République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université 8 Mai 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 Technologies**
Filière : **Télécommunications**
Spécialité : **Réseaux et Télécommunications**

Identification du contenu offensif écrit en Bambara

Présenté par :

BAH Mamadou Aliou

Sous la direction du :

Dr. ABAINIA Kheireddine

JUIN 2025

Remerciements

Avant toute chose, je tiens à exprimer ma profonde gratitude à **Dr. ABAINIA Kheireddine**, mon encadrant, pour sa disponibilité, son encadrement rigoureux, ses conseils éclairés et sa bienveillance tout au long de ce travail. Son accompagnement a été essentiel à l'aboutissement de ce mémoire.

Je remercie également l'ensemble du corps enseignant du département d'Électronique et Télécommunications et du département des sciences et technologies de l'Université 8 Mai 1945 Guelma, pour la qualité des enseignements dispensés durant ces années de formation.

Mes sincères remerciements vont aussi à mes camarades de promotion pour leur soutien, leurs encouragements, et les échanges constructifs qui ont enrichi ce parcours.

Je n'oublie pas ma famille, en particulier mes parents, pour leur patience, leur confiance et leurs sacrifices constants. Leur soutien moral m'a toujours donné la force d'aller au bout de mes objectifs.

Enfin, je remercie toute personne, de près ou de loin, ayant contribué à la réussite de ce travail.

Résumé

Ce mémoire aborde la détection automatique du contenu offensif en bambara, qui est une langue moins ressourcée. Plusieurs modèles d'apprentissage ont été évalués sur des corpus annotés (équilibrés et déséquilibrés). Les modèles d'apprentissage profonds, notamment CNN, BiLSTM et FastText, ont obtenu les meilleures performances, avec un avantage d'accuracy globale de 88,4 % pour BiLSTM et FastText, surtout sur le corpus déséquilibré. Le CNN, ainsi que l'architecture hybride BiLSTM+CNN, s'est révélé stable et offre de bons compromis sur l'ensemble des corpus.

Les modèles classiques ont également obtenu de bons résultats, notamment SVM et Naive Bayes, avec une accuracy globale de 89 %, mais sont moins performants sur la classe minoritaire du corpus à trois classes, souvent mal détectée (rappel inférieur à 40%). Les modèles profonds, quant à eux, affichent une légère supériorité dans la détection de cette classe, grâce à une meilleure capacité de généralisation. L'utilisation de la technique SMOTE a permis une légère amélioration du rappel pour les modèles classiques.

Malgré les ressources limitées, il est possible de concevoir des systèmes fiables pour des langues peu représentées, à condition d'adapter les approches aux spécificités linguistiques et aux déséquilibres de données.

Mots clés : Contenu offensif, bambara, langue moins ressourcée, détection automatique, apprentissage automatique, apprentissage profond, traitement du langage naturel.

ABSTRACT

This thesis addresses the problem of automatic identification of offensive Bambara content, i.e. a low-resourced language. Several models have been evaluated on annotated corpora (balanced and imbalanced), where deep learning models have achieved the best performance (over 88% of accuracy) on the imbalanced corpus. On the other hand, the combination of BiLSTM+CNN architecture proved to be stable and offered good trade-offs across all corpora.

Traditional machine learning models also performed well, especially SVM and Naive Bayes, reaching an overall accuracy of 89%. However, they are less effective in detecting the minority class in the three-class corpus, which was often poorly recognized (<40% of recall). Deep models showed a slight advantage in detecting this class due to better generalization capabilities. The use of SMOTE technique produced a slight improvement in recall for traditional models.

Despite limited resources, it is possible to design reliable systems for low-resourced languages by adapting this approaches to linguistic specificities and data imbalances.

Keywords: Offensive content, Bambara, low-resourced language, automatic detection, machine learning, deep learning, natural language processing.

Tables des matières

Liste des figures

Liste des tableaux

Liste des abréviations

Adam : Adaptive Moment Estimation

BERT : Bidirectional Encoder Representations from Transformers

BoW : Bag of Words

CNN : Convolutional Neural Network

GPT : Generative Pre-trained Transformer

GRU : Gated Recurrent Unit

K-NN : K-Nearest Neighbors

LSTM : Long Short-Term Memory

NLP : Natural Language Processing

ReLU : Rectified Linear Unit

RNN : Recurrent Neural Network

SGD : Stochastic Gradient Descent

SMOTE : Synthetic Minority Over-sampling Technique

SVM : Support Vector Machine

T5 : Text-to-Text Transfer Transformer

TALN : Traitement Automatique du Langage Naturel

TF-IDF : Term Frequency - Inverse Document Frequency

Introduction Générale

Avec l'explosion des interactions en ligne, les plateformes numériques jouent aujourd'hui un rôle central dans la communication entre les individus. Que ce soit via les réseaux sociaux, les forums ou les services de messagerie, chacun peut librement partager ses idées, opinions et émotions. Cependant, cette liberté d'expression s'accompagne d'un phénomène préoccupant : la multiplication de contenus offensifs, haineux, discriminatoires ou violents. Ce type de langage nuit au climat des échanges, favorise la stigmatisation, et peut avoir de graves conséquences psychologiques, en particulier sur les jeunes publics.

La détection automatique de ces contenus s'est imposée comme un enjeu crucial à la fois technique, éthique et sociétal. De nombreuses recherches ont été menées dans des langues à fortes ressources comme l'anglais, le français ou l'arabe, où les outils linguistiques, corpus annotés et modèles pré-entraînés sont largement disponibles. En revanche, les langues africaines dont le bambara (largement parlée au Mali et en Afrique de l'Ouest) restent peu représentées dans le domaine du traitement automatique du langage naturel (TALN). Cette sous-représentation freine la capacité des outils numériques à comprendre, filtrer ou modérer les contenus rédigés dans ces langues.

Le bambara présente des défis spécifiques : absence de standardisation orthographique, diversité dialectale, rareté des ressources numériques disponibles, écriture souvent phonétique sur les réseaux sociaux, etc. Face à ces contraintes, il devient nécessaire d'exploiter des approches adaptées pour permettre la détection automatique du contenu offensif dans cette langue.

Ce mémoire vise donc à concevoir et évaluer différents modèles de classification textuelle capables d'identifier automatiquement les propos offensants rédigés en bambara. Il s'appuie sur des techniques d'apprentissage automatique et d'apprentissage profond, en adaptant certaines méthodes modernes telles que les modèles à base de Transformers.

Le mémoire est structuré en trois grandes parties : une première partie consacrée à la typologie du contenu offensif et aux défis linguistiques propres au bambara ; une deuxième partie présentant les bases théoriques du TALN et les principaux algorithmes utilisés dans la détection de textes offensifs ; enfin, une troisième partie dédiée à la mise en œuvre expérimentale, à l'analyse des résultats et à l'évaluation des performances des modèles.

Ce travail ambitionne ainsi de contribuer à la recherche sur les langues peu dotées, en proposant des outils concrets pour une modération de contenu plus équitable et adaptée au contexte africain.

Chapitre 1 : Contenu Offensif et Bambara en ligne

1. Introduction

Les plateformes de médias sociaux offrent aux utilisateurs un espace pour exprimer leurs opinions, partager des messages et interagir, mais elles peuvent aussi diffuser des contenus offensants, rendant l'environnement numérique toxique. La détection de ce type de contenu reste compliquée, en particulier dans les langues peu dotées comme le bambara [1].

Dans ce chapitre, nous allons explorer les différentes formes de contenu offensif, puis nous aborderons ensuite leur propagation à travers les plateformes numériques et leurs effets psychologiques sur les utilisateurs. L'attention sera portée sur la langue bambara, afin de comprendre les défis spécifiques qu'elle pose dans la détection automatique de contenu offensant.

2. Définition et typologie du contenu offensif

2.1. Contenu offensif

Le langage offensif désigne tout propos injurieux ou toute attaque, explicite ou implicite, dirigée contre autrui ou un groupe de personnes. Il inclut également les formes de langage indécent telles que les menaces (verbales ou non verbales), l'intimidation, les incitations à la violence, les insultes à visée dénigrante, ainsi que les termes exprimant le mépris [2][3].

Ce type de langage peut engendrer un climat d'hostilité et porter atteinte à l'estime de soi des personnes visées [4]. Il est fréquemment présent dans les discussions en ligne et peut influencer négativement les interactions sur les plateformes en lignes. On trouve différentes formes du contenu offensif telles que le langage agressif, langage vulgaire et langage discriminatoire.

2.1.1. Langage agressif

Le langage agressif se caractérise par l'utilisation de propos hostiles ou menaçants, dont l'objectif est de blesser ou d'intimider une personne ou un groupe, notamment par des appels directs ou indirects à la violence. Il se distingue par l'intention claire de nuire, souvent par des menaces ou des attaques personnelles [5].

Exemple : Tu vas le regretter, espèce d'imbéc***.

2.1.2. Langage vulgaire

Le langage vulgaire est un message offensant qui contient des grossièretés, comme des références ou des mentions de parties intimes ou d'actes sexuels [2]. Il peut également comporter des blasphèmes qui peuvent ou non se référer à un individu ou à un groupe de personnes [5]. Il est également essentiel de noter que le langage vulgaire, même s'il peut être

brutal et choquant, n'a pas forcément l'intention de blesser directement, mais peut simplement dénoter un manque de politesse ou de respect.

Exemple : Ferme ta gueule, on s'en fout de ta vie.

2.1.3. Langage discriminatoire

Cela peut inclure la stigmatisation d'un individu ou d'un groupe d'individus en raison de sa race, de son orientation sexuelle, de sa foi ou de tout autre élément lié à son identité. Il peut entraîner des préjudices émotionnels, psychologiques et corporels chez les individus qui en subissent les conséquences [3].

Exemple : Retourne dans ton pays, ici ce n'est pas pour les gens comme toi.

2.2. Discours haineux et incitation à la haine

Les discours haineux ou hatespeech sont des propos offensants qui ciblent un groupe en fonction de caractéristiques communes, telles que : la race, l'origine ethnique, le groupe ou le parti politique et la religion [2]. Le discours haineux se définit également comme un langage dirigé contre un groupe spécifique, avec l'intention de nuire aux individus ou de provoquer une perturbation sociale [6]. Cela peut également être interprété comme une tentative de discours haineux de déclencher une réaction violente ou de perturber certains groupes spécifiques. Il a la capacité de déclencher des conflits à une plus grande échelle, en encourageant l'intolérance ou la discrimination.

On peut distinguer plusieurs catégories dans le discours de haine :

• **Discours raciste :** désigne des propos ou comportements qui discriminent une personne ou un collectif en fonction de sa race ou de son origine ethnique [7].

• **Discours de nature sexiste :** se réfère aux propos ou comportements qui discriminent un individu ou un groupe de personnes en raison de leur sexe, leur identité de genre ou leur orientation sexuelle [7].

