur la détection automatique du langage offensant en arabe.

- **Chapitre 3** : Ce chapitre décrit en détail les principales étapes nécessaires pour effectuer la détection automatique du langage offensant, incluant le prétraitement des données et l'extraction des fonctionnalités. Nous détaillons par la suite les modèles de machine learning et de deep learning employés pour la détection du langage offensif.

- **Chapitre 4** : Ce dernier chapitre présente les résultats des expérimentations menées et analyse l'efficacité des différents modèles. Nous discutons également des implications de ces résultats et des perspectives pour des recherches futures. En outre, nous décrivons le développement d'une interface utilisateur en Python, conçue pour rendre nos modèles de détection accessibles à un public non technique.

Chapitre 01 : Le Langage Offensif

1.1 Introduction

La prolifération croissante des plateformes de médias sociaux a considérablement élargi le champ d'expression et de partage des opinions en ligne. Cependant, cette expansion s'accompagne également d'une augmentation des discours offensants sur ces plateformes. Le filtrage manuel de ces contenus s'avère extrêmement difficile, conduisant à des efforts croissants pour automatiser ce processus [1].

Dans ce qui suit, nous examinerons la définition et la classification du langage offensant, en découvrant les méthodes traditionnelles et avancées de détection et de modération de ce type de langage. Nous discuterons également des approches de détection automatique, en mettant en lumière leurs avantages et leurs limitations.

1.2. Définitions et classification du langage offensif

1.2.1 langage offensif

Selon le dictionnaire Oxford, le terme "offensant" se réfère à une expression grossière ou impolie qui contrarie ou agace quelqu'un en montrant un manque de respect [2].

Dans le dictionnaire Collins, "offensant" décrit quelque chose qui dérange ou embarrasse les gens en raison de son impolitesse ou de son caractère insultant [3].

D’autre part, "offensant" est défini comme une attitude ou une position d'agression.

En se basant sur ces deux définitions, on peut définir le langage offensant dans le cadre de cette étude comme suit :

Le langage offensant implique l'utilisation d'une expression inacceptable et agressive, qu'elle soit verbale ou écrite, contre un individu ou un groupe [4]. Cela peut inclure des attaques directes

telles que des menaces, des insultes, l'usage de mots grossiers ou obscènes, même si cela se produit sous forme d'humour. Il peut également englober tout comportement antisocial considéré comme vulgaire par la majorité des gens.

1.2.2 Types de langage offensif

Dans la littérature, plusieurs types de langage offensant ont été identifiés. Dans cette sous-section, nous clarifions la distinction entre ces concepts et le langage offensant.

- **Violence verbale :** L'abus verbal, ou violence verbale, implique l'utilisation de mots pour nuire à une personne, prenant diverses formes et souvent difficile à définir clairement. Le préjudice causé est souvent difficile à mesurer, avec des formes courantes telles que les insultes. Cela peut inclure crier, insulter, intimider, menacer, humilier ou utiliser un langage désobligeant. Bien que la violence verbale soit généralement perçue comme des insultes orales envers autrui, elle peut également se présenter sous forme écrite. Le blasphème semble être un déclencheur courant de la violence verbale [5].

- **Discours haineux :** on entend habituellement des propos discriminatoires à l'encontre de personnes ou de groupes pour des motifs comme l'appartenance ethnique ou culturelle, l'origine, la nationalité, la religion, le sexe, l'orientation sexuelle ou le handicap. Cependant, le discours de haine englobe aussi des expressions non verbales, comme celles véhiculées par des images, des vidéos ou toute forme de communication en ligne et hors ligne [6].

- **Langage abusif** : C'est un type de langage offensif qui vise à blesser, humilier ou d'evaluer une personne ou un groupe de personnes. Il peut être utilisé pour intimidation, contrôle, manipulation ou domination. Le langage abusif peut prendre plusieurs formes, telles que les insultes, la critique constante, la menace, la mise en doute de la valeur personnelle et le cyberharcèlement[7].

- **Langage discriminatoire :** Cela peut inclure la stigmatisation d'une personne ou d'un groupe de personnes en raison de sa race, de son orientation sexuelle, de sa religion ou d'autres aspects de son identité. Il peut causer des dommages physiques, psychologiques et émotionnels aux victimes, Il peut inclure entre autres le langage raciste, le langage sexiste, et le langage xénophobe[8].

1.2.3 Niveaux de gravité et impact

Le langage offensif peut avoir des conséquences graves sur les individus, les groupes et la société dans son ensemble. Voici quelques exemples de conséquences du langage offensif :

- **Effets psychologiques** : Le langage offensif peut engendrer diverses répercussions sur le plan psychologique, comme la dépression, l'anxiété, le stress, le manque de confiance en soi, la colère, et la frustration. Les personnes ciblées peuvent également ressentir de la honte, de la culpabilité, et un sentiment d'isolement social [9].

- **Effets sociaux** : Le langage offensif peut aussi avoir des conséquences néfastes sur le plan social, incluant la stigmatisation, la discrimination, la marginalisation, et l'exclusion. Cela peut conduire à une mise à l'écart de la société ainsi qu'à des préjudices économiques et éducatifs pour les victimes[10].

- **Effets physiques :** Dans certains cas, le langage offensif peut même mener à des violences physiques et des comportements agressifs. Les menaces et les insultes peuvent encourager la violence ainsi que des attitudes haineuses envers les individus visés[11].

- **Effets sur la communauté :** Le langage offensif peut également impacter négativement l'ensemble de la communauté, en renforçant les préjugés, les stéréotypes, et les discriminations, et en contribuant à la polarisation et à la division sociale [12].

Voici quelques Cas concrets de violence liée à des discours haineux :

- L'attaque de la mosquée de Québec en 2017, au cours de laquelle six personnes ont perdu la vie et de nombreuses autres ont été blessées, perpétrée par un individu ayant proféré des commentaires haineux contre les musulmans sur les réseaux sociaux [13].
- Les violences envers les personnes d'origine asiatique aux États-Unis pendant la pandémie de COVID-19, où des discours offensants associant la pandémie à la Chine ont entraîné des actes de violence et de discrimination à l'encontre des Américains d'origine asiatique (BBC 2021).
- Les violences envers les migrants et les réfugiés, où des discours dégradants et déshumanisants ont été utilisés pour justifier la discrimination, la détention, voire des actes de torture et de violence mortelle [14].
- Les actes de terrorisme perpétrés par des groupes extrémistes, où des discours haineux sont

employés pour diaboliser des groupes entiers et légitimer des actions violentes[14].

1.3 Approches traditionnelles de modération

La propagation croissante du langage offensif en ligne a incité les propriétaires de plateformes à rechercher de nouvelles solutions de modération. Traditionnellement, la modération manuelle était largement utilisée, offrant l'avantage de comprendre le contexte et les nuances subtiles du langage. Cependant, cette approche est confrontée à des limitations telles que la fatigue, le potentiel de biais et l'incapacité à traiter de grandes quantités de contenu de manière rapide et cohérente. Afin de surmonter ces défis, des méthodes automatiques basées sur le contenu textuel ou la structure des conversations ont été proposées comme alternatives [15].

1.3.1 Filtres de mots et d’expression

Les filtres de mots et d'expressions font référence à un ensemble de règles ou d'algorithmes mis en œuvre sur diverses plateformes, telles que les réseaux de médias sociaux ou les forums en ligne, pour détecter et bloquer automatiquement l'affichage ou le partage de certains mots ou expressions. Ces filtres sont conçus pour empêcher la diffusion de contenus offensants, inappropriés ou préjudiciables, garantissant ainsi un environnement en ligne plus sûr et plus inclusif. En identifiant et en censurant une langue spécifique, les filtres de mots et d'expressions visent à maintenir les directives de la communauté et à faire respecter les normes éthiques de communication [16].

Les filtres de mots et d'expressions en ligne rencontrent des défis majeurs, notamment la difficulté à équilibrer la nécessité de filtrer avec le respect de la libre expression. Identifier et filtrer précisément les termes offensants tout en détectant le contexte et le sarcasme pose un problème. De plus, la mise à jour constante des filtres est nécessaire pour maintenir leur efficacité [17].

1.3.2 Analyse de la fréquence des termes offensant

L'analyse de la fréquence des termes offensants repose sur l'idée que certains termes ou expressions offensants peuvent être identifiés en analysant leur fréquence dans le langage. La méthodologie de cette analyse comprend plusieurs étapes :

L'analyse de la fréquence des termes offensants dans le langage comprend plusieurs étapes :

- ✧ Définir la variable à analyser : Identifier et catégoriser les termes offensants.
- ✧ Collecter les données : Rassembler les données pertinentes à partir de différentes sources comme des questionnaires ou des bases de données.
- ✧ Calculer la fréquence : Déterminer le nombre d'occurrences de chaque terme offensant par rapport au total des observations.
- ✧ Créer un tableau de fréquences : Organiser les fréquences calculées dans un tableau pour une meilleure compréhension.
- ✧ Visualiser les données : Utiliser des histogrammes ou des diagrammes à barres pour représenter graphiquement la distribution des termes offensants.
- ✧ Interpréter les résultats : Analyser les résultats en fonction du contexte de l'étude, en identifiant les termes les plus fréquents et en décelant des tendances ou des anomalies dans les données.

1.4 Approches basées sur l’apprentissage automatique

L'apprentissage automatique, également connu sous le nom de machine Learning, est une branche de l'intelligence artificielle qui se concentre sur le développement de techniques permettant aux ordinateurs d'apprendre à partir de données et d'améliorer leurs performances sans être explicitement programmés pour chaque tâche. En d'autres termes, au lieu de suivre des instructions strictes pour exécuter une tâche spécifique, un système d'apprentissage automatique est capable d'analyser des données, de détecter des modèles et d'effectuer des prédictions ou des décisions en se basant sur ces modèles [18]. Plusieurs approches basées sur l’apprentissage automatique sont couramment utilisées pour identifier automatiquement le langage offensif :

- Les méthodes de classification supervisée sont largement employées. Elles consistent à entraîner des algorithmes sur des jeux de données étiquetés, où chaque exemple est associé à une catégorie (offensant ou non offensant). Ces algorithmes apprennent à partir de ces exemples pour classifier de nouveaux textes comme étant offensants ou non. Les modèles classiques incluent les réseaux de neurones, les machines à vecteurs de support (SVM) et

les arbres de décision[18].

- Les approches basées sur les règles sont également utilisées. Elles consistent à définir manuellement des règles linguistiques ou des schémas de comportement offensif, puis à les appliquer aux textes pour détecter les expressions ou les intentions offensantes. Ces règles peuvent être élaborées par des experts du domaine ou apprises à partir de données[18].

- En outre, les méthodes de détection d'anomalies sont parfois employées pour identifier les comportements aberrants ou inhabituels qui pourraient indiquer un langage offensant. Ces méthodes utilisent souvent des techniques de modélisation statistique ou d'apprentissage non supervisé pour repérer les schémas inhabituels dans les données[18].

Figure 1.1 Processus du Machine Learning.