• **Discours xénophobe :** cela concerne les propos ou actes qui discriminent une personne ou un groupe, en raison de leur nationalité, culture ou religion [7].

• **Discours antireligieux :** il fait référence à des discours qui visent à critiquer, parfois de manière assez sévère, la croyance d'autrui, une croyance manifestement différente de celle exprimée par l'orateur [8].

2.3. Cyberintimidation ou intimidation en ligne

L'intimidation en ligne se définit comme une forme de comportement hostile sur internet impliquant un processus continu, tel qu'une succession de mots ou d'expressions blessants relayés par un harceleur dans le but d'infliger du préjudice à la personne ciblée [9].

On peut la trouver dans les médias sociaux, les applications de messagerie, les sites de jeux vidéo en ligne ou même à travers les téléphones portables. Ce genre de comportement se distingue par sa répétition et a pour objectif d'éveiller chez la personne ciblée des émotions telles que la peur, la colère ou l'humiliation [10].

La cyberintimidation peut se manifester de différentes façons, comme le dénigrement, l'isolement, les injures, les rumeurs ou encore les menaces. Ces actes peuvent être effectuées directement ou indirectement contre une personne. Il arrive parfois que l'individu harcelé ne sache pas qui est l'auteur des actes perpétrés. Voici des exemples de cyberintimidation [11], [12]:

- Il s'agit d'envoyer des e-mails ou des messages textuels offensants ou intimidants, ou de publier ce type de commentaires sur la page des médias sociaux d'un individu.
- Diffuser des rumeurs, des informations confidentielles et gênantes concernant une personne sur les plateformes des réseaux sociaux, par courrier électronique ou via des SMS ;
- Capturer une image d'un individu ou enregistrer une vidéo compromettante à l'aide d'une caméra numérique, puis la partager avec des tiers ou l'afficher sur Internet sans son consentement, ni sans qu'il en ait connaissance.
- Utiliser le mot de passe d'une autre personne pour se connecter à son compte sur les réseaux sociaux et y poster des contenus gênants ou scandaleux ;
- Diffuser des informations personnelles (numéro de téléphone, adresse) dans le but que des individus lui portent atteinte ou, tout du moins, qu'elle perde son sentiment de sécurité ;
- Élaborer des sondages en ligne et attribuer une note négative et offensante aux individus.

2.4. Violence médiatique et comportementale

Avant de voir l'influence du contenu offensif sur les plateformes il est essentiel de préciser deux termes : la violence médiatique et le comportement violent.

On définit la violence médiatique comme visuelle. C’est la représentation d'actes d'agressivité physique d'un individu envers un autre à travers des écrans. Cette définition de la

violence médiatique n'inclut pas les empoisonnements hors écran qui pourraient être sous-entendus, mais elle fait plutôt référence aux actes d'agression physique illustrés visuellement d'une personne à l'autre. On met l'accent sur l'aspect physique, manifeste et direct de l'agression [13].

Tandis que le comportement violent se définit comme un comportement agressif visant à porter atteinte ou à déranger une autre personne, qu'il soit de nature physique ou non. Ces actions physiques peuvent englober des actions telles que les bagarres, les agressions ou d'autres types d'attaques directes. Il est aussi important de noter que les comportements violents et agressifs se manifestent sous diverses formes et à des niveaux d'intensité variés, dont certains ne correspondent pas nécessairement à la définition conventionnelle de la violence comme insulter ou propager des rumeurs nuisibles [13].

3. Réseaux sociaux et propagation du contenu offensif

3.1. Influence des plateformes

Les plateformes de médias sociaux se sont imposées comme des lieux essentiels, utilisés tant par l'audience générale que par les figures publiques, pour partager leurs opinions sur des sujets d'actualités. Leur expansion remarquable a entraîné une création constante de contenu produit par les utilisateurs [14].

La croissance de ces dernières a complètement transformé la façon dont les personnes communiquent à l'échelle mondiale. Ces plateformes proposent une communication instantanée, personnelle et parfois anonyme. Bien qu'elles favorisent la diffusion facile de l'information et l'expression libre, elles ont aussi donné lieu à des problèmes tels que le harcèlement, les injures ou la propagation de fausses informations.

Des plateformes telles que Facebook, Twitter ou Instagram sont fréquemment utilisées pour propager des discours offensants ou de haine. L'absence d'une régulation appropriée, liée à l'anonymat des utilisateurs, incite la diffusion de contenus offensants. Ce phénomène engendre une préoccupation particulière chez les jeunes, qui sont souvent plus exposés et vulnérables.

Bien que ces plateformes aient mis en place des mécanismes de surveillance, l'identification automatique de ces discours demeure un défi. Les technologies actuelles ne parviennent pas toujours à identifier en temps réel toutes les formes de contenu inapproprié, mettant en évidence la nécessité de développer des outils plus performants [15].

Parallèlement, face à la montée des comportements antisociaux tels que le harcèlement ou les propos abusifs, la recherche et l'industrie s'intéressent de plus en plus au développement de systèmes capables d'identifier automatiquement les contenus problématiques. Ces initiatives s'inscrivent dans une volonté commune de préserver un environnement numérique sain et respectueux pour l'ensemble des utilisateurs [14].

3.2. Effets du contenu offensif sur la santé mentale

L'augmentation des commentaires offensants ou dégradants participe à l'établissement d'une ambiance hostile sur les plateformes digitales. Ce type d'environnement, généralement alimenté par des discours violents ou dégradants, peut susciter chez les victimes un sentiment d'exclusion, de peur ou d'isolement. Il arrive parfois que des utilisateurs délaissent ces espaces, voire les quittent totalement, ce qui nuit à l'objectif initial de ces plateformes (encourager le dialogue et la communication). Les conséquences psychologiques de ces interactions néfastes peuvent être particulièrement graves chez les jeunes, qui sont les plus exposés [15].

Les utilisateurs ordinaires peuvent être victimes de commentaires haineux ou d'attaques personnelles sur les réseaux sociaux. Dans les cas les plus graves, ces situations ont mené à des drames, comme le rapportent certains cas de suicide liés au cyberharcèlement [14]. Voici quelques effets négatifs sur la santé mentale :

- Anxiété et stress
- Dépression et isolement social
- Baisse de l'estime de soi
- Pensées suicidaires et le suicide
- Impact sur la santé mentale et les relations

3.3. Normalisation du langage offensif en ligne

Face à la montée en croissance des utilisateurs sur les plateformes sociales et au temps qu'ils y passent, l'emploi de discours haineux et d'un vocabulaire violent est devenu courant. Ces attitudes sont progressivement devenues « normales » dans une multitude d'interactions sur internet. Cette banalisation a été favorisée par l'anonymat, l'absence de répercussions directes et le défaut de modération systématique.

Le phénomène de cyberintimidation, autrefois sous-estimé et considéré comme un sujet peu sérieux, en raison du faible nombre d'utilisateurs et des réponses peu approfondies

proposées, telles que le simple fait de ne pas tenir compte des messages offensants. Actuellement, les circonstances sont tout autre.

Malgré de multiples actions mises en œuvre, même des grandes plateformes comme Facebook et Twitter (ou X) semblent avoir du mal à maîtriser ce phénomène. Dans certaines situations, les déclarations faites sur internet peuvent désormais entraîner des répercussions judiciaires, mettant en évidence la sévérité de cette dérive et la nécessité d'une sensibilisation collective. Il est donc essentiel de former les utilisateurs à une utilisation responsable du langage sur internet et d'améliorer les dispositifs de détection afin de maintenir un environnement respectueux [15].

Bien que les plateformes des médias sociaux permettent une plus grande liberté d'expression et une certaine forme d'anonymat, ces bénéfices sont parfois exploités pour véhiculer des discours haineux ou offensants, ce qui participe à la banalisation du langage agressif [14].

4. Cas du bambara et les défis du traitement du langage

4.1. Travaux antérieurs sur la détection de contenu offensif en bambara

La détection automatique de discours offensants est un domaine en pleine expansion, notamment dans les langues disposant de nombreuses ressources telles que l'anglais ou l'arabe. En revanche, pour les langues africaines comme le bambara, les recherches sont encore limitées.

Un travail notable sur ce sujet a été réalisé par Diallo Abdoul Karim en 2023. Il a créé une base de données en bambara en collectant des commentaires sur Facebook, où ces derniers ont été annotées manuellement pour identifier les contenus offensants. À partir de cette base, plusieurs algorithmes de classification et des modèles de deep learning ont été testés pour identifier les textes offensants des textes normaux [3], même si cela reste un défi. Diallo note que plusieurs obstacles linguistiques rendent ce travail difficile, comme la diversité des dialectes, l'absence d'une écriture standardisée et le manque de ressources numériques. Bien que le bambara soit largement parlé, sa transcription écrite varie beaucoup, ce qui complique encore le traitement automatique des textes.

4.2. Comparaison avec d'autres langues et dialectes moins ressourcés

Le bambara, à l'instar de nombreuses autres langues africaines telles que l'arabe dialectal, le haoussa, ainsi que des langues comme l'hindi en Inde, l'urdu au Pakistan, le bengali ou encore certaines langues dravidiennes, fait face à des défis uniques en matière de détection de contenu

offensif. Ces langues font face à des problèmes, tels que la diversité des accents, l'écriture informelle et le manque de données annotées [16], [17], [18], [19].

Cependant, l'arabe et l'hindi bénéficie de projets collaboratifs qui ont permis de constituer des bases de données plus étendues et des modèles plus performants, et certains efforts notables et jeux de données émergent, contribuant progressivement à l'avancement de ce domaine, contrairement au bambara et d'autres langues moins ressourcées qui restent sous-représentées. L'haoussa, même s'il est plus répandu, a aussi des défis, comme des expressions idiomatiques ambiguës, des insultes qui varient selon le contexte, l'influence du français dans les conversations [20]. Tout ça montre qu'il est important de créer des outils qui tiennent compte des réalités linguistiques en Afrique pour mieux détecter les contenus offensants.

Adam et ses collègues [20] montrent que pour le haoussa, il faut combiner des connaissances locales avec des modèles multilingues pour mieux détecter le contenu offensif, surtout quand il y a peu de ressources disponibles. De plus, AfriHate (Muhammad et ses collègues) [21] souligne que cette détection doit être adaptée culturellement, ce qui signifie qu'il est essentiel de faire appel à des locuteurs natifs et d'ajuster les modèles multilingues en conséquence.

4.3. Défis du traitement automatique du bambara

Il y a plusieurs difficultés avec le traitement du bambara, surtout à cause des caractéristiques de la langue et du manque de numérique. Même si beaucoup de gens parlent le bambara en Afrique de l'Ouest, il n'est pas assez présent dans les outils de traitement des textes. Comme l'a souligné Abdoul Karim Diallo [3], la diversité des dialectes est un vrai problème. Les différentes façons dont les gens s'expriment et écrivent en bambara rendent la création de modèles fiables assez difficile. Sur les réseaux sociaux les utilisateurs écrivent souvent de manière phonétique ou selon leur propre style, donc le fait qu'il n'y a pas de norme d'orthographe complique encore plus la gestion des données.

Majumder et ses collègues ajoutent, comme le note Diallo, que l'écriture en bambara est souvent informelle, ce qui complique les systèmes de traitement. En plus, Tapo et ses collègues, également mentionnés par Diallo, remarquent que le mélange des dialectes affecte la fiabilité des modèles.

De plus, les problèmes d'ambiguïté des mots, d'influence du français, et d'une grammaire qui change rendent la création de modèles performants encore plus compliquée. Tout cela

montre qu'il est important de développer des outils qui s'adaptent spécialement au bambara pour le traitement automatique.

5. Détection automatique du contenu offensif

5.1. Méthodes utilisées pour les langues ressourcées

Parmi les plus de 7000 langues parlées dans le monde, l'anglais occupe une position dominante en tant que langue standard internationale. Cette prédominance se reflète fortement dans le domaine de la recherche en traitement automatique du langage naturel (TALN), où la majorité des études et des applications sont axées sur l'anglais [22].

Cela s'explique par plusieurs facteurs : l'abondance de données disponibles sur les plateformes numériques, la richesse des outils de prétraitement linguistique, et la disponibilité de jeux de données annotés en libre accès. Des plateformes comme Twitter (actuellement X), Reddit ou Facebook servent de sources majeures pour la collecte de données textuelles, facilitant la création de corpus spécialisés pour la détection de discours haineux, offensants ou discriminatoires [22]. De nombreuses méthodes ont été développées pour détecter du contenu en anglais, notamment :

- Des modèles de classification supervisée utilisant des algorithmes ;
- Des réseaux de neurones profonds ;
- L'utilisation de vecteurs d'embedding comme Word2Vec, GloVe, ou des embeddings spécifiques à des domaines particuliers ;
- Des modèles de type transformers, tels que BERT et ses variantes, entraînés ou adaptés à la détection de propos offensants ;
- Des corpus multi-domaines, permettant d'élargir la couverture sémantique des modèles;
- Et des protocoles d'évaluation standardisés visant à tester la robustesse et la sensibilité des systèmes de détection [22].

5.2. Approches basées sur les algorithmes à base d'apprentissage

Aujourd'hui, pour détecter le contenu offensant, on utilise principalement des techniques d'apprentissage automatique et d'apprentissage profond. Ces méthodes entraînent des modèles qui, en analysant de grandes quantités de texte, apprennent à repérer les signes d'un langage offensif, devenant ainsi de plus en plus experts pour identifier ce type de textes au fil du temps. Mountaga Diallo et al. [23] dans leur étude destinée à l'analyse des sentiments, ont présenté

deux versions de données en bambara (V1 et V2), et des modèles tels que SVM et LSTM ont été testés. Les résultats ont montré une meilleure performance du SVM par rapport aux modèles profonds. Les auteurs ont alors rendu leur corpus public et prévoient d'en élargir la taille. Dans le travail de Diallo Abdoul Karim [3], il s'est focalisé sur la détection de contenus offensifs en bambara. Sept modèles d'apprentissage ont été comparés (dont SVM, NB, FastText, CNN, BiLSTM), testés sur des corpus équilibrés et déséquilibrés. Les résultats ont mis en évidence la performance du modèle BiLSTM, qui a obtenu un score F1 de 0.94 pour deux classes (Normal, offensif), et 0.88 pour trois classes. Bien que les modèles classiques aient obtenu de bons résultats, les performances des modèles profonds, en particulier le BiLSTM, démontrent leur capacité supérieure à modéliser la complexité linguistique du bambara, et ce malgré la faible quantité de données disponibles.

6. Conclusion

Ce chapitre nous a permis de comprendre la complexité et la diversité du contenu offensif en ligne. Nous avons vu comment il se manifeste sous plusieurs formes, impacte négativement les interactions sociales, et menace la santé mentale des utilisateurs, en particulier sur les réseaux sociaux. Nous avons aussi constaté que la langue bambara, bien que moins ressourcés, souffre d'un manque de ressources, les variations dans l'écriture et la diversité des dialectes qui rendent sa modélisation automatique difficile, contrairement à l'anglais qui est une langue ressourcée et d'autres langues peu dotées qui bénéficient des projets collaboratifs. Cela montre qu'il est important de développer des outils adaptés aux langues locales pour mieux détecter ces contenus.

Chapitre 2 : Apprentissage Automatique et Traitement du Langage Naturel (TALN)

1. Introduction

Avec la montée des interactions numériques, les technologies de traitement automatique du langage naturel (TALN) sont devenues incontournables pour comprendre, analyser et modérer les contenus en ligne [24]. Ce chapitre a donc pour objectif de présenter les fondements théoriques de l'apprentissage automatique, les principaux algorithmes utilisés en TALN, ainsi que les approches modernes telles que les modèles de deep learning et les transformers. Un accent particulier sera mis sur l'adaptation de ces techniques aux langues peu ressourcées, comme le bambara. Cela nous permettra d'identifier les méthodes les plus pertinentes à mettre en œuvre dans notre projet de détection automatique du contenu offensif en bambara.

2. Apprentissage automatique

L'apprentissage automatique (ou *machine learning* en anglais) est un sous domaine clé de l'intelligence artificielle. Il se concentre sur la conception des algorithmes permettant à une machine d'apprendre de façon autonome à partir de données et d'expériences passées. C'est une méthode qui offre aux systèmes informatiques la possibilité d'apprendre à partir des données, d'améliorer de façon continue leur efficacité à travers l'expérience, et de faire des prédictions sans avoir été spécifiquement programmés pour chaque tâche [25].

L'apprentissage automatique peut être divisé en trois catégories principales, notamment l'apprentissage supervisé, non supervisé et par renforcement. L'objectif est que ces systèmes soient capables d'en tirer des connaissances ou des résultats utiles. Ces différentes méthodes d'apprentissage constituent les fondations du développement futur de l'intelligence artificielle, qui, avec le temps, jouera un rôle de plus en plus important dans l'assistance aux tâches quotidiennes des êtres humains (16).

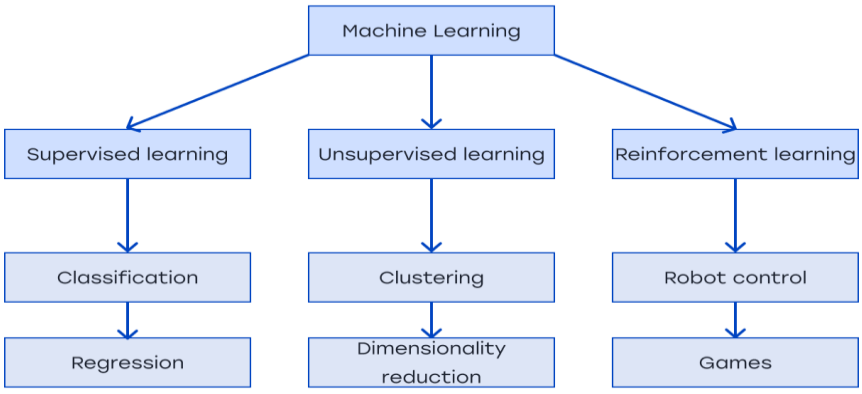

Figure 1. *Différentes catégories de machine learning [27].*

2.1. Apprentissage supervisé

L'apprentissage supervisé consiste généralement à entraîner un modèle doté d'une fonction qui associe une entrée à une sortie, à partir d'exemples d'entrées et de sorties déjà connus[27]. Comme un élève guidé par un enseignant, l'algorithme apprend à associer des entrées (x) à des sorties (y) en s'appuyant sur un ensemble de données d'entraînement. Si les prédictions s'écartent des attentes, ces données servent à le réajuster. Méthode d'intelligence artificielle la plus répandue, elle permet de structurer l'apprentissage en fournissant les données, les résultats et le scénario [26]. Il y a deux catégories d'apprentissage supervisé telles que la classification et la régression.

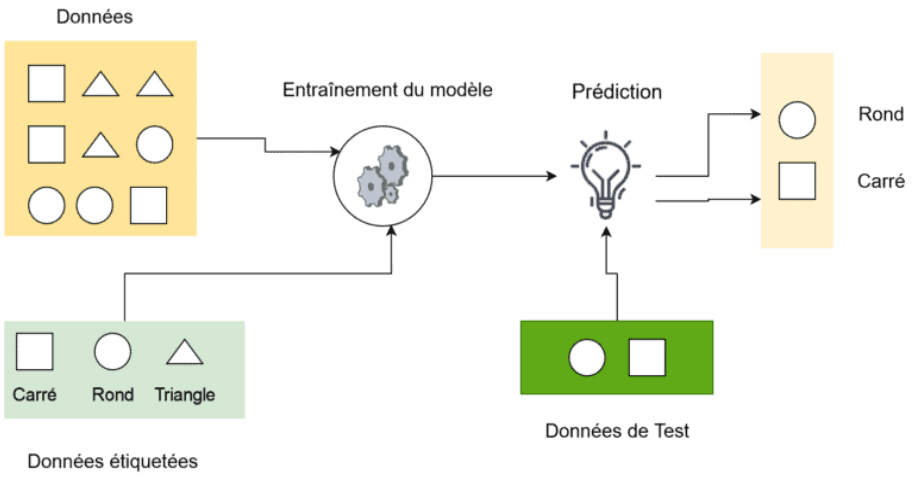

Figure 2. *Exemple d'apprentissage supervisé [25].*

2.1.1. Classification

La classification est un type d'apprentissage automatique supervisé dans lequel les algorithmes apprennent à partir de données pour prédire un résultat ou un événement futur [28]. Par exemple, elle sert à distinguer les courriels comme étant du spam ou non, ou encore à regrouper des personnes en fonction de critères tels que le genre, l'état matrimonial ou la tranche d'âge. Il y a une multitude d'algorithme d'apprentissage avec des principes de fonctionnement différents tels que les arbres de décision, l'algorithme du plus proche voisin, l'algorithme du forêt aléatoire, réseaux de neurones, SVM, etc.

2.1.2. Régression

La régression est utilisée lorsque la variable cible est continue et est de nature numérique. Elle permet d'estimer des valeurs, tels que le coût d'une maison en se basant sur ses caractéristiques (superficie, emplacement, nombre de pièces, etc.). De telles tâches ont fréquemment recours à des modèles statistiques comme la régression linéaire[26], [29].

Dans la figure 3, la ligne pointillée dans la classification représente une frontière droite qui sépare les deux classes. Tandis qu'en régression, elle représente la relation linéaire entre les deux variables[30].