1.5 Approches basées sur l'apprentissage profond

L'apprentissage profond (DL) est une forme sophistiquée d'intelligence artificielle (IA) qui utilise des réseaux de neurones artificiels pour apprendre à partir de grandes quantités de données. Ces réseaux neuronaux, calqués sur les connexions complexes au sein du cerveau humain, sont composés de plusieurs couches qui traitent et transmettent les informations[20].

DL excelle dans la résolution de problèmes complexes en les décomposant en composants plus petits et plus gérables. Chaque couche de neurones du réseau reçoit et interprète les informations de la couche précédente, affinant progressivement sa compréhension. Par exemple, dans les tâches de reconnaissance d'images, le réseau apprend d'abord à identifier des pixels individuels, puis progresse vers la reconnaissance de formes et d'objets et, finalement, peut distinguer différents visages sur une photographie [20].

Les algorithmes d'apprentissage profond sont des architectures profondes d'apprentissage basé sur des couches consécutives sur plusieurs niveaux de représentation et d'abstraction. Comme le montre la figure suivante :

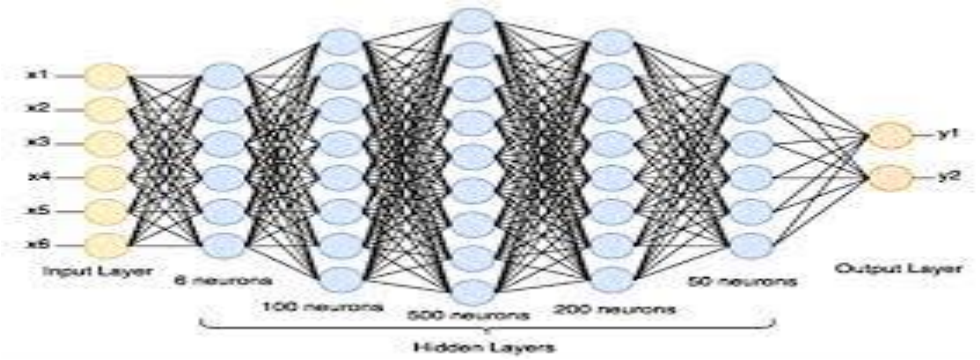


Figure 1.2 Architecture de Deep Learning.

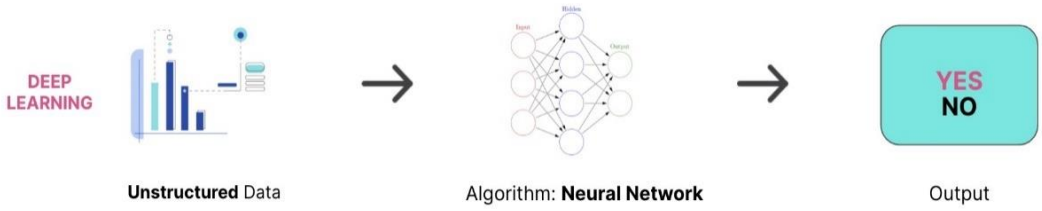


Figure 1.3 Processus de Deep Learning.

1.6 Approches basées sur le traitement automatique du langage naturel (TALN)

Le traitement automatique du Langage Naturel est un domaine à l'intersection du Machine Learning et de la linguistique. Il a pour but d'extraire des informations et une signification d'un contenu textuel.

Dans la littérature, plusieurs approches ont été explorées pour l'utilisation du traitement automatique du langage naturel (NLP) dans l'identification du langage offensif en ligne.

a) **L'analyse syntaxique et sémantique :** L'analyse syntaxique et sémantique est utilisée pour comprendre la structure grammaticale des phrases et repérer les nuances ou les sarcasmes

qui pourraient être utilisés pour dissimuler des propos offensants. Cette approche permet de prendre en compte le contexte et les intentions derrière le langage offensif[19].

b) **L'analyse de sentiment (Sentiment Analysis):** Cette approche consiste à évaluer l'orientation émotionnelle d'un texte pour détecter les expressions négatives ou offensantes. Des techniques telles que l'analyse de polarité et l'identification des émotions sont utilisées pour évaluer le ton général du texte[20].

c) **La détection d'entités nommées (Named Entity Recognition):** Les techniques de détection d'entités nommées sont souvent employées pour identifier des termes spécifiques associés à des insultes ou des propos offensants. Cela inclut l'identification des insultes, des termes discriminatoires ou des références à des groupes sociaux spécifiques[21].

d) **Les modèles de langage pré-entrainés :** L'utilisation de modèles de langage pré-entraînés comme BERT (Bidirectional Encoder Representations from Transformers) a gagné en popularité. Ces modèles capturent des informations contextuelles complexes dans le langage et peuvent être adaptés spécifiquement à la détection des discours haineux et des insultes[22].

1.7 Limitations des approches automatiques de détection du langage offensant

1.7.1 Erreurs fréquentes de la détection

- Faux Positifs : Identification erronée de termes ou expressions comme offensants, même s'ils ne le sont pas réellement, pouvant entraîner des actions inappropriées.
- Faux Négatifs : Non-détection de termes ou expressions réellement offensants, conduisant à des omissions et à des lacunes dans la détection.
- Sensibilité au Contexte : Certains modèles peuvent mal interpréter ou échouer à détecter le langage offensif dans des contextes complexes ou variés [23].
- Biais dans les Données d'Entraînement : Les données d'entraînement peuvent ne pas représenter adéquatement la diversité du langage offensif, entraînant des performances suboptimales [23].

- Adaptation au Bruit : Sensibilité aux variations non significatives dans les données, ce qui peut conduire à des détections incorrectes ou instables du langage offensif [23].

1.7.2 Problèmes liés à la diversité linguistique

- Manque de Données : Les langues moins courantes peuvent manquer de données adéquates pour former des modèles de détection du langage offensif, limitant ainsi leur efficacité[24].
- Biais dans les Données : Les données utilisées pour l'entraînement des modèles peuvent refléter des biais culturels et linguistiques, affectant la détection du langage offensif.
- Adaptation des Modèles : Les modèles entraînés sur des langues couramment utilisées peuvent avoir du mal à généraliser efficacement à des langues moins répandues, impactant la détection du langage offensif dans ces langues [24].
- Variations Dialectales : Les variations dialectales au sein d'une langue peuvent rendre la détection du langage offensif plus difficile, car les modèles doivent être capables de reconnaître ces différences [24].

1.8 Conclusion

L'évolution rapide des médias sociaux a ouvert de nouvelles possibilités de communication, mais aussi des défis en matière de modération du langage offensant. Malgré les progrès de la détection automatique de ce type de discours, des défis persistent, notamment la sensibilité culturelle et linguistique. La recherche continue est essentielle pour améliorer l'efficacité des outils de modération et garantir un environnement en ligne plus sûr pour tous.

Chapitre 02 : Détection du Langage Offensant en Arabe

2.1. Introduction

Les Arabes utilisent souvent les médias sociaux à diverses fins, telles que rechercher et partager des informations, établir une communication, faire de la publicité ou simplement pour se défouler. Un grand nombre d'utilisateurs des réseaux sociaux conduit souvent à une communication incontrôlée, ce qui peut augmenter le risque d'utiliser un langage offensant.

Dans ce chapitre, nous aborderons la situation actuelle de la détection du langage offensant en arabe, en explorant les défis spécifiques rencontrés et les solutions existantes. Nous examinerons également les progrès récents et les approches innovantes qui ont été proposées pour améliorer l'efficacité de la détection dans ce contexte linguistique particulier.

2.2. Le contenu arabe en ligne

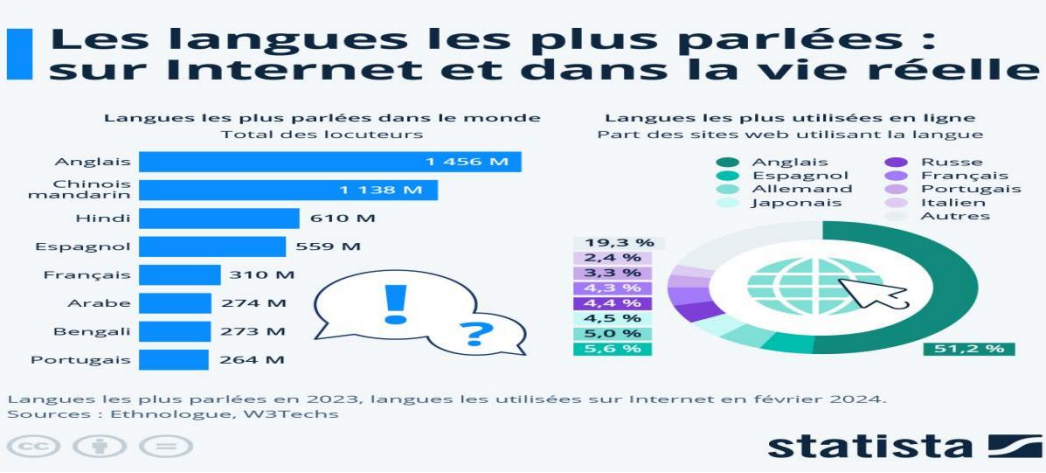


Figure 2.1 Langues courantes utilisées sur Internet 2024.

La langue arabe connaît un taux de croissance très élevé des moyens d'utilisation des réseaux

sociaux. Les statistiques réalisées par Statista [25] sur les langues les plus couramment utilisées sur Internet en février 2024, par part d'utilisateurs d'Internet, montrent que la langue arabe était classée sixième (voir figure 1).

Sur Internet et sur les réseaux sociaux en particulier, la situation linguistique de l'arabe se distingue des autres langues par deux aspects : le degré élevé de différence entre l'arabe standard et ses dialectes, et le fait que l'arabe standard n'est la langue maternelle d'aucun Arabe. Ce qui augmente la variation et la diversité du contenu arabe que l'on peut trouver. En effet:

- La majorité des textes arabes existants sur les plateformes de médias sociaux sont constitués de dialectes utilisés pour la communication quotidienne. Ils sont généralement rédigés en utilisant des formes courtes et des argots.
- Les publications en arabe standard sur les réseaux sociaux sont généralement officielles, professionnelles, publiées par des journalistes, des écrivains, etc., ou à des fins éducatives.

2.3. L'importance de la détection du contenu offensant publié en arabe

La détection des langues offensantes en arabe est particulièrement cruciale en raison de plusieurs facteurs spécifiques à la langue et à son contexte d'utilisation:

- Chaque langue, y compris l'arabe, a ses propres nuances et contextes culturels. La détection des langues offensantes en arabe permet de comprendre ces nuances et d'appliquer des politiques de modération culturellement sensibles[26].
- L'arabe est l'une des langues les plus utilisées sur les réseaux sociaux dans le monde arabe, générant une grande quantité de contenu quotidiennement, ce qui nécessite des outils efficaces pour surveiller et modérer ce contenu.
- De plus, le contexte arabe est souvent marqué par des conflits et des menaces. Un système capable de détecter le contenu offensant peut aider à identifier et à atténuer ces menaces, contribuant ainsi à la sécurité des utilisateurs des réseaux sociaux.