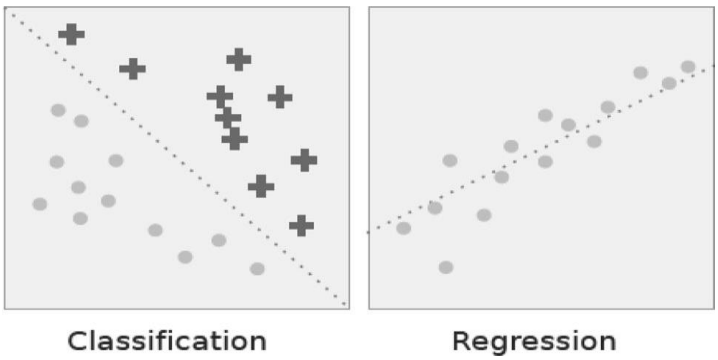

Figure 3. *Classification vs Regression [30].*

2.2. Apprentissage non supervisé

L'apprentissage non supervisé est une forme d'apprentissage automatique où les modèles sont entraînés avec un ensemble de données qui n'a pas été étiqueté et sont habilités à travailler sur ces informations sans supervision. L'apprentissage non supervisé analyse des ensembles de données non étiquetées, sans nécessiter d'intervention humaine [27], contrairement à l'apprentissage supervisé. L'algorithme explore des données brutes et identifie seulement des structures ou des regroupements sans indication préalable. Cette méthode, bien que moins répandue aujourd'hui, suscite un intérêt croissant pour son potentiel à permettre une plus grande autonomie des systèmes intelligents [26].

On utilise des algorithmes d'apprentissage non supervisé dans plusieurs applications comme regrouper des données, réduire les dimensions, analyser les relations entre les variables, et même créer de nouvelles caractéristiques à partir de données.

Le clustering est une méthode qui regroupe des objets similaires dans un ensemble de données. Par exemple, elle peut être utilisée pour segmenter des clients selon leurs habitudes d'achat. Tandis que la réduction de la dimensionnalité est une méthode visant à diminuer le nombre de variables d'un jeu de données tout en conservant les informations essentielles, ce qui permet également d'atténuer le bruit. Un exemple courant est la compression d'images sans perte notable de contenu informatif [29].

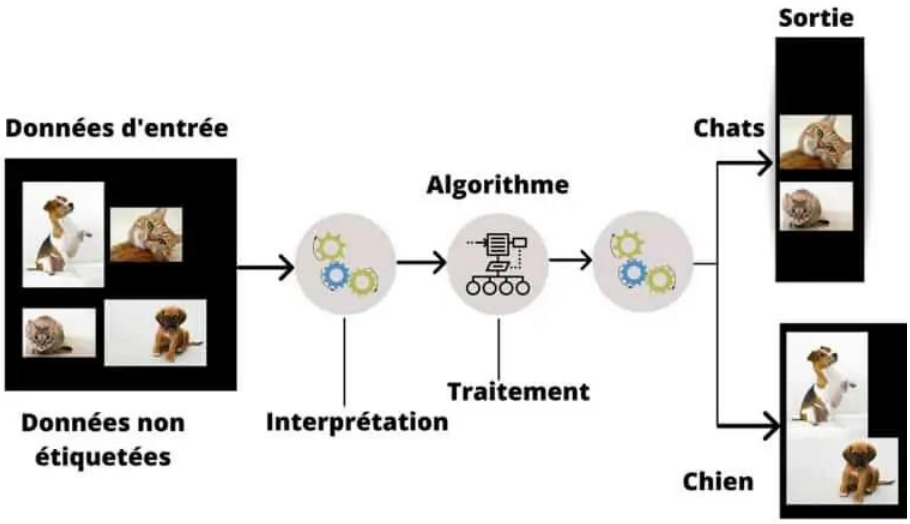

Figure 4. *Apprentissage non supervisé [25].*

2.3. Apprentissage par renforcement

L'apprentissage par renforcement est une forme d'apprentissage automatique où un agent apprend en interagissant avec son environnement. À l'inverse des autres approches, celle-ci ne s'appuie pas sur des données préalablement annotées, mais elle opère grâce à un mécanisme de récompenses et de sanctions qui guide l'agent vers des décisions plus pertinentes [26]. Il s'inspire des comportements d'apprentissage que l'on observe chez les humains et les animaux. L'agent essaie différentes actions, voit ce qui se passe, et ajuste sa façon de faire pour obtenir plus de récompenses. Grâce aux progrès récents, on peut maintenant créer des systèmes qui fonctionnent bien dans des environnements compliqués et changeants [26]. On cite quelques exemples d'application, comme le contrôle du robot où il apprend afin qu'il exécute des actions spécifiques dans son environnement pour réaliser des buts prédéterminés. On trouve également le cas des jeux, où la formation d'un agent pour jouer à des jeux, tels que les échecs ou les jeux vidéo, avec l'objectif d'améliorer son score [29].

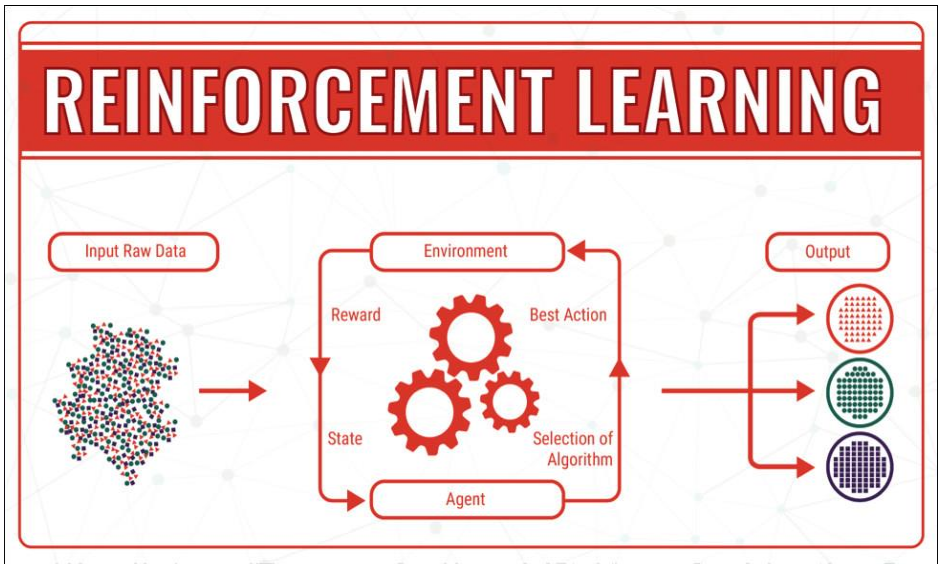

Figure 5. *Apprentissage par renforcement [26].*

3. Algorithmes d'apprentissage automatique appliqués au TALN

3.1. Modèles classiques

Les modèles classiques sont les premières méthodes utilisées en apprentissage automatique pour le traitement du langage. Ils restent efficaces pour des tâches comme la classification des textes, avec des algorithmes simples à mettre en place et rapides à exécuter.

3.1.1. Machines à vecteurs de support (SVM)

C'est un algorithme de classification supervisée qui élabore un hyperplan optimal en se basant sur les données d'apprentissage, ce qui facilite la distinction des catégories tout en classifiant les données inédites [31].

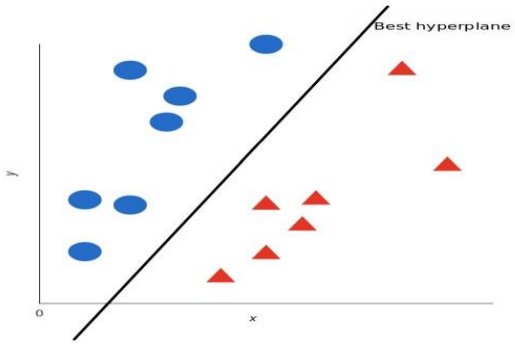

Figure 6. *Classification de texte en utilisant SVM [32].*

Tout d'abord l'algorithme commence par transformer les données d'origine pour les projeter dans un espace avec plus de dimensions. Cela facilite la séparation des données. Par la suite, le SVM recherche une limite linéaire (nommée hyperplan) qui divise les données en deux classes distinctes. Le SVM détermine cette limite en se basant sur quelques points clés du jeu de données, connus sous le nom de vecteurs de support. Ces points servent également à déterminer la marge, qui est la distance séparant la frontière des points les plus proches de chaque côté [27]. Les SVMs sont aussi des méthodes utilisées pour classer des données, qu'elles soient simples (linéaires) ou complexes (non linéaires) [27].

3.1.2. Régression Logistique

La régression logistique utilise une fonction mathématique spéciale (appelée *fonction logistique*) pour estimer les probabilités comme indiqué dans l'équation (1). Elle fonctionne bien lorsque les données peuvent être séparées clairement par une ligne droite. En revanche, si le jeu de données contient trop de variables, elle peut s'adapter excessivement aux données, ce qu'on appelle le surapprentissage [30].

$$g(z) = \frac{1}{1+exp(-z)} \quad \textbf{(1)}$$

3.1.3. Naive Bayes

Le classificateur Naive Bayes multinomial se repose sur le calcul des probabilités conditionnelles des différentes classes à partir des caractéristiques extraites de documents. Une fois entraîné sur un corpus annoté, il est capable de prédire la catégorie d'un nouveau texte en s'appuyant sur ces probabilités. Bien qu'il soit fondé sur l'hypothèse parfois restrictive d'indépendance entre les variables, cet algorithme demeure apprécié pour sa simplicité, sa rapidité d'exécution, et ses résultats généralement fiables dans de nombreux contextes [32].

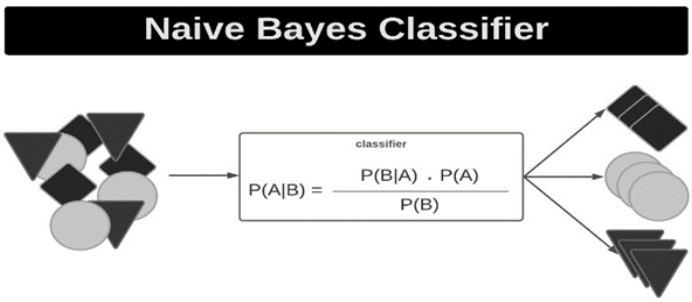

Figure 7. *Classificateur Naives Bayes [32].*

On peut aussi dire que c'est un algorithme de classification probabiliste basé sur le théorème de Bayes, supposant que chaque variable est indépendante des autres [31].

3.1.4. Descente de gradient stochastique (SGD)

La descente de gradient stochastique (SGD) est une méthode utilisée pour optimiser une fonction objective de manière répétée. Le terme "stochastique" fait référence à l'aspect aléatoire du processus. Cette approche permet de réduire le temps de calcul, surtout lorsqu'on travaille avec des données très complexes ou de grande dimension. En contrepartie, elle peut être un peu plus lente à atteindre une solution optimale. Le gradient correspond à la pente d'une fonction et mesure comment une variable change en réponse à une autre [30].