Cependant, les outils de traitement automatique des langues (TAL) pour l'arabe sont encore en développement et moins avancés que ceux pour d'autres langues comme l'anglais, en raison de la complexité de la langue arabe, qui comprend une riche morphologie et de nombreux dialectes [27]. La détection du contenu offensant en arabe présente des défis spécifiques supplémentaires. En effet, les divers dialectes arabes peuvent varier considérablement en termes de vocabulaire, de grammaire et de syntaxe [28]. De plus, le texte des médias sociaux est souvent informel et non standardisé, ce qui augmente encore la complexité de la détection du contenu offensant. Ces défis exigent des approches sophistiquées pour assurer une modération efficace et protéger les utilisateurs en ligne [29].

2.4. Cas particuliers et exemples de langage arabe offensant

En arabe, les mots offensants proviennent généralement de :

- Les animaux, font référence à sa certaine mauvaise caractéristique, qui est dégoûtante pour certaines personnes par exemple ‘يا حيوان’ ‘كلب العسكر’ ...
- Les références à des handicaps ou à des maladies, attaquent le lecteur en utilisant ses défauts par exemple ‘معوق مهبول’...
- Humiliation, action d'humilier quelqu'un, généralement en parlant de manière irrespectueuse ou en faisant référence à certaines professions, particulièrement aux occupations de classe inférieure par exemple ‘ زبال , بايرة’...
- Les propos obscènes, les propos grossiers ou profanes interdits par la religion par exemple ’ ‘الله يحرقكم , ربنا ياخدكم’...
- Racisme, notamment identitaire et racial (Africains réfugiés) par exemple ‘الارهاب, لقطاع طرق’...
- Déloyauté, tentative de condamner le lecteur par exemple ‘خونة ,حركي’.
- Malédiction, Dieu maudissant ou souhaitant de mauvaises choses au lecteur.
- Maudire la religion, généralement la religion du lecteur ou celle des membres de sa famille par exemple ‘لعنه الله’...

- Menace, expressions agressives pouvant provoquer de la colère ou de la violence et incitant à la violence par exemple ‘المنافقين , قله تربيه احثاله, يامعفنين’...

2.5. Détection du langage arabe offensant : Revue de la littérature

Un nombre limité de recherches ont contribué à l'automatisation de la détection du langage offensant en arabe. Voici un aperçu de certaines contributions majeures :

1) Moubarak H. et al. [30]: Ils ont créé un ensemble de données non biaisé par sujet, dialecte ou cible, incluant des balises pour la vulgarité et les discours de haine. Leur analyse a identifié les sujets, dialectes et genres les plus associés aux tweets offensants.

2) Mubarak H. et Darwish K. [31]: Ils ont élargi une liste initiale de mots offensants en contrastant des tweets offensants et non offensants, formant un classificateur d'apprentissage profond basé sur des n-grammes de caractères, avec un score F1 de 90 %.

3) Mohaouchane H. et al. [32]: Utilisant des commentaires YouTube étiquetés, ils ont testé quatre architectures de réseaux neuronaux pour la détection de langage offensant. Le modèle CNN-LSTM a obtenu le meilleur rappel de 83,46 %.

4) Husain F. [33] : Il a démontré que l'approche d'apprentissage automatique d'ensemble, notamment le bagging, surpassait les approches à apprenant unique, avec un score F1 de 88 %.

5) Alakrota A., Murray L. et Nikolov S. N. [34]: Ils ont utilisé des commentaires YouTube pour former un classificateur SVM, obtenant de bons résultats en combinant des fonctionnalités au niveau des mots et des N-grammes.

6) Keleg A. et al. [35]: Ils ont employé un modèle basé sur un transformateur (BERT) pour détecter le contenu offensant dans le cadre d'une tâche partagée sur la détection de langage offensant.

7) Haddad B. et al. [36] : Ils ont utilisé des modèles CNN et Bi-GRU augmentés par des couches d'attention pour détecter les propos offensants et les discours de haine, avec un score F1 de 0,859 pour le langage offensant et de 0,75 pour les discours de haine.

8) Sabit H. et al. [37]: Pour la détection de langage offensant et des discours de haine, ils ont combiné des SVM et des DNN, obtenant un score F1 de 90,51 % pour la détection de langage offensant.

Ces études illustrent les divers efforts et approches pour améliorer la détection du langage offensant en arabe, malgré les défis posés par la complexité et la diversité de la langue.

2.6. Limites de la détection de la langue arabe dans les médias sociaux

La détection de langage offensant en arabe sur les réseaux sociaux présente plusieurs limites :

- Langage informel : les publications sur les réseaux sociaux utilisent souvent un langage informel, des formes courtes et des argots difficiles à traiter sémantiquement et à comprendre par le classificateur[38].
- Diversité des dialectes arabes : La langue arabe possède de multiples dialectes avec des vocabulaires et des structures divers, ce qui augmente la complexité pour obtenir des performances de classification élevées [39].
- Manque de recherche approfondie : la plupart de la littérature présente des limites et sa portée ne couvre pas le sujet de la détection du langage offensant de manière exhaustive [40].
- Ensembles de données limités : Certaines études dépendent de très petits ensembles de données, ce qui ne suffit pas pour généraliser leurs résultats [40].

2.7. Conclusion

Ce chapitre a mis en lumière les défis et les solutions concernant la détection du langage offensant en arabe sur les médias sociaux. En explorant la situation actuelle, nous avons identifié les obstacles uniques rencontrés dans ce contexte linguistique spécifique. De plus, nous avons examiné les progrès récents et les approches innovantes proposées pour renforcer l'efficacité de la détection. Ce travail souligne l'importance cruciale de poursuivre la recherche dans ce domaine afin de développer des outils plus efficaces pour promouvoir un environnement en ligne plus respectueux et sécurisé.

Chapitre 03 : Méthodologie

3.1 Introduction

Dans ce chapitre, nous détaillerons la méthodologie et la conception de notre étude sur la détection du langage offensant en arabe. Nous commencerons par décrire la base de données utilisée, en mettant en évidence les étapes essentielles pour réaliser la détection automatique du langage offensant. Ces étapes comprennent le prétraitement des données et l'extraction de fonctionnalités, nécessaires pour l'application des méthodes d'apprentissage automatique. L'objectif final est de concevoir une application de détection du langage offensif spécifiquement adaptée à la langue arabe.

3.2 Description du projet

Ce projet vise à identifier la prévalence des discours offensants dans les contenus publiés par les communautés arabes sur la plateforme de médias sociaux Twitter. Pour atteindre cet objectif, nous avons élaboré une application de détection automatique en combinant des techniques d'apprentissage automatique, notamment le Deep Learning et le Machine Learning, spécifiquement adaptées au traitement du langage naturel arabe.

Voici les principales étapes de conception :

- Chargement des données à partir d'une base de données publique.
- Nettoyage, prétraitement et préparation des données en vue de la détection du discours offensant.
- Vectorisation du texte pour une représentation numérique.
- Classification des données en utilisant des modèles d'apprentissage adaptés.
- Évaluation des performances des modèles et analyse des résultats obtenus.

3.3 Le jeu des données

Pour notre application, nous avons combiné deux jeux de données publiques: 'TweetClassification-Summary.xlsx' et 'AJCommentsClassification-CF', accessibles via le lien http://alt.qcri.org/~mubarak/offensive/. Ces deux jeux de données, extraite du réseau social Twitter, contient 31100 tweets rédigés en arabe standard moderne (MSA) ainsi que dans différents dialectes. Elle offre des exemples représentatifs de divers types de textes (commentaires) offensants, comme illustré ci-dessous.

La base de données est organisée en trois classes distinctes, permettant une classification claire et précise:

a. **Mots Obscènes** : comprend des termes grossiers notés par (-2) par exemple "هبل ولاد كلب".
b. **Mots Offensives** : sont des mots ou expressions considérés comme blessants ou irrespectueux notée par (-1) par exemple "انت شارب"
c. **Mots Propres :** qui ne contient pas de langage offensant noté par (0) par exemple : " لا فلسطين أرض العرب والمسجد الاقصي قضيه كل مسلم".

3.3 Nettoyage et prétraitement de données

Avant d'avoir introduire les données dans les modèles d'apprentissage automatique, un prétraitement a été appliqué. L'objectif du prétraitement est de réduire les dimensions et de nettoyer les données des mots bruyants et dénués de sens. Cette étape peut améliorer la précision de la classification.

Le nettoyage et le prétraitement consiste à supprimer les caractères qui peuvent être bruyants, ce qui affecte la qualité des données. Il est important de filtrer les textes arabes par :

- Suppression des signes diacritiques arabes (Fatha, Damma, kasra, Sukun, tanwin etc.).
- Suppression des signes de ponctuation (?. : , ;), des caractères spéciaux (@ # $ % ').
- Normalization (Tatweel,la ligature ,hamza)
- Suppression des hashtags, des mentions (@), et des Emojis.

- **Tokenisation :** la tokenisation est le processus de division d'une séquence de chaînes, de texte en une liste de jetons tels que des mots, des mots-clés, des expressions, des symboles et d'autres éléments [41].

3.4 Vectorisation du texte

La vectorisation du texte (en Anglais: word embedding) est une technique pour représenter les mots sous forme de vecteurs numériques. Ces vecteurs capturent les relations sémantiques entre les mots, permettant aux machines de comprendre le sens et le contexte des mots dans un texte[44].

Les techniques suivantes ont été utilisées pour extraire des attributs (caractéristiques) à partir de données textuelles :

3.4.1 Sac de mots (Bag of Words)

Le sac de mots est l'une des méthodes les plus courantes pour transformer des jetons en un ensemble de fonctionnalités. Le modèle BoW est utilisé dans la classification de documents, où chaque mot est utilisé comme fonctionnalité pour former le classificateur.

3.4.2 TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF est une mesure statistique utilisée pour évaluer l'importance d'un mot dans un document d'une collection ou d'un corpus. La fréquence des termes (TF) représente la fréquence à laquelle un mot apparaît dans un document. Si nous avons plusieurs occurrences du même mot dans un même document, nous pouvons nous attendre à ce que le TF-IDF augmente.

La fréquence inverse des documents (IDF) représente la fréquence d'un mot dans les documents. Si un mot est utilisé dans plusieurs documents alors le TF-IDF diminuera.

3.5 Classification

Après les étapes de prétraitement des données et d'extraction des caractéristiques, vient l'étape d'utilisation de différents algorithmes de classification pour construire le modèle de classification qui nous permet de détecter les textes offensants. Nous avons testé quatre méthodes de

classification de base d'apprentissage automatique et deux architectures d'apprentissage profond :

1. Machines à Vecteurs de Support (SVM).
2. Arbre de Décision (Decision Tree).
3. Classificateur Naïve de Bayes.
4. Forêt Aléatoire (Random Forest).
5. Réseaux Neuronaux Convolutifs (CNN).
6. Réseaux Neuronaux Récurrents (RNN).