3.1.5. Arbres de décision

C'est une méthode de classification qui crée un arbre de règles : chaque nœud correspond à une condition, chaque branche à un résultat, et chaque feuille à une classe [31]. Les modèles d'arbre de décision, qui se comparent à un arbre de probabilité, divisent continuellement les données dans le but de les catégoriser ou de réaliser des prévisions basées sur les réponses des questions successives. Le modèle examine les données et fournit des réponses aux questions pour vous accompagner dans la prise de décisions plus avisées [28].

3.1.6. Random Forest

Le Random Forest est aussi basé sur des arbres mais c'est une méthode d'ensemble composée de multiples arbres de décision. La classification se fait par vote majoritaire entre les prédictions des arbres [31].

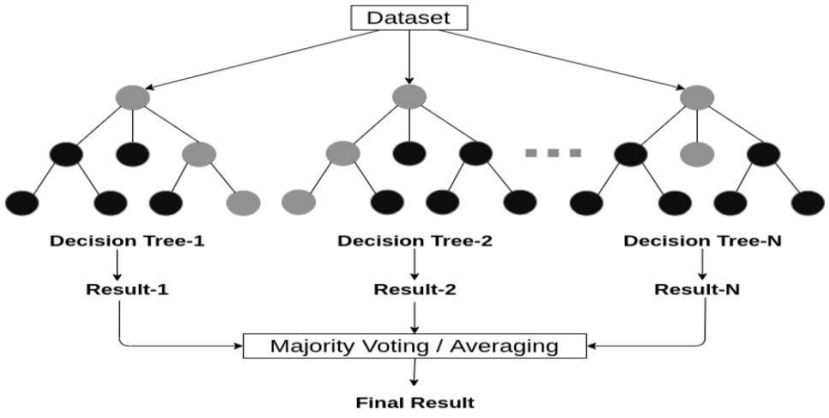

Figure 8. *Classificateur Random Forest avec des arbres de décision [30].*

3.1.7. K-plus proches voisins (K-NN)

C'un algorithme simple qui classifie les nouvelles données en fonction de leur similarité avec les exemples connus [31]. La méthode des K-voisins est une méthode statistique qui examine la proximité d'une donnée par rapport à une autre afin de déterminer si un groupement de ces deux données est envisageable ou pas. Le niveau de similarité entre les points de données est représenté par leur proximité [28].

3.2. Modèles d'apprentissage profond

Le deep learning ou apprentissage profond, est une branche de l'intelligence artificielle qui repose sur des réseaux de neurones artificiels à plusieurs couches. Il permet à une machine d'apprendre automatiquement à partir de données brutes, en extrayant des caractéristiques pertinentes sans intervention humaine directe. Cette méthode est l'une des plus performantes pour le traitement de grandes quantités de données complexes [33].

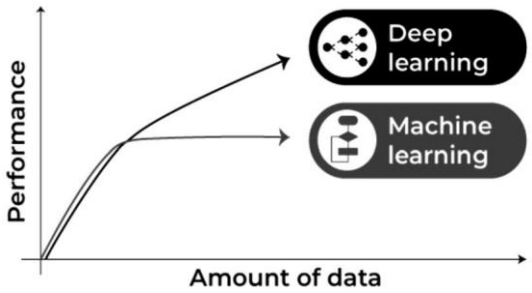

Figure 9. *Performance entre le machine learning et le deep learning en terme de données [30].*

Un réseau de neurones artificiels est un modèle de calcul dont l'architecture s'inspire, de manière simplifiée, du fonctionnement des neurones biologiques. Ces réseaux sont généralement optimisés à l'aide de méthodes d'apprentissage probabilistes, notamment bayésiennes [34]. Ils sont constitués de neurones interconnectés qui coopèrent pour détecter des motifs, apprendre à partir de données et formuler des prédictions. À l'image des neurones biologiques, chaque neurone artificiel reçoit des informations de neurones voisins, les traite, puis transmet le résultat à d'autres neurones, formant ainsi un réseau dynamique d'apprentissage [35].

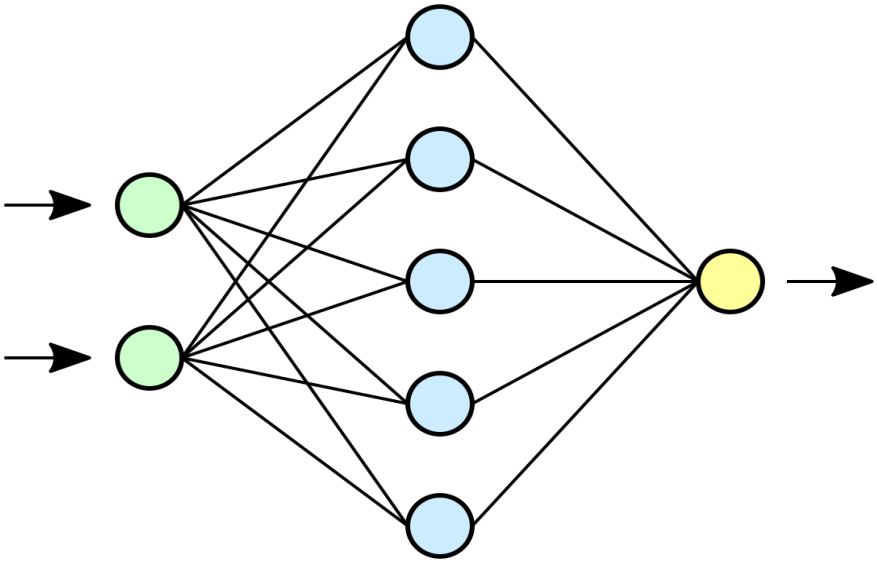

Figure 10. *Réseau de neurones artificiels [34].*

L'architecture d'un réseau de neurones artificiels se compose de trois types de couches :

Couche d'entrée (input layer) : elle reçoit les données brutes à traiter, comme les pixels d'une image ou les mots d'une phrase.

Couches cachées (hidden layers) : elles transforment les informations en identifiant des motifs complexes et en éliminant le superflu. Lorsque plusieurs couches sont empilées, on parle de réseau profond.

Couche de sortie (output layer) : elle génère le résultat final, que ce soit une prédiction de classe ou une valeur numérique [35].

3.2.1. Réseaux de neurones convolutifs (CNN)

Un réseau de neurones convolutif est un type de réseau de neurones qui est fait pour traiter des données organisées en grille, comme des images ou des séquences. Il est composé de plusieurs couches, comme les couches de convolution et de sous-échantillonnage, ainsi que des couches entièrement connectées. Les couches de convolution se chargent d'identifier automatiquement les caractéristiques clés des données d'entrée. Les couches de sous-échantillonnage, elles, réduisent la taille des données extraites. Cela aide à simplifier le modèle et à éviter le surajustement tout en le rendant plus stable, même avec du bruit ou des erreurs dans les données [36].

3.2.2. Réseaux de neurones récurrents (RNN)

Les réseaux de neurones récurrents (RNN) sont un type de réseau profond qui a des connexions en boucle. Cela leur permet de traiter des données en séquence en se souvenant d'informations précédentes [36].

Dans le monde de l'apprentissage profond, les réseaux de neurones récurrents (RNN) sont largement utilisés, surtout pour le traitement du langage et de la voix. Contrairement aux réseaux de neurones classiques, les RNN sont conçus pour travailler avec des données qui viennent dans un certain ordre. Cela leur permet de capter le contexte important dans ces données. Cette capacité à gérer le contexte est super utile pour plein d'applications, comme comprendre le sens des mots dans une phrase, selon le contexte global [37].

3.2.3. Long Short-Term Memory (LSTM)

Le LSTM est une version des RNN qui aide à gérer les relations sur le long terme dans les données. À la différence des RNN normaux, il a un système de mémoire appelé cellule de mémoire. Cela permet de garder des informations importantes plus longtemps sans perdre ce qu'on a appris sur le long terme.

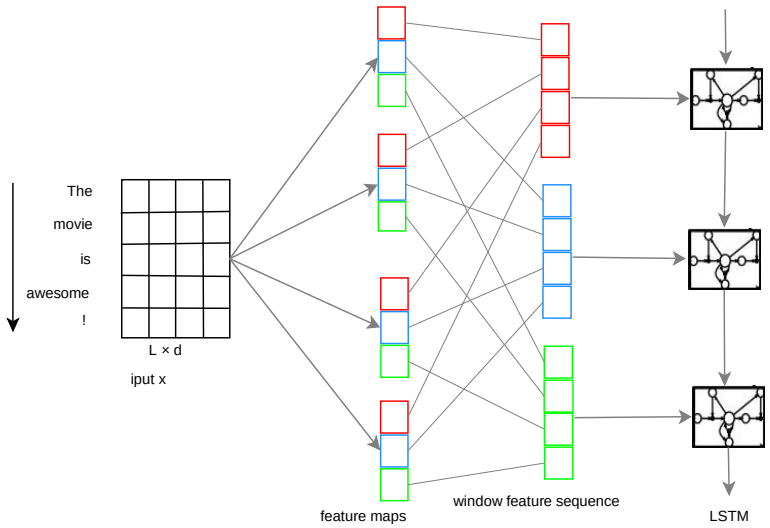

Figure 11. *Architecture du modèle LSTM pour la modélisation de phrases [38].*

Les blocs de la même couleur dans la carte des caractéristiques et dans la séquence des fenêtres de caractéristiques montrent que l'information vient de la même fenêtre. Les lignes pointillées montrent comment les caractéristiques d'une fenêtre se relient à leur source sur la

carte. Le résultat final du modèle provient de la dernière unité cachée mise en place par le LSTM [38]. L'unité récurrente fermée (GRU) a été conçue comme une variante simplifiée de LSTM et plus efficace en termes de puissance de calcul par rapport au LTSM et le RNN classique.

3.3. Techniques modernes des Transformers

Les transformers ont révolutionné le domaine du traitement automatique du langage naturel grâce à leur capacité à capturer les relations contextuelles dans les textes. Parmi ces modèles, on distingue :

3.3.1. BERT

BERT, grâce à son approche d'entraînement bidirectionnelle, excelle dans la capture du sens contextuel du langage. Cela le rend bien adapté pour des tâches où il faut vraiment saisir le sens du texte. Par contre, cela nécessite beaucoup de ressources, ce qui signifie que ça peut prendre plus de temps pour s'entraîner et que ça utilise beaucoup de mémoire [39].

3.3.2. GPT

GPT, quant à lui, a une méthode unidirectionnelle, ce qui l'aide à générer du texte qui a du sens. Cela peut affecter ses performances sur des tâches comme l'analyse des sentiments, où le contexte est important. Malgré cela, GPT a souvent besoin de moins de temps pour s'entraîner et produire des résultats et des modèles comme BERT [39].