3.5.1 Machine à Vecteurs de Support ou SVM

Une machine à vecteurs de support ou Support Vector Machine (SVM) est un type d'algorithmes d'apprentissage automatique utilisé pour la classification et l'analyse de régression. Les SVM fonctionnent en trouvant l'hyperplan qui sépare le mieux les données en différentes classes (ou catégories). Les SVM sont de plus en plus populaires dans la Machine Learning et bien d'autres domaines [42].

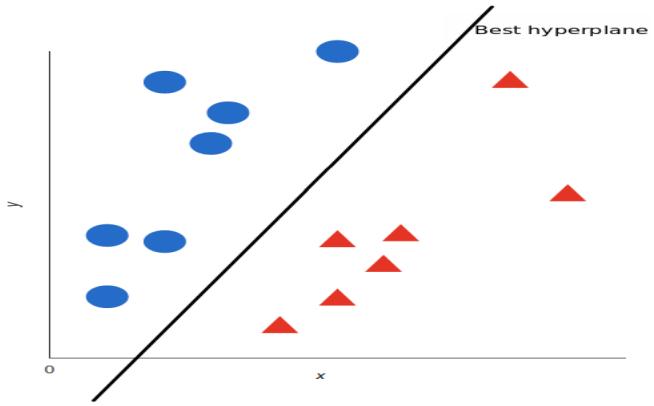


Figure 3.1 Classification de texte en utilisant SVM.

3.5.2 Arbre de Décision

Un arbre de décision (DT) est un modèle hiérarchique d'apprentissage supervisé par lequel la région locale est identifiée dans une séquence de divisions récursives dans un plus petit nombre

d'étapes. Un arbre de décision est composé de nœuds internes et de feuilles terminales. Chaque nœud de décision m met en œuvre une fonction d'attribution $f_m(x)$ avec des résultats discrets étiquetant les branches. Sur une entrée donnée, un test est appliqué à chaque nœud et l'une des branches est sélectionnée en fonction du résultat [43].

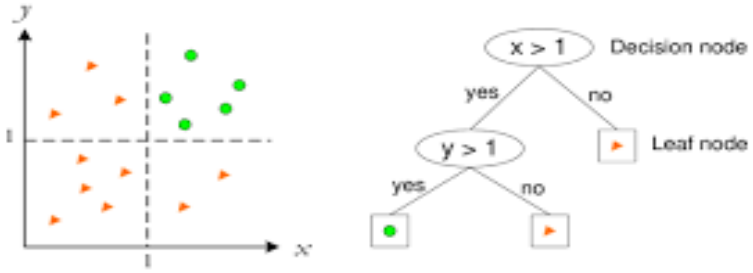


Figure 3.2 Classificateur Arbre de décision.

3.5.3 Naive de Bayes

Le classificateur Naive de Bayes est largement utilisé pour la classification de texte. Il repose sur le principe du théorème de Bayes et suppose que les caractéristiques sont indépendantes entre elles, d'où le terme "naïf".

Le classificateur multinomial Naive Bayes est un outil efficace pour la classification de texte. Il fonctionne en calculant les probabilités conditionnelles des classes données les caractéristiques extraites des documents. Après avoir été entraîné sur un ensemble de données étiqueté, il peut prédire la classe d'un nouveau document en utilisant ces probabilités. Malgré sa simplicité et sa rapidité, il peut être limité par son hypothèse d'indépendance entre les caractéristiques. Cependant, il reste largement utilisé en raison de sa robustesse et de ses performances satisfaisantes dans de nombreux cas d'utilisation [44].

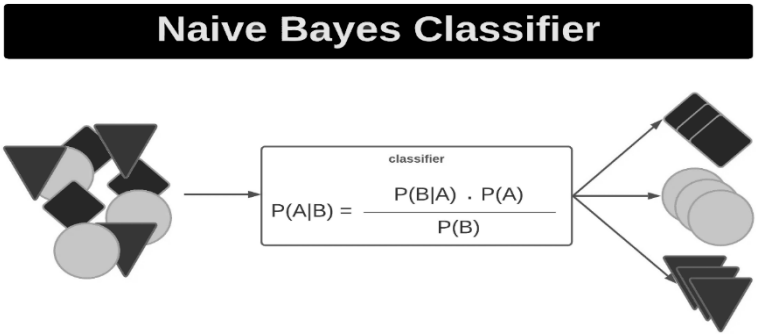


Figure 3.3 Classificateur Naive Bayes.

3.5.4 Random Forest (Forêt Aléatoire)

La Forêt Aléatoire est une méthode d'ensemble, c'est-à-dire qu'elle combine les résultats de plusieurs arbres de décision pour obtenir un résultat final Chaque arbre de décision dans la forêt est construit à partir d'un échantillon aléatoire de données, Elle est facile à interpréter, stable, et présente en général de bonnes précisions [48].

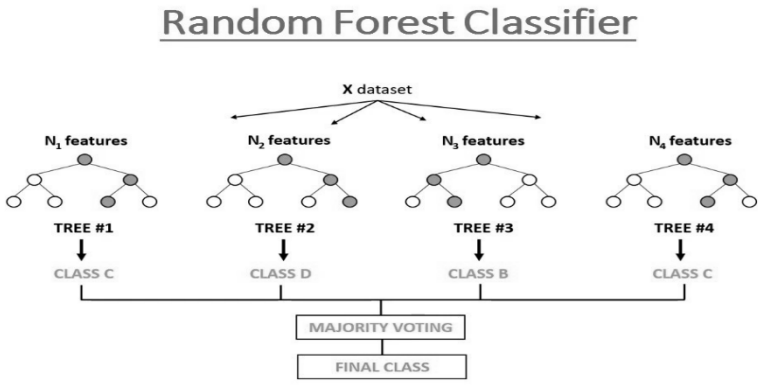


Figure 3.4 Classificateur Forêt Aléatoire.

3.5.5 Réseaux Neuronaux Convolutifs (CNN)

Les réseaux de neurones convolutifs (CNN) sont des réseaux profonds basés sur des filtres locaux capables de découvrir la corrélation au sein des données d'entrée grâce à l'opération de convolution. Les cartes de caractéristiques de sortie de cette convolution peuvent identifier

différents types de caractéristiques à chaque position temporelle. Les réseaux convolutifs comprennent principalement des couches empilées d'opérations de convolution et de pooling. L'opération de convolution applique chaque filtre local sur tous les sous-ensembles de l'entrée où les poids de ces filtres sont partagés sur tous les sous-ensembles. Ensuite, l'opération de regroupement divise les entités en sortie et applique une fonction pour réduire la taille de la couche précédente afin de préserver la propriété d'échelle de variance des entités [46].

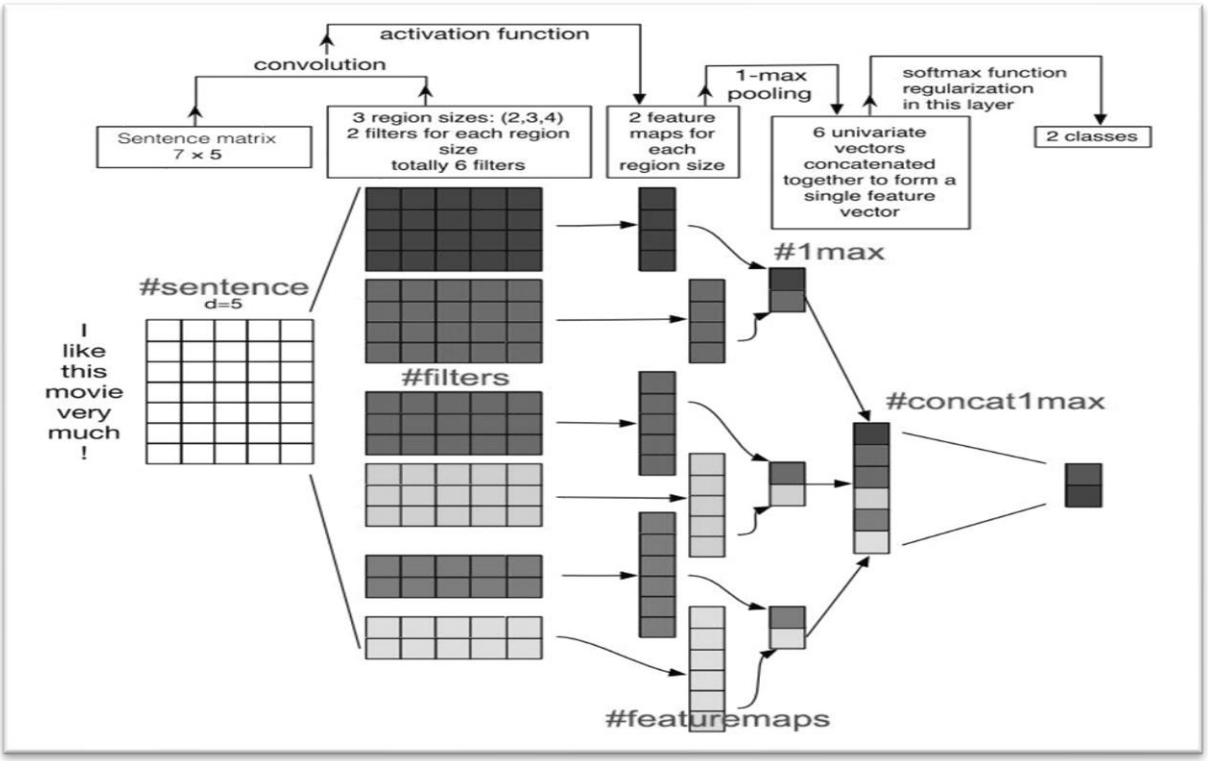


Figure 3.5 Illustration d'une architecture CNN pour la classification de texte.

3.5.6 Réseaux de Neurones Récurrents (RNN)

Les réseaux de neurones récurrents (RNN) sont des algorithmes largement utilisés dans le domaine de l'apprentissage profond (DL), en particulier dans le traitement du langage naturel (NLP) et de la parole. Contrairement aux réseaux de neurones classiques, les RNN sont conçus pour traiter des données séquentielles, ce qui leur permet de capturer les informations contextuelles importantes présentes dans la séquence de données. Cette capacité à prendre en compte le contexte séquentiel est essentielle pour de nombreuses applications, comme la compréhension du sens des mots dans une phrase en fonction du contexte global[47].

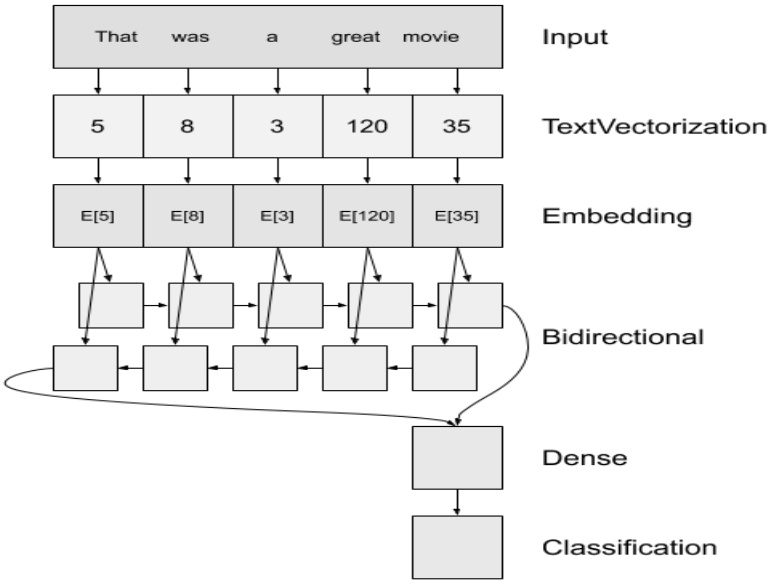


Figure 3.6 Schéma d'une architecture RNN.

La première couche est l'encodeur, qui transforme le texte en une séquence d'indices de jetons. Ensuite, il y a une couche d'intégration où chaque mot est représenté par un vecteur. Ces vecteurs sont entraînables et permettent de capturer les similarités sémantiques entre les mots. Le RNN traite ensuite la séquence d'entrée en itérant à travers chaque élément. Les sorties de chaque pas de temps sont transmises à l'entrée du pas de temps suivant. Enfin, le RNN convertit la séquence en un seul vecteur, puis deux couches denses sont utilisées pour effectuer un traitement final et convertir cette représentation vectorielle en un seul logit comme sortie de classification[48].

3.6 Evaluation

L'efficacité des algorithmes de classification est généralement estimée sur la base de mesures telles que la précision (Precision), le rappel (Recall), le score F1 (F1-Score) et l'exactitude (Accuracy). Ces métriques peuvent d'être calculés à partir de la matrice de confusion.

La matrice de confusion présente une comparaison entre les valeurs prédites par un modèle et les valeurs réelles dans un ensemble de données de test. La matrice est organisée en lignes et en colonnes (voir la figure ci-dessous), où chaque ligne représente les occurrences réelles d'une classe et chaque colonne représente les occurrences prédites par le modèle. Les cellules diagonales de la matrice indiquent les prédictions correctes, tandis que les cellules hors-diagonales représentent les

erreurs de classification.

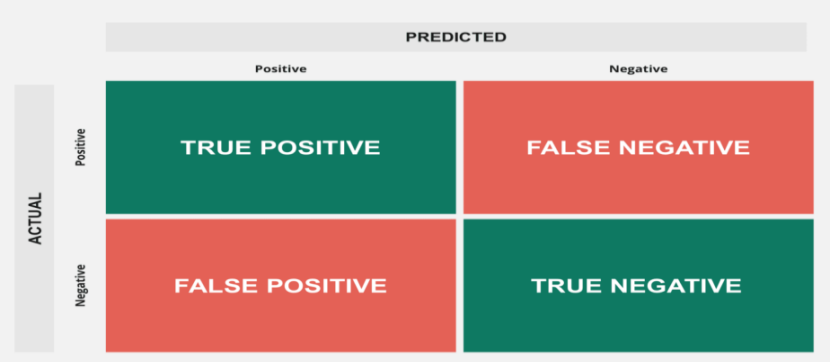


Figure 3.7 Matrice de confusion.

3.7 Conclusion

Dans ce chapitre, nous avons détaillé les différentes étapes que nous avons suivies pour développer notre application. Ces étapes comprennent le chargement des données, le prétraitement des données, ainsi que l'extraction des fonctionnalités, toutes visant à concevoir le meilleur modèle et à obtenir des performances optimales. Nous avons également exposé les approches méthodologiques que nous avons adoptées.

Dans le chapitre suivant, nous présenterons les implémentations concrètes, les résultats obtenus et les discuterons en détail.

Chapitre 04 : Implémentation et Résultats

4.1 Introduction

Tout au long du chapitre précédent, nous avons présenté les différentes étapes de conception et développement de notre projet de fin d'études. Ce chapitre est consacré à donner la description des étapes de réalisation de l'application de détection du langage offensant sur les médias sociaux.

4.2 Environnement et outils de travail

4.2.1 Environnement matériel

L'environnement matériel dans lequel notre application était développé est caractérisé par :

Tableau 1 Environnement de travail

N °	**Modèle PC**	**Processeur**	**Système D'exploitation**	**RAM**
Poste de travail	LENOVO	AMD Ryzen 5	Windows 11	8GB

4.2.2 Langage de programmation

Pour mettre en œuvre notre application de détection du langage offensant, nous avons opté pour Python version 3.12. Python est un langage de programmation orienté objet de haut niveau, reconnu pour sa simplicité et sa facilité d'utilisation. Son interactivité et sa nature interprétable en font un choix idéal, car les instructions sont traduites en langage machine que l'ordinateur peut exécuter en temps réel. Python est réputé pour sa facilité d'apprentissage par rapport à d'autres langages, offrant aux utilisateurs la possibilité de développer des programmes de grande qualité.

4.2.3 Éditeur de code

Pour éditer le code de notre système, nous avons utilisé **Pycharm-Community Edition**. C'est un est un environnement de développement intégré (IDE) utilisé pour la programmation en Python. Il a été développé par l'entreprise tchèque JetBrains et est disponible pour Windows, macOS et Linux. Il est disponible en deux versions, PyCharm Pro et PyCharm Community, et prend en charge le flux de travail Python complet dans cette dernière, y compris les frameworks Web, les technologies frontales, les bases de données et les outils scientifiques.

Figure 4.1 Éditeur de code et biothèques python utilisés.

4.2.4 Librairies et bibliothèques Python

- **NLTK :** Natural Language Toolkit est une bibliothèque logicielle en Python permettant un traitement automatique des langues, Une variété de tâches peuvent être effectuées à l'aide de NLTK, telles que la tokenisation,la supression des mots vides et la visualisation de l'arbre d'analyse, etc[49].
- **PYARABIC :** Une bibliothèque Python dédiée à la langue arabe offre des fonctions essentielles pour manipuler les lettres et le texte arabes. Elle permet notamment de détecter les lettres arabes, d'identifier les groupes et les caractéristiques des lettres, ainsi que de supprimer les signes diacritiques[50].
- **KERAS :** Keras est une bibliothèque open source qui fournit une interface Python pour les réseaux de neurones artificiels. Servant d'interface pour la bibliothèque TensorFlow,

Keras est conçu pour permettre une expérimentation rapide avec les réseaux de neurones profonds. Elle met l'accent sur la convivialité, la modularité et l'extensibilité[51].

- **SKLEARN :** Sklearn est probablement la bibliothèque la plus utile pour l'apprentissage automatique en Python. Elle offre une gamme complète d'outils performants pour l'apprentissage automatique et la modélisation statistique, incluant la classification, la régression, le clustering et la réduction de la dimensionnalité[52].
- **MATPLOTLIB :** est une bibliothèque complète pour créer des visualisations statiques, animées et interactives en Python, il offre une alternative open source viable à MATLAB.

4.3 Application de détection du langage offensant

Nous avons développé une interface graphique (GUI) pour la détection du langage offensant en arabe en utilisant la bibliothèque Python **CustomTkinter**. Cette application offre plusieurs fonctionnalités, telles que le chargement des données, le prétraitement des données, et l'application de techniques d'apprentissage automatique et d'apprentissage profond pour la détection du langage offensant.

Dans ce qui suit, nous illustrerons l'application avec des figures et expliquerons en détail chacune de ses fonctionnalités.

Figure 4.2 Interface principale de l'application.

❖ Chargement des données

Figure 4.3 Chargement de données.

L'application comprend une section de chargement des données où les utilisateurs peuvent sélectionner un fichier de base de données (Excel ou CSV) via une boîte de dialogue de fichier. Le chemin du fichier sélectionné est affiché dans un champ de saisie et imprimé sur la console. Si aucun fichier n'est sélectionné, un message indiquant « Aucun fichier sélectionné » est imprimé. Cette fonctionnalité de chargement de données constitue une première étape essentielle dans le processus de détection du langage offensant dans le contenu des réseaux sociaux.

❖ Prétraitement des données

Figure 4.4 Prétraitement des données.

Cette partie de l'application est conçue pour prétraiter les données textuelles chargées. Elle commence par définir plusieurs fonctions de prétraitement qui effectuent des tâches telles que la suppression des mots vides, des caractères spéciaux, de la ponctuation et des emojis, ainsi que la normalisation du texte arabe. Une fois les données prétraitées, elles sont enregistrées dans un nouveau fichier Excel. Enfin, les données prétraitées sont affichées dans une nouvelle fenêtre sous forme de tableau. Ce processus complet permet un prétraitement et une visualisation efficaces et efficients des données textuelles arabes.

❖ Informations sur les données

Le bouton **Afficher** affiche la base de données chargée sous forme de tableau. Le bouton **Infos** fournit des informations détaillées sur les données, telles que le nombre de lignes et de colonnes, la longueur du texte et les détails des colonnes. Le bouton **Word Cloud** génère un nuage de mots des 200 premiers mots prétraités du base de données. Enfin, le bouton **N-gram** affiche ses bi-grammes et trigrammes.

Figure 4.5 Informations sur les données.

❖ **Detection du langage offensant**

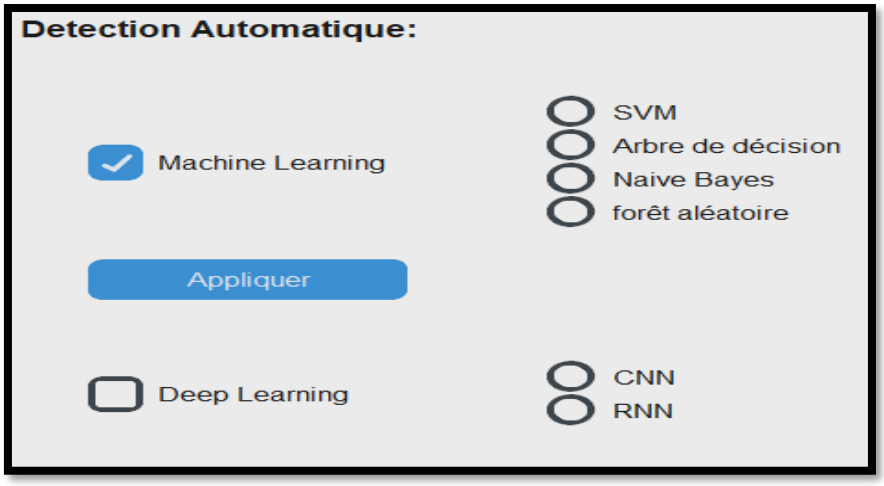


Figure 4.6 Detection du langage offensant.

Dans cette partie de l'application, nous avons l'option de choisir entre deux approches d'apprentissage : **Machine Learning (ML)** et **Deep Learning (DL)**. Les boutons radio sur l'interface de la détection représentent les méthodes implémentées dans notre application pour chaque approche (SVM, Decision Tree, Naive Bayes et Random Forest pour l'approche de

Machine Learning **(ML)**; CNN et RNN pour l'approche de Deep Learning**)**. Il suffit ensuite de cliquer sur le bouton **Appliquer** pour lancer une méthode de classification et afficher enfin les résultats obtenus.