3.3.3. T5

T5 (ou Text-to-Text Transfer Transformer) se distingue par son approche texte à texte. Il traite l'analyse des sentiments en transformant les textes d'entrée en résultats qui montrent les prédictions de sentiments. Ce changement de méthode permet à T5 de bien gérer diverses tâches de traitement du langage, même si cela nécessite aussi plus de ressources pendant l'entraînement et la production de résultats [39].

3.4. Représentation des données textuelles

Avant d'entraîner un modèle, il est nécessaire de transformer les textes en données numériques. Pour cela, on utilise différentes techniques de représentation textuelle, qu'il s'agisse d'approches traditionnelles basées sur la fréquence des mots, ou d'approches avancées qui prennent en compte le sens et le contexte des termes.

3.4.1. Approches traditionnelles

Les approches traditionnelles de la représentation des données textuelles sont basées sur le modèle de Bag of words (BOW) ou TF-IDF.

3.4.1.1. Bag of words

Le modèle de sac de mots (Bag of Words ou BoW) est une méthode courante pour transformer des mots en traits qu'on peut utiliser. On l'utilise souvent pour classer des textes, où chaque mot est vu comme un trait à part entière. D'une manière générale, la présence ou la fréquence des mots aide à entraîner un classificateur [32].

3.4.1.2. TF-IDF

La méthode TF-IDF (abbréviation de *term frequency inverse document frequency*), qui veut dire *Fréquence de Terme - Fréquence Inverse de Document*, est un calcul qui aide à savoir combien un mot est important dans un document par rapport à un ensemble de corpus. La fréquence des termes (TF) montre combien de fois un mot apparaît dans un document : plus il est fréquent, plus sa valeur TF est grande. D'un autre côté, la fréquence inverse des documents d'un corpus (IDF) mesure la rareté du mot dans tous les documents : si un mot est courant dans plusieurs documents, sa valeur IDF sera basse. Donc, le score TF-IDF est plus élevé pour les mots qui sont fréquents dans un document, mais peu présents dans le reste des documents, ce qui en fait un bon moyen de déterminer leur pertinence [32].

3.4.2. Approches avancées

Contrairement aux méthodes statistiques qui sont basées sur les fréquences des mots, il existe des méthodes plus avancées pour mieux représenter les documents et leurs caractéristiques telles que Word2Vec, Doc2Vec, GloVe et FastText.

Word2Vec: C'est une méthode d'apprentissage, où les mots sont transformés en vecteurs, ce qui est utile pour les modèles qui font des prédictions [31].

Doc2Vec: C'est une version de Word2Vec qui crée des vecteurs pour des documents complets au lieu de se concentrer uniquement sur des mots [31].

GloVe: C'est un algorithme d'apprentissage non supervisé qui sert à créer des représentations vectorielles des mots. Il s'entraîne en regardant comment souvent les mots apparaissent ensemble dans un ensemble de textes. Les résultats montrent des liens intéressants entre les mots dans cet espace vectoriel [40].

FastText: C'est une méthode qui mélange plusieurs techniques de traitement des langues, comme le sac des mots, les n-grammes (qui sont des groupes de *n* mots qui aident à repérer des

motifs dans le texte), et des sous-mots pour enrichir les modèles. Chaque mot est transformé en un petit vecteur, et tout le texte est construit en additionnant ces vecteurs. Cette méthode aide à mieux généraliser et à partager les informations entre différentes catégories. Des recherches ont montré que FastText fonctionne presque aussi bien que les modèles plus complexes, tout en étant beaucoup plus rapide à entraîner et à utiliser [41].

3.5. Applications du TALN dans la détection de contenu offensif

3.5.1. Modération du contenu en ligne

La modération du contenu, c'est tout ce qui permet d'analyser, filtrer et de gérer ce qui est posté en ligne. L'idée est de rendre l'internet plus sûr et respectueux en repérant les contenus inappropriés comme la violence, la haine, les fausses informations, ou les discours de discrimination.

Le traitement automatique du langage étant devenu un pilier important dans ce domaine. Ces techniques permettent de détecter et d'analyser les contenus offensants, en utilisant des outils de traitement de texte, d'analyse de sentiment et des systèmes de modération de contenu basés sur l'intelligence artificielle tels que d'apprentissage automatique pour comprendre le sens et le contexte des textes [42].

3.5.2. Analyse de sentiment et classification des textes

L'analyse de sentiments (ou opinion mining) consiste à extraire et interpréter les émotions exprimées dans des sources textuelles dématérialisées sur de grandes quantités de données [43]. L'analyse de sentiments est aussi définie comme le traitement et la détection informatique des opinions, émotions, attitudes ou sentiments positifs, négatifs ou neutres présents dans tout texte, ainsi que de la subjectivité du texte [44].

La classification des sentiments de textes se divisent en trois approches de classifications telles que l'apprentissage automatique utilisant des algorithmes pour détecter les sentiments à partir des données. Il y a également l'approche lexicale qui s'appuie sur un lexique de mots porteurs de sentiments. Elle se divise en deux méthodes. Il y a une méthode qui utilise un dictionnaire pour trouver des synonymes et antonymes des mots d'opinion. Une autre méthode, basée sur des textes, explore un grand nombre de textes pour repérer des mots liés aux émotions dans un contexte particulier. Enfin, l'approche hybride qui combine ces deux techniques, d'où elle est populaire vu qu'elle tire le meilleur des deux mondes [44].

3.6. Adaptation des techniques de TALN au bambara

L'adaptation des techniques de TALN au bambara s'appuie sur l'entrainement des modèles mutltilingues, ou sur les techniques de transfert d'apprentissage. Cette dernière est un ensemble de méthodes de machine learning où un modèle conçu pour une tâche spécifique est réutilisé en tant que point de départ pour un modèle sur une autre tâche. Le concept fondamental du transfert de l'apprentissage est que les connaissances obtenues d'un domaine peuvent être transférées et mises en œuvre dans un autre domaine, améliorant ainsi la performance et l'efficience de l'apprentissage [45]. Voici les méthodes principales d'apprentissage par transfert :

Ajustement (Fine-Tuning) : cette technique implique de perfectionner un modèle préalablement formé sur un vaste ensemble de données afin qu'il convienne à une nouvelle tâche.

• **Extraction de caractéristiques (Feature Extraction) :** les caractéristiques acquises pendant la phase d'entraînement initiale constituent le fondement du nouveau modèle.

• **Adaptation de domaine (Domain Adaptation) :** cette méthode vise à ajuster les données du domaine source afin qu'elles soient similaires à celles du domaine cible [45]. Avec le bambara, par exemple, l'apprentissage par transfert permet de s'appuyer sur des modèles pré-entraînés sur d'autres langues pour compenser le manque de données. Cela va améliorer l'analyse linguistique, ainsi que les performances des modèles sur cette langue moins ressourcées.

4. Conclusion

Ce chapitre a permis d'introduire les concepts fondamentaux de l'apprentissage automatique et du traitement du langage naturel, ainsi que les principales méthodes utilisées dans ce domaine. Nous avons détaillé les algorithmes classiques et les approches de deep learning, en insistant sur les techniques de représentation textuelle. Ces connaissances sont essentielles pour concevoir des modèles capables d'analyser automatiquement les textes en bambara. À la fin, on a montré que certaines techniques, comme le transfert d'apprentissage, peuvent aider à adapter ces modèles aux langues moins ressourcés comme le bambara. Le prochain chapitre mettra en œuvre ces concepts à travers l'expérimentation et l'évaluation de différents modèles de classification pour la détection de contenu offensif.

Chapitre 3 : Implémentation et Expérimentations

1. Introduction

Après avoir exploré les concepts théoriques de l'apprentissage automatique et les défis du traitement automatique dans les langues moins ressourcées, ce chapitre présente la mise en œuvre pratique de la détection de contenu offensif en bambara. Nous y décrivons les différentes étapes du processus expérimental, que ce soit la préparation des données jusqu'à l'évaluation des modèles d'apprentissage. Plusieurs corpus ont été construits afin de tester les modèles dans des contextes variés : corpus équilibré, déséquilibré, binaire ou à trois classes. Ce chapitre expose également la configuration des algorithmes classiques et profonds, ainsi que les métriques utilisées pour mesurer leurs performances. L'objectif est d'identifier les approches les plus efficaces pour la classification automatique de textes en langue bambara.

2. Prétraitement des données

2.1. Base de données

La base de données utilisée dans ce travail est fournie dans le cadre de ce projet par mon encadrant. Il s'agit d'une base en langue bambara préexistante, enrichie et augmentée par des collaborateurs dans un cadre académique antérieur. Elle contient un ensemble varié de textes environ 14.000 issus de sources réelles (commentaires, discussions, publications en ligne), annotés manuellement selon trois catégories :

- **0** : texte normal qui représente les propos non offensants.

  Exemple : basitɛ qui veut dire **il n'y a pas de soucis.**

- **1** : texte abusif qui contient des propos particulièrement graves, insultants ou dégradants.

  Exemple : fougariden bilalen qui veut dire **espèces d'incapables que tu es**.

- **2** : texte offensif qui désigne des propos blessants, provocateurs ou dénigrants, mais d'intensité moins faible que les textes abusifs.