❖ **Classification du texte**

Notre application offre la possibilité de vérifier si un texte donné est offensant. La séction de classification contient un champ de saisie de texte dans lequel l'utilisateur peut saisir un texte. De plus, il existe un bouton intitulé « **Vérifier** » qui, une fois cliqué, analysera le texte saisi pour détecter tout langage offensant et affichera les résultats appropriés (tels que « propre», « offensant », « obscène »).

Figure 4.7 Prédiction du texte.

4.4 Étude de cas

4.4.1 Informations générales

Le tableau suivant présente les informations générales de nos données, nombre de lignes et nombre de colonnes et les types de données.

Tableau 2 Informations générales

L'ensemble de données	
Nombre de lignes	31100

Nombres de colonnes	8
Type de données	Int64, object.
Noms des colonnes	Text , Aggregated Annotation..
La taille De la base de données	117 KO

4.4.2 Distribution de classes

Les 31100 commentaires du jeu de données sont distribués sur trois classes :

- 6276 commentaires sont de type **Propre**.
- 757 commentaires sont de type **Offensant.**
- 24067 commentaires sont de type O**bscènes**.

Le tableau suivant présente le nombre de commentaires dans chaque catégorie.

Tableau 3 Statistiques sur les données.

Jeu de données	**Catégorie**	**Nombre de Commentaires**
TweetClassification-Summary + AJCommentsClassification-CF	Propre	6276
	Offensif	757
	Obscène	24067
	Totale	31100

La figure ci-dessous montre la répartition statistique des catégories de notre base de données.

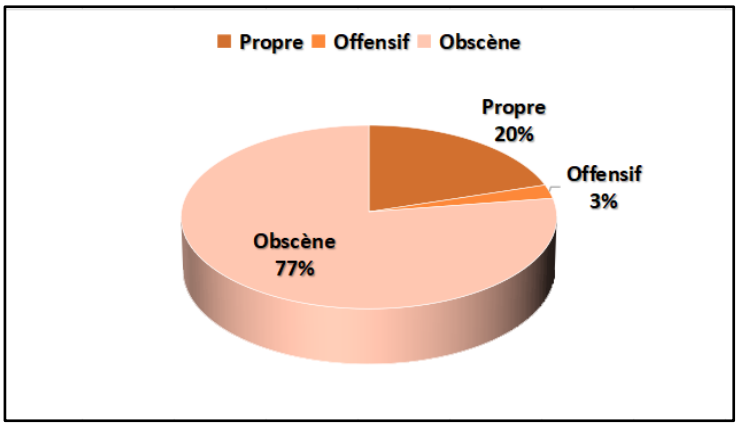


Figure 4.8 Répartition des catégories.

4.4.3 Fréquences des termes (Word Cloud et N-Grammes)

Un nuage de mots (ou « wordcloud » en anglais) est un outil de visualisation qui permet, à travers une image, de percevoir rapidement les mots les plus fréquents dans un texte ou un corpus de texte. La figure 9 montre les mots les plus fréquents dans notre corpus.

Figure 4.9 Les mots les plus fréquents dans l'ensemble de données

Le N-Gramme est une séquence contiguë de n éléments provenant d'un échantillon donné de texte ou de parole. Les éléments peuvent être des syllabes, des lettres ou des mots selon l'application. Les figures 10 et 11 montrent les bigrammes et les trigrammes fréquents dans notre jeu de données.

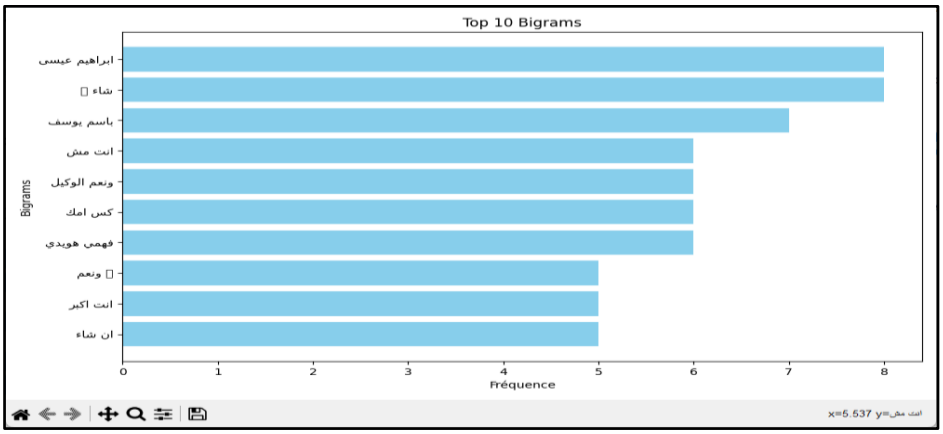


Figure 4.10 Les bigrammes les plus fréquents dans l'ensemble de données.

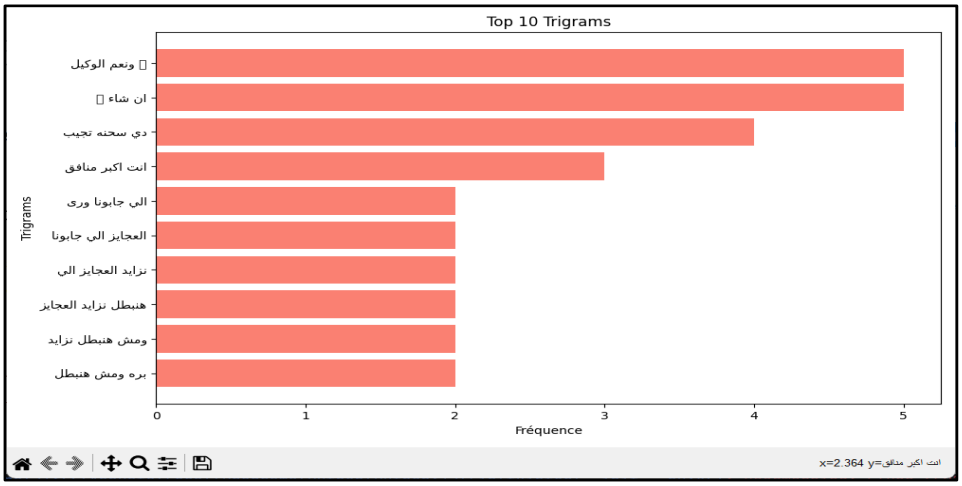


Figure 4.11 Les trigrammes les plus fréquents dans l'ensemble de données.

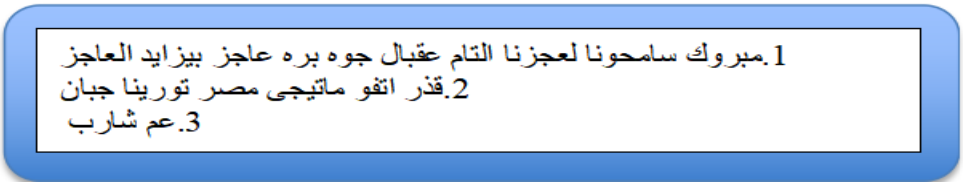
4.4.4 Nettoyage et prétraitement

Nous avons appliqué les fonctions de nettoyage et de prétraitement évoquées précédemment à notre corpus. L'exemple suivant montre les textes avant et après le prétraitement. :

❖ **Texte brute**

1 مبروك و سامحونا لعجزنا التام. عقبال اللي جوه. اللي بره يا عاجز يا بيزايد على العاجز.
2.قذر اتفوو ماتيجى مصر وتورينا نفسك كدا ياجبان
3.يا عم انت شارب أيه؟؟؟

❖ **Texte pré-traité**



4.4.5 Vectorisation

La figure (12) ci-dessous présente un exemple d'encodage de données avec BoW, généré par notre application.

	جوه	جابونا	بيزايد	بره	بدل	المجايز	العاجز	الصحرا	التام	ازرع
0	0	1	0	1	0	0	1			
1	0	0	0	0	0	1	0	1		
2	1	0	1	0	0	0	1	1		

	وري	هنبطل	نزايد	ميروك	لعجزنا	كلنا	قاعد	عقبال	عاجز	سامحونا
0	1	1	1	0	0	1	1			
1	0	0	0	0	1	0	0			
2	0	0	0	1	0	0	0			

Figure 4.12 Présentation des données à l'aide de BoW.

La figure (13) ci-dessous présente un exemple d'encodage de données avec TF-IDF, généré par notre application.

	بره	بدل	المجايز	العاجز	الصحرا	التام	ازرع \
0	0.000000	0.327055	0.000000	0.327055	0.000000	0.000000	
1	0.000000	0.000000	0.000000	0.000000	0.396875	0.000000	
2	0.479528	0.000000	0.479528	0.000000	0.000000	0.479528	

	قاعد	عقبال	عاجز	سامحونا	جوه	جابونا	بيزايد \
0	0.327055	0.000000	0.327055	0.327055	0.327055	0.327055	
1	0.000000	0.396875	0.000000	0.000000	0.000000	0.000000	
2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	

	وري	هنبطل	نزايد	ميروك	لعجزنا	كلنا
0	0.000000	0.327055	0.327055	0.000000	0.000000	0.000000
1	0.396875	0.000000	0.000000	0.396875	0.396875	0.396875
2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Figure 4.13 Présentation des données à l'aide de TF-IDF.

4.5 Classification du texte offensant

La classification du texte offensant a été effectuée en utilisant les classificateurs de base et les réseaux de neurones implémentées par notre application :

- SVM avec un noyau RBF (Radial Basis Function)
- Arbre de décision avec le critère « Gini» comme fonction de mesure de la qualité de la division.
- Naive de Bayes.
- Forêt Aléatoire avec un nombre d'arbres égale à 100 (n_estimators =100) et utilisant l'agrégation par bootstrap (bagging), l'indice de Gini pour mesurer la qualité de la division.
- Un réseau de neurone convolutif (CNN) avec couche de convolution (voir Tableau 4).
- Un réseau de neurones récurrent (RNN) un simple RNN avec une simple couche LSTM (voir Tableau 5).

Tableau 4 Les paramètres du classificateur CNN

Couche (type)	**Forme de sortie**	**Nombre de paramètres**
Embedding (Embedding)	(None, 512, 50)	500,100
Conv1D (Conv1D)	(None, 508, 128)	32,128
GlobalMaxPooling1D (GlobalMaxPooling1D)	(None, 128)	0
Dense (Dense)	(None, 256)	33,024
Dropout (Dropout)	(None, 256)	0
Dense (Dense)	(None, 3)	771

Tableau 5 Les paramètres du classificateur RNN

Couche (type)	Forme de sortie	Nombre de paramètres
Embedding (Embedding)	(None, 1024, 100)	1.000.000
lstm(LSTM)	(None, 128)	117.248
dense (Dense)	(None, 256)	33.024
dropout (Dropout)	(None, 256)	0
Dense_1(Dense)	(None, 3)	771

4.6 Résultats et évaluation

4.6.1 La performance des classificateurs de base

❖ Avec BOW

Le tableau 6 présente la performance moyenne des classificateurs de base utilisant la technique de vectorisation de texte BOW. Le tableau 7 également présente leurs performance exacte par rapport à chacune des trois catégories «Offensif», «Obscene» , et «Propre».