  Exemple : wuya kan fola yi qui veut dire **menteur.**

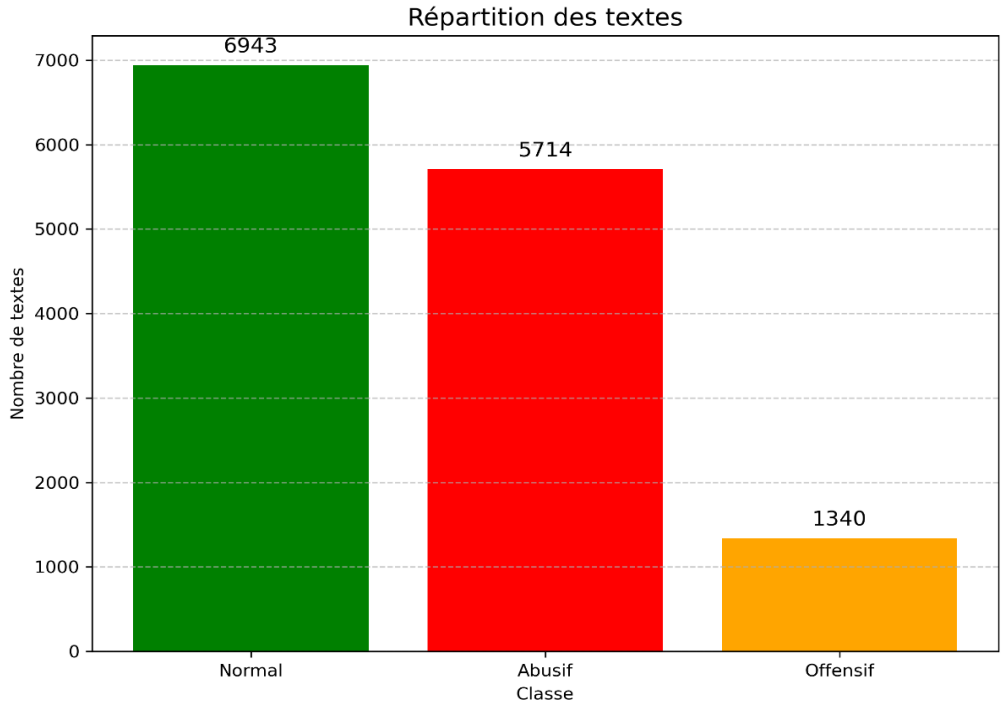

Figure 12. *Nombre de textes par classe.*

2.2. Construction des corpus

Dans le but d'évaluer l'impact du déséquilibre des classes sur les l'entrainement des modèles, nous avons réorganisé la base de données selon plusieurs configurations, en nous inspirant partiellement de la démarche adoptée par Abdoul Karim Diallo en 2023, nous avons généré trois sous-corpus distincts :

- Un corpus déséquilibré à trois classes, reflétant la distribution naturelle des données, c'est la base de données originale. Ce corpus contient les textes classés en normal, abusif et offensif. Il est déséquilibré car certaines classes, comme la classe offensif, qui sont beaucoup moins représentées que les autres. Cela permet d'évaluer les modèles dans une configuration réaliste, proche de celle que l'on retrouve dans la vie réelle.
- Un corpus binaire déséquilibré, construit en regroupant tous les textes offensants (abusif et offensif) dans une seule classe appelée « offensif », opposée aux textes normaux. Ce regroupement simplifie la tâche de classification tout en conservant le déséquilibre naturel entre les classes, ce qui met en évidence la capacité des modèles à détecter les textes offensants, même s'ils sont minoritaires.
- Un corpus binaire équilibré, dans lequel les textes offensants et les textes normaux sont présents en proportions égales. Ce corpus permet de comparer les performances des

modèles dans un contexte neutre, sans influence de déséquilibre, et de mieux observer leur aptitude réelle à distinguer les deux catégories.

2.3. Nettoyage des données

Pour alléger la base de données afin d'éviter de le surcharger avec des données qui ne vont pas nous servir, et améliorer ainsi la qualité des modèles on va procéder comme suit :

- ✓ Supprimer les URL
- ✓ Supprimer les emojis
- ✓ Supprimer les mentions et les hashtags
- ✓ Supprimer la ponctuation
- ✓ Changer les chiffres en textes
- ✓ Mettre en minuscule
- ✓ Supprimer les espaces redondants

2.4. Tokenisation et vectorisation

Dans le NLP, il est important de transformer les textes bruts en une forme compréhensible par les algorithmes d'apprentissage automatique. On distingue deux étapes importantes et nécessaires telle que la tokenisation et la vectorisation.

2.4.1. Tokenisation

La tokenisation est le processus qui consiste à découper un texte en unités plus petites appelées tokens ou jetons. Ces jetons peuvent être des mots, des caractères ou des sous-mots, etc. Cette étape permet de structurer le texte afin qu'il soit plus facilement exploitable par les modèles. Grâce à cette opération, on peut ensuite appliquer différentes méthodes de traitement ou d'analyse sur chaque jeton individuellement [46].

Exemple :

Texte original : « Ce mémoire est intéressant »

Après tokenisation : « Ce », « mémoire », « est », « intéressant »

2.4.2. Vectorisation

La vectorisation permet de transformer chaque mot ou phrase en vecteurs de nombres. Une fois que les textes sont divisés en jetons, il faut les convertir en valeurs numériques. En effet, les algorithmes de machine learning ne travaillent qu'avec des données numériques. La vectorisation ne sert pas uniquement à convertir le texte : elle peut aussi préserver la

signification des mots, leur importance dans le texte, ou leur relation avec d'autres mots. Cela permet aux modèles d'apprentissage d'apprendre plus efficacement [46].

3. Conception et entraînement des modèles

3.1. Phase d'entraînement et de test

L'entraînement et le test des modèles sont des étapes essentielles pour évaluer leur capacité à détecter automatiquement le contenu offensif. Pour cela, la base de données est divisée en trois parties :

- **Ensemble d'entraînement (80%)** : utilisé pour apprendre aux modèles à reconnaître les différentes catégories de textes.
- **Ensemble de validation (10%)** : utilisé pour ajuster les paramètres du modèle et éviter le surapprentissage.
- **Ensemble de test (10%)** : utilisé à la fin pour évaluer la performance réelle du modèle sur des données jamais vues auparavant.

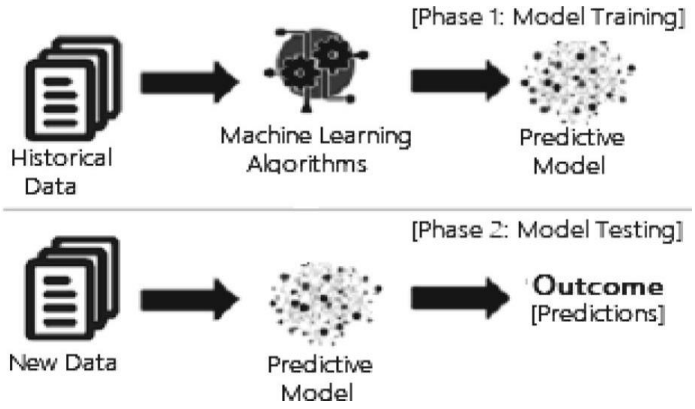

Figure 13. *Les phases d'entrainement et de test [30].*

3.2. Configuration des modèles

Les hyperparamètres peuvent avoir un impact significatif sur la performance du modèle. Par exemple, en utilisant des valeurs inappropriées, il est alors possible de surapprendre (overfitting) ou de sous-apprendre (underfitting) sur les données d'entraînement [47].

Dans ce travail, on a testé plusieurs algorithmes de machine et deep learning avec différents hyperparamètres pour comparer leur efficacité dans la détection automatique de contenu offensif en bambara.

A. Modèles classiques

Les modèles de machine learning explorés incluent SVM, Naïve Bayes, Régression Logistique, SGD et Random Forest. Pour chacun, plusieurs valeurs d'hyperparamètres ont été testées afin d'optimiser les performances. Par exemple, pour SVM et la régression logistique, le paramètre **C** ([0.1, 0.5, 1.0, 1.02, 1.05, 1.2, 2.0, 10.0]) a été ajusté pour contrôler la régularisation. Le SVM a été évalué avec deux types de noyaux (*linéaire* et *RBF*) pour s'adapter à la nature des données.

En outre, Naïve Bayes a utilisé différents niveaux de lissage via le paramètre alpha ([0.1, 0.5, 1.0, 1.02, 1.05, 1.2, 2.0, 10.0]). Le modèle SGD a été configuré avec différentes fonctions de perte (*hinge* et *log*) et pénalités (*L2*, *elasticnet*), tandis que Random Forest a été évalué avec plusieurs nombres d'arbres (n_estimators = [10, 20, 30, 40, 40, 50, 60, 70, 80, 90, 100]). Ces réglages ont permis d'identifier les combinaisons optimales pour chaque corpus, en fonction du niveau d'équilibre entre les classes.

B. Modèles profonds

Les modèles de deep learning testés dans ce travail incluent CNN, BiLSTM, GRU, FastText et une architecture hybride BiLSTM+CNN. Pour chacun d'eux, nous avons utilisé une couche d'embedding pour représenter les mots, suivie de couches spécifiques au modèle : convolution pour le CNN, unités récurrentes pour BiLSTM et GRU, et pooling global pour FastText.

Les paramètres courants tels que la fonction d'activation (*ReLU* ou *Sigmoid*), l'optimiseur (*Adam*), la fonction de perte (*binary* ou *categorical crossentropy*), la taille du lot (batch size) et le nombre d'époques (epochs) ont été fixés selon les configurations standards dans la littérature. Ces choix ont permis d'assurer une convergence stable et des résultats comparables entre les modèles, tout en tenant compte des contraintes de calcul.

4. Évaluation et analyse des résultats

4.1. Evaluation du classificateur

Le classificateur est utilisé pour prédire la catégorie de textes (langage abusif, offensant ou neutre) à partir de l'ensemble de test. Sa performance est mesurée via la matrice de confusion, basée sur quatre indicateurs : TP (True Positive), TN (True Negative), FP (False Positive) et FN (False Negative).

	Predicted No	Predicted Yes
Actual No	TN	FP
Actual Yes	FN	TP

Figure 14. *Matrice de confusion [31].*

Dans ce qui suit, nous décrivons quelques métriques courantes utilisées dans la classification des textes permettant d'évaluer les performances.

4.1.1. Precision

La précision est également connue sous le nom de valeur prédite positive. Il s'agit de la proportion de prédictions positives qui sont effectivement positives.

$$Precision = \frac{TP}{(TP+FP)} \quad \textbf{(2)}$$

4.1.2. Recall (Rappel)

Il s'agit de la proportion des résultats positifs réels par rapport aux résultats positifs prévus.

$$Recall = \frac{TP}{TP + FN} \quad \textbf{(3)}$$

4.1.3. F-Measure

Il s'agit de la moyenne harmonique de la précision et du rappel. La mesure F-measure ou F-score (appelée également *F1*) accorde la même importance à la précision et au recall.

$$F - measure(F1) = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \quad \textbf{(4)}$$

4.1.4. Accuracy

Il s'agit du nombre d'instances correctement classées (vrais positifs et vrais négatifs).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad \textbf{(5)}$$

4.2. Résultats des métriques et discussions

On a utilisé les deux techniques de vectorisation à savoir : TFIDF et BoW, pour obtenir les meilleurs résultats possibles.

Le modèle SVM a montré des performances globalement robustes, d'où sur le corpus équilibré à deux classes, le meilleur résultat a été obtenu avec un noyau linéaire et C=2.0, en utilisant TF-IDF, atteignant une accuracy de 87,5% avec des scores bien équilibrés entre les classes. Sur le corpus déséquilibré à deux classes, le noyau linéaire avec la même valeur de C a donné de très bonnes performances également (accuracy de 89%), et sur le corpus à trois classes déséquilibrées, le noyau RBF avec C=10.0 a permis d'atteindre une accuracy de 81,3%,

mais avec une chute importante du rappel sur les classes minoritaires. Le recours à la représentation TF-IDF a globalement donné de meilleures performances que BoW, en particulier sur les corpus complexes ou déséquilibrés, grâce à une pondération plus fine des termes. Toutefois, même avec des réglages optimaux, SVM reste sensible à la distribution des classes.