La performance a été mesurée à l'aide des métriques : Exactitude (Accuracy), Précision (Precision), Rappel (Recall) et score F1 (F1-Score).

Tableau 6 Performance des classificateurs utilisant BOW

Classificateur	Accuracy	Precision	Recall	F1-Score
SVM	80.3%	81.2%	80.33%	**80.76%**
Arbre de Décision	72%	74%	72,09%	73%
Naive de Bayes	80.78%	78.77%	80.78%	79.76%
Forêt Aléatoire	77.75%	76.14%	77.57%	76.84%

Tableau 7 Performance des classificateurs utilisant BOW selon chaque catégorie

Méthodes/Catégories	Offensive			Obscene			Propre		
	Prec.	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
SVM	88.88%	15.28%	26%	80.26%	**99.62%**	88.90%	**84.4%**	7%	13.39%
Arbre de Décision	61.67%	**40.12%**	48.64%	**82.87%**	85.85%	84.34%	35.28%	**31.46%**	33.26%
Naive de Bayes	**89.47%**	21.65%	34.87%	81.37%	98.23%	**89.01%**	66.66%	14.70%	24%
Forêt Aléatoire	83.58%	35.66%	**50%**	**82.86%**	92.25%	87.3%	45.53%	27%	**33.92%**

❖ Avec TF-IDF

Le tableau 8 présente la performance moyenne des classificateurs de base utilisant la technique de vectorisation de texte TF-IDF. Le tableau 9 également présente leurs performance exacte par rapport à chacune des trois catégories «Offensif», «Obscene» , et «Propre».

La performance a été mesurée à l'aide des métriques : Exactitude (Accuracy), Précision (Precision), Rappel (Recall) et score F1 (F1-Score).

Tableau 8 Performance des classificateurs utilisant TF-IDF

Classificateur	Accuracy	Precision	Recall	F1-Score
SVM	**81.04%**	80.81%	**81.04%**	**80.92%**
Arbre de Décision	71.91%	74.23%	71.91%	73.05%
Naive de Bayes	79.24%	**81.24%**	79.24%	**80.22%**
Forêt Aléatoire	79.46%	76.97%	79.46%	78.19%

Tableau 9 Performance des classificateurs utilisant TF-IDF selon chaque catégorie

Méthodes/Catégories	Offensive			Obscene			Propre		
	Prec.	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
SVM	**91.42%**	20.38%	33.33%	80.86%	**99.43%**	**89.19%**	83.33%	10.76%	18.92%
Arbre décision	53.44%	**39.49%**	45.42%	**82.85%**	84.80%	83.81%	34.49%	**32.17%**	**33.29%**
Naive de Bayes	79%	38%	**52%**	79.2%	**100%**	88.39%	**100%**	1%	3%
Foret Aléatoire	88.88%	30.57%	45.49%	**82.13%**	96.58%	88.77%	59.06%	20%	29.97%

4.6.2 La performance des modèles d'apprentissage profond

Le tableau 10 et 11 présentent la performance du modèle CNN et du modèle RNN respectivement à travers les trois catégories de notre jeu de données.

Tableau 10 performance du modèle CNN

Catégorie	Precision	Recall	F1-score	Accuracy	Nombre d'erreur de prédiction
Offensive	48%	38%	42%	76,6%	1540
Propre	44%	58%	50%		
Obscene	84%	87%	86%		
Total	58.67%	61%	59.33%		

Tableau 11 Performance de RNN

Catégorie	Precision	Recall	F1-score	Accuracy	Nombre d'erreur de prédiction

Offensive	63%	54%	55%	76,3%	1600
Propre	44%	61%	51%		
Obscene	84%	88%	86%		
Total	63.67%	67.67%	64%		

Les figures 14 et 15 montrent le développement de l'exactitude (Accuracy) des deux modèles et de leurs fonctions de pertes par rapport au nombre d'époques (Epochs) déterminé pour l'apprentissage.

Figure 4.14 L'exactitude et la perte de la fonction d'apprentissage par rapport au nombre époques pour le modèle CNN.

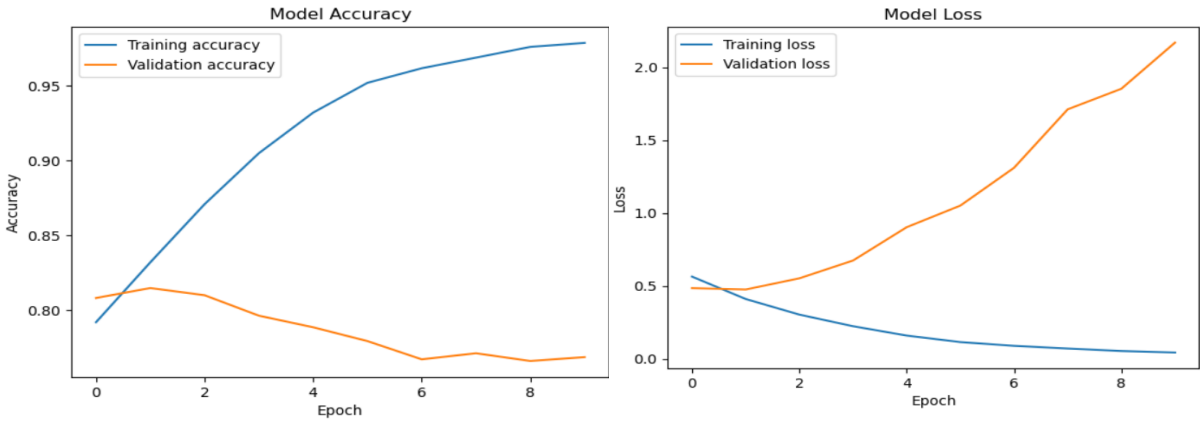


Figure 4.15 L'exactitude et la perte de la fonction d'apprentissage par rapport au nombre époques pour le modèle RNN.

4.7 Discussion

❖ Classificateurs de base avec BOW

Pour la tâche de classification du langage offensant en arabe, divers classificateurs montrent des niveaux d'efficacité différents. Le SVM montre une bonne performance globale avec la précision la plus élevée parmi les classificateurs. Connu pour son efficacité dans les espaces de haute dimension et sa robustesse face au surapprentissage, surtout avec des fonctions noyau appropriées, le SVM est bien adapté pour classer des données textuelles, comme le langage offensant en arabe, qui peut être de haute dimension en raison de la riche morphologie de la langue. Il excelle en précision, indiquant que lorsque le SVM classe un message dans une certaine catégorie (par exemple, obscène), il est probable que ce soit correct. Le fort rappel démontre que le SVM identifie efficacement la plupart des instances pertinentes dans chaque catégorie. L'équilibre entre la précision et le rappel, reflété dans le score F1, souligne sa fiabilité en termes de justesse et de complétude.

D'autre part, le Naive de Bayes a une précision légèrement inférieure à celle du SVM mais un rappel comparable, avec une précision légèrement inférieure. Particulièrement adapté à la classification de textes en raison de sa nature probabiliste et de l'hypothèse d'indépendance des caractéristiques, qui souvent se vérifie bien dans la représentation bag-of-words couramment

utilisée en classification de texte, le Naive de Bayes montre une forte performance, avec le score F1 indiquant un bon équilibre entre précision et rappel.

Le classificateur Arbre de Décision, cependant, montre les performances les plus faibles parmi les classificateurs. Les arbres de décision ont tendance à surajuster, surtout avec des données complexes et bruyantes comme les textes en langue naturelle. La performance relativement inférieure peut être due à un surajustement ou à l'incapacité du modèle à capturer les nuances de la langue aussi efficacement que les autres modèles.

Les Forêts Aléatoires fonctionne mieux que les arbres de décision mais moins bien que le SVM et le Naive Bayes. En tant que méthode d'ensemble, le Random Forest réduit le surajustement observé dans les arbres de décision. Malgré sa robustesse, sa performance peut encore être affectée par la complexité des données et le besoin d'un plus grand nombre d'arbres ou d'un meilleur ajustement des hyperparamètres pour obtenir de meilleurs résultats. Il trouve un équilibre mais n'excelle dans aucune métrique particulière par rapport au SVM et au Naive Bayes.

Deux notes importantes doivent être ajoutées à cette discussion. Premièrement, l'impact du déséquilibre des classes : la dominance de la catégorie obscène fausse les performances de tous les classificateurs vers une meilleure performance sur cette catégorie, aux dépens des catégories offensives et propres. Deuxièmement, la difficulté avec les catégories minoritaires : les catégories offensive et propre sont sous-représentées, ce qui entraîne de faibles scores de rappel et de F1 pour tous les classificateurs, indiquant que de nombreuses instances de ces catégories sont manquées.

En conclusion, le Naive de Bayes et le SVM sont tous deux de forts candidats pour cette tâche. Le Naive de Bayes est légèrement meilleur en précision, tandis que le SVM excelle en précision. Les arbres de décision sont moins adaptés sans optimisation supplémentaire, et les Forêts Aléatoires offrent un juste milieu mais ne surpassent pas les performances du Naive de Bayes et du SVM. En considérant ces forces et ces faiblesses, ainsi que l'impact du déséquilibre des classes et les difficultés rencontrées avec les catégories minoritaires, le SVM et le Naive de Bayes émergent comme les meilleurs choix pour cette tâche de classification, compte tenu de leur capacité à gérer la nature de haute dimension des données textuelles et leur performance globale équilibrée.

- **Classificateurs de base avec TF-IDF**

L’utilisation de TF-IDF (Term Frequency-Inverse Document Frequency) au lieu des représentations Bag-of-Words (BoW) avec les classificateurs de base semble ne pas garantir nécessairement une meilleure performance. Bien que TF-IDF prenne en compte l'importance des termes dans le corpus en les pondérant en fonction de leur fréquence et de leur rareté à travers les documents, les résultats expérimentaux indiquent que cette approche nuancée n’améliore pas systématiquement la précision de classification ou d’autres métriques de performance. Malgré sa capacité à potentiellement capturer des caractéristiques plus significatives et à réduire l’impact des termes courants, TF-IDF peut ne pas offrir d’avantages significatifs par rapport à BoW dans ce contexte. Par conséquent, son adoption en tant que représentation alternative doit être soigneusement évaluée, en tenant compte des caractéristiques spécifiques de l'ensemble de données et de la nature de la tâche de classification.

- **Réseaux de neurones CNN et RNN**

Pour la tâche de classification du langage offensant en arabe, le CNN et le RNN montrent une précision globale similaire, avec le CNN à 76,6 % et le RNN à 76,3 %. Les deux modèles performent bien dans la catégorie dominante des messages obscènes, mais rencontrent des difficultés dans les catégories normales et offensives.

Le CNN excelle dans la capture des motifs locaux, ce qui explique sa performance élevée dans la catégorie obscène, mais il peut avoir du mal avec les dépendances séquentielles, impactant sa capacité à traiter les classes minoritaires comme les messages offensifs. En revanche, le RNN est conçu pour gérer les données séquentielles et capturer les dépendances à long terme, ce qui le rend plus efficace pour identifier les messages offensifs, bien que cela puisse augmenter la complexité computationnelle et entraîner une plus grande variabilité des performances.

En somme, le CNN peut être préféré pour une performance équilibrée grâce à sa précision légèrement supérieure et à son nombre réduit d'erreurs, tandis que le RNN est supérieur pour identifier les instances de la catégorie minoritaire offensive. Pour améliorer la performance globale, en particulier dans les classes minoritaires, des techniques telles que l'augmentation des données, la pondération des classes ou des modèles hybrides combinant les architectures CNN et RNN pourraient être explorées.

4.8 Conclusion

Dans ce chapitre, nous avons présenté les résultats que nous avons obtenus à partir des expériences que nous avons menées sur notre ensemble de données, en utilisant six classificateurs. On peut noter que chaque algorithme a sa capacité intrinsèque à surpasser les autres algorithmes en fonction de la situation.

Conclusion Générale

L'objectif principal de notre travail était de détecter le langage offensif dans le contenu textuel en langue arabe, en particulier sur les réseaux sociaux, afin de fournir une solution efficace à ce problème qui affecte différentes catégories de personnes. Pour atteindre cet objectif, nous avons proposé une approche basée sur les domaines de l'Intelligence Artificielle, Spécialement de l'Apprentissage Automatique et de l'Apprentissage Profond, en appliquant différentes méthodes de classification des textes.

Nous avons préparé notre dataset qui contient ensembles de données collectés à partir de Twitter, avec trois tâches distinctes: l'offensant et l'obscene et propre. Ces ensembles de données comprenaient des tweets en arabe standard ainsi que des dialectes variés, afin d'atteindre notre objectif de couvrir un large spectre de la langue arabe. Ensuite, nous avons utilisé des techniques de traitement du langage naturel, lors de l'étape de prétraitement des données, au cours desquelles nous avons nettoyé notre ensemble de données du bruit, normalisé certaines lettres, supprimé les mots vides et vectoriser nos données en utilisant TF-IDF et BOW. Enfin, différents modèles d'apprentissage automatique et profond sont appliqués pour la détection automatique du langage offensant : quatre classificateurs d'apprentissage automatique ont été testés : Machines à Support de Vecteurs, Naïve Bayes, Forêt Aléatoire et Arbre de Décision.

Pour améliorer les performances et obtenir de meilleurs résultats, les travaux futurs incluront l'exploration de nouvelles approches de prétraitement des commentaires et d'algorithmes d'apprentissage. Des idées à tester pourraient inclure l'augmentation de la taille de l'ensemble de données en intégrant des données provenant de différents domaines et plateformes pour améliorer la précision de la classification, l'application d'un correcteur orthographique pour éliminer les fautes de frappe fréquentes dans les commentaires des utilisateurs, et l'utilisation d'algorithmes d'apprentissage plus avancés pour améliorer les capacités de classification.

Les références

[1] Chilwant, N., Rizvi, S. T. A., & Soliman, H. (2022). Offensive Language Detection on Twitter. ArXiv, 2209.14091.

[2]Oxford dictionary.https://www.oxfordlearnersdictionaries.com/definition/english/offensiv

[3] Collins dictionary. https://www.collinsdictionary.com/dictionary/english/ offensive.

[4] "A Review on Offensive Language Detection" par Rahul Pradhan, Ankur Chaturvedi, Aprna Tripathi et Dilip Kumar Sharma.

[5] "Exploring the Impact of Verbal Abuse on Recovery: A Mediation Study" par Mark Salzer et Nirit Karni-Vizer.

[6] Joni Salminen, Maximilian Hopf, Shammur A Chowdhury, Soon-gyo Jung, Hind Almerekhi, and Bernard J Jansen. Developing an online hate classifier for multiple social media plat forms. Human-centric Computing and Information Sciences, 10(1):1, 2020.

[7] Muhammad Okky Ibrohim and Indra Budi. A dataset and preliminaries study for abusive language detection in indonesian social media. Procedia Computer Science, 135:222–229, 2018.

[8]: Cervone, C., Augoustinos, M., & Maass, A. (2020). The Language of Derogation and Hate: Functions, Consequences, and Reappropriation. Journal of Language and Social Psychology, 40(1), 80-101.

[9] Jay, T., & Janschewitz, K. (2012). The Science of Swearing. Association for Psychological Science – APS[1].

[10] Cervone, C., Augoustinos, M., & Maass, A. (2020). The Language of Derogation and Hate: Functions, Consequences, and Reappropriation. Journal of Language and Social Psychology, 40(1), 80-101.

[11] https://time.com/4602680/profanity-research-why-we-swear.

[12] https://journals.sagepub.com/doi/pdf/10.1177/0261927X20967394

[13] Wikipedia. (2024). Quebec City mosque shooting[5]. Cet article de Wikipedia donne un aperçu détaillé de l'attaque contre la mosquée de Québec en 2017.

[14] violences contre les migrants et les réfugiés : OHCHR. (2021).

[15]"Fighting Adversarial Attacks on Online Abusive Language Moderation" par Nestor Rodriguez et Sergio Rojas-Galeano.

[16] Gongane, V. U., Munot, M. V., & Anuse, A. D. (2022). Detection and moderation of detrimental content on social media platforms: current status and future directions. Social Network Analysis and Mining, 12, Article 129.

[17] "Hate speech detection: Challenges and solutions" publié dans PLOS ONE2 .

[18] Kebriaei, E., Homayouni, A., Faraji, R., Razavi, A., Shakery, A., Faili, H., & Yaghoobzadeh, Y. (2023). Persian offensive language detection. *Machine Learning*[1]

[19] Liu, J., Yang, Y., Fan, X., Ren, G., Yang, L., & Ning, Q. (2022). Offensive-Language Detection on Multi-Semantic Fusion Based on Data Augmentation[1].

[20] Wei, B., Li, J., Gupta, A., Umair, H., Vovor, A., & Durzynski, N. (2021). Offensive Language and Hate Speech Detection with Deep Learning and Transfer Learning.

[21] Keraghel, I., Morbieu, S., Nadif, M. (2024). A survey on recent advances in Named Entity Recognition.

[22] Ranasinghe, T., & Zampieri, M. (2023). A Text-to-Text Model for Multilingual Offensive Language Identification.

[23] Pradhan, R., Chaturvedi, A., Tripathi, A., & Sharma, D. K. (2020). A Review on Offensive Language Detection.

[24] Jiang, A., & Zubiaga, A. (2024)Cross-lingual Offensive Language Detection: A Systematic Review of Datasets, Transfer Approaches and Challenges.

[25] https://fr.statista.com/infographie/14919/langues-les-plus-parlees-dans-le monde-et-les-plus-utilisees-sur-internet/

[26] Al-Sallab, A. M., Elmadany, A., El-Beltagy, S. R., & Rafea, A. (2019). Arabic Offensive Language Detection with Attention-based Deep Neural Networks

[27] Al-Sallab, A. M., Elmadany, A., El-Beltagy, S. R., & Rafea, A. (2020). Enhancing Arabic offensive language detection with BERT-BiGRU model.

[28] Mubarak, H., Darwish, K., & Magdy, W. (2017). Arabic Offensive Language on Twitter: Analysis and Experiments.

[29] Mubarak, H., Darwish, K., & Magdy, W. (2019). Towards Accurate Detection of Offensive Language in Online Communication in Arabic.

[30] Hamdy Mubarak, Ammar Rashed, Kareem Darwish, Younes Samih, and Ahmed Abde lali. Arabic offensive language on twitter: Analysis and experiments. arXiv preprint arXiv:2004.02192, 2020.

[31] Hamdy Mubarak and Kareem Darwish. Arabic offensive language classification on twitter. In International Conference on Social Informatics, pages 269–276. Springer, 2019.

[32] HananeMohaouchane, AsmaaMourhir, and Nikola S Nikolov. Detecting offensive language on arabic social media using deep learning. In 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), pages 466–471. IEEE, 2019.

[33] Fatemah Husain. Arabic offensive language detection using machine learning and ensemble machine learning approaches. arXiv preprint arXiv:2005.08946, 2020.

[34] Azalden Alakrot, Liam Murray, and Nikola S Nikolov. Towards accurate detection of offensive language in online communication in arabic. Procedia computer science, 142:315 320, 2018.

[35] Amr Keleg, Samhaa R El-Beltagy, and Mahmoud Khalil. Asu_opto at osact4-offensive language detection for arabic text. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, pages 66–70, 2020.

[36] Bushr Haddad, Zoher Orabe, Anas Al-Abood, and Nada Ghneim. Arabic offensive language detection with attention-based deep neural networks. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, pages 76–81, 2020.

[37] Sabit Hassan, Younes Samih, Hamdy Mubarak, Ahmed Abdelali, Ammar Rashed, and Sham mur Absar Chowdhury. Alt submission for osact shared task on offensive language detec tion. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, pages 61–65, 2020.

[38] Zadorozhnyy, A., & Yu, B. (2021). Addressing the impact of informal language learning practices in digital wilds on the development of L2 digital literacies. In N. Zoghlami, C. Brudermann, C. Sarré, M. Grosbois, L. Bradley, & S. Thouësny (Eds), CALL and professionalisation: short papers from EUROCALL 2021 (pp. 307-311).

[39] Jiang, A., & Zubiaga, A. (2024) discusses the limitations of current datasets used in offensive language detection.

[40] Jiang, A., & Zubiaga, A. (2024). Cross-lingual Offensive Language Detection: A Systematic Review of Datasets, Transfer Approaches and Challenges

[41] DTPCAM Janssens. Natural language processing in requirements elicitation and require ments analysis: a systematic literature review. Master's thesis, 2019.

[42] Jiawei Han, Jian Pei, and Micheline Kamber. Data mining: concepts and techniques. Elsevier, 2011.

[43] Ethem Alpaydin. Radial basis functions, 2010.

[44] Gurinder Singh, Bhawna Kumar, Loveleen Gaur, and Akriti Tyagi. Comparison between multinomial and bernoulli naïve bayes for text classification. In 2019 International Confer ence on Automation, Computational and Technology Management (ICACTM), pages 593–596. IEEE, 2019.

[45] Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5-32.

[46] Aliaa Rassem, Mohammed El-Beltagy, and Mohamed Saleh. Cross-country skiing gears classification using deep learning. arXiv preprint arXiv:1706.08924, 2017.

[47] CNN & RNN: https://link.springer.com/content/pdf/10.1186/s40537-021-00444-8.pdf

[48] https://www.tensorflow.org/text/tutorials/text_classification_rnn#create_the_text_encder.

[49] https://fr.wikipedia.org/wiki/Natural_Language_Toolkit.

[50] pyarabic https://pyarabic.sourceforge.io/

[51] keras Developer guides https://keras.io/guides/.

[52] scikit-learn User Guide https://scikit-learn.org/stable/user_guide.html