Tableau III-1. *Performances de SVM sur les Corpus.*

Corpus	Précision	Rappel	F1-score	Accuracy
Équilibré 2 classes				**0.875**
Normal	0.887	0.879	0.883	
Offensif	0.862	0.872	0.867	
Avg	*0.876*	*0.875*	*0.875*	
Déséquilibré 2 classes				**0.890**
Normal	0.898	0.886	0.892	
Offensif	0.882	0.895	0.888	
Avg	*0.890*	*0.890*	*0.890*	
Déséquilibré 3 classes				**0.813**
Normal	0.797	0.941	0.863	
Abusif	0.869	0.770	0.816	
Offensif	0.567	0.274	0.370	
Avg	*0.806*	*0.813*	*0.801*	

La régression logistique a obtenu de très bonnes performances, notamment sur le corpus déséquilibré binaire (accuracy jusqu'à 88,9%) et équilibré (87,5%), particulièrement avec TF-IDF. Sur le corpus déséquilibré à trois classes, une meilleure performance a été atteinte avec BoW (accuracy de 82,6%), surtout pour des valeurs de C entre 2.0 et 10.0. Cela montre que BoW peut parfois mieux capter les structures simples dans des contextes de déséquilibre fort. Globalement, le modèle reste robuste avec les deux représentations, mais TF-IDF se démarque par une meilleure précision globale sur les classes majoritaires.

Naïve Bayes a présenté une bonne adaptabilité sur l'ensemble des corpus. En particulier, sur le corpus équilibré, BoW a légèrement surpassé TF-IDF (accuracy de 88,6% contre 88,1%), suggérant que pour ce modèle probabiliste, une représentation simple mais fréquente comme BoW peut être plus efficace. Sur les corpus déséquilibrés, TF-IDF donne de meilleurs résultats (accuracy de 89,4% en binaire), mais reste limité sur les classes rares (comme dans le corpus à trois classes, avec un rappel très faible pour la classe minoritaire). Le paramètre alpha a

également montré une influence significative : des valeurs modérées (0.5 à 1.0) permettent de bien gérer la variabilité tout en conservant une certaine sensibilité aux mots rares.

Tableau III-2. *Performances de la régression logistique sur les Corpus.*

Corpus	Précision	Rappel	F1-score	Accuracy
Équilibré 2 classes				**0.875**
Normal	0.889	0.876	0.883	
Offensif	0.860	0.875	0.867	
Avg	*0.876*	*0.875*	*0.876*	
Déséquilibré 2 classes				**0.889**
Normal	0.896	0.886	0.891	
Offensif	0.881	0.892	0.886	
Avg	*0.889*	*0.889*	*0.889*	
Déséquilibré 3 classes				**0.826**
Normal	0.876	0.879	0.878	
Abusif	0.806	0.883	0.843	
Offensif	0.485	0.266	0.344	
Avg	*0.801*	*0.809*	*0.799*	

Tableau III-3. *Performances de Naives Bayes sur les Corpus.*

Corpus	Précision	Rappel	F1-score	Accuracy
Équilibré 2 classes				**0.886**
Normal	0.886	0.902	0.894	
Offensif	0.885	0.867	0.876	
Avg	*0.886*	*0.886*	*0.885*	
Déséquilibré 2 classes				**0.894**
Normal	0.882	0.915	0.898	
Offensif	0.907	0.871	0.889	
Avg	*0.894*	*0.894*	*0.893*	
Déséquilibré 3 classes				**0.816**
Normal	0.801	0.945	0.867	
Abusif	0.869	0.776	0.820	
Offensif	0.579	0.266	0.365	
Avg	*0.809*	*0.816*	*0.803*	

Le classifieur linéaire avec descente de gradient stochastique (SGD) s'est avéré très performant, notamment sur les corpus déséquilibrés, binaire (accuracy jusqu'à 89%), 3 classes (82,6%) et équilibré (87,2%). La combinaison TF-IDF avec une faible régularisation (alpha

bas) et la fonction de perte *hinge* a donné les meilleurs résultats. L'utilisation de BoW a été globalement moins efficace, surtout pour la détection des classes minoritaires. Cela s'explique par la sensibilité de SGD aux représentations fortement redondantes, que BoW a tendance à produire. TF-IDF, en normalisant les poids des termes fréquents, rend le modèle plus stable et discriminant.

Tableau III- 4. *Performances de SGD sur les Corpus*

Corpus	Précision	Rappel	F1-score	Accuracy
Équilibré 2 classes				**0.872**
Normal	0.886	0.872	0.879	
Offensif	0.856	0.872	0.864	
Avg	*0.872*	*0.872*	*0.872*	
Déséquilibré 2 classes				**0.890**
Normal	0.895	0.890	0.892	
Offensif	0.885	0.890	0.888	
Avg	*0.890*	*0.890*	*0.890*	
Déséquilibré 3 classes				**0.826**
Normal	0.873	0.879	0.876	
Abusif	0.807	0.882	0.843	
Offensif	0.493	0.266	0.346	
Avg	*0.813*	*0.826*	*0.816*	

Tableau III-5. *Performances de random forest sur les Corpus.*

Corpus	Précision	Rappel	F1-score	Accuracy
Équilibré 2 classes				**0.822**
Normal	0.776	0.947	0.853	
Offensif	0.919	0.686	0.786	
Avg	*0.843*	*0.826*	*0.822*	
Déséquilibré 2 classes				**0.816**
Normal	0.759	0.940	0.840	
Offensif	0.916	0.687	0.785	
Avg	*0.836*	*0.816*	*0.813*	
Déséquilibré 3 classes				**0.786**
Normal	0.745	0.958	0.838	
Abusif	0.871	0.705	0.779	
Offensif	0.724	0.169	0.275	
Avg	*0.794*	*0.786*	*0.764*	

Random Forest affiche de bonnes performances sur les trois types de corpus, avec une accuracy maximale de 82,2% sur le corpus équilibré. TF-IDF a permis d'obtenir des résultats stables avec un bon compromis précision/rappel, particulièrement pour les classes majoritaires. Cependant, sur le corpus déséquilibré à trois classes, même avec un grand nombre d'arbres, la classe minoritaire reste peu détectée (rappel < 0.2). BoW n'a pas montré de gain significatif par rapport à TF-IDF. Cela confirme que, pour ce modèle basé sur des arbres de décision, la représentation TF-IDF est généralement plus adaptée, notamment grâce à une meilleure gestion des valeurs rares et extrêmes.

Tableau III-6. *Performances de CNN sur les Corpus.*

Corpus	Précision	Rappel	F1-score	Accuracy
Équilibré 2 classes				**0.864**
Normal	0.885	0.857	0.871	
Offensif	0.842	0.872	0.856	
Avg	*0.865*	*0.864*	*0.864*	
Déséquilibré 2 classes				**0.878**
Normal	0.889	0.870	0.879	
Offensif	0.867	0.886	0.876	
Avg	*0.878*	*0.878*	*0.878*	
Déséquilibré 3 classes				**0.826**
Normal	0.880	0.879	0.879	
Abusif	0.845	0.855	0.850	
Offensif	0.415	0.395	0.405	
Avg	*0.825*	*0.826*	*0.825*	

Le modèle CNN donne de très bons résultats sur les trois types de corpus. Sur le corpus équilibré, l'accuracy atteint 86,4% avec un bon équilibre entre précision et rappel. Sur le corpus déséquilibré binaire, les performances restent élevées (accuracy 87,8%). Même sur le corpus déséquilibré à trois classes, le CNN reste compétitif (accuracy 82,6%), avec un rappel de 0.395 pour la classe minoritaire, ce qui montre une certaine capacité à gérer les classes rares. Cela s'explique par sa capacité à extraire des motifs locaux efficaces via les couches convolutives.

Le modèle BiLSTM affiche des performances solides sur les trois corpus. Il atteint une accuracy de 86,2% sur le corpus équilibré, et jusqu'à 88,4% sur le corpus déséquilibré binaire. Sur le corpus à trois classes, il obtient 81,6%, avec un rappel de 0.371 pour la classe minoritaire. La modélisation bidirectionnelle permet au modèle de mieux capter le contexte des séquences, ce qui se traduit par une bonne généralisation sur les données.

Tableau III-7. *Performances de BiLSTM sur les Corpus.*

Corpus	Précision	Rappel	F1-score	Accuracy
Équilibré 2 classes				**0.862**
Normal	0.882	0.856	0.869	
Offensif	0.840	0.869	0.854	
Avg	*0.862*	*0.862*	*0.862*	
Déséquilibré 2 classes				**0.884**
Normal	0.870	0.908	0.889	
Offensif	0.899	0.858	0.878	
Avg	*0.884*	*0.884*	*0.884*	
Déséquilibré 3 classes				**0.816**
Normal	0.879	0.863	0.871	
Abusif	0.804	0.855	0.829	
Offensif	0.455	0.371	0.409	
Avg	*0.811*	*0.816*	*0.813*	

Tableau III-8. *Performances de GRU sur les Corpus.*

Corpus	Précision	Rappel	F1-score	Accuracy
Équilibré 2 classes				**0.855**
Normal	0.854	0.877	0.866	
Offensif	0.855	0.828	0.841	
Avg	0.855	0.855	0.854	
Déséquilibré 2 classes				**0.869**
Normal	0.879	0.862	0.870	
Offensif	0.858	0.876	0.867	
Avg	0.869	0.869	0.869	
Déséquilibré 3 classes				**0.831**
Normal	0.897	0.875	0.886	
Abusif	0.806	0.889	0.845	
Offensif	0.464	0.315	0.375	
Avg	0.822	0.831	0.824	

Le GRU montre des résultats proches de ceux du BiLSTM, avec une accuracy de 85,5% sur le corpus équilibré, 86,9% sur le déséquilibré binaire, et 83,1% sur le corpus à trois classes. Le rappel de la classe minoritaire (0.315) reste correct compte tenu du déséquilibre. Ce modèle, plus léger que le BiLSTM, parvient à garder une bonne performance tout en étant plus rapide à entraîner.

Tableau III-9. *Performances de fast Text sur les Corpus.*

Corpus	Précision	Rappel	F1-score	Accuracy
Équilibré 2 classes				**0.870**
Normal	0.890	0.864	0.877	
Offensif	0.849	0.878	0.863	
Avg	0.871	0.870	0.871	
Déséquilibré 2 classes				**0.884**
Normal	0.879	0.898	0.888	
Offensif	0.891	0.870	0.880	
Avg	0.884	0.884	0.884	
Déséquilibré 3 classes				**0.804**
Normal	0.834	0.883	0.858	
Abusif	0.840	0.800	0.820	
Offensif	0.413	0.363	0.386	
Avg	0.86	0.86	0.86	

FastText offre de bonnes performances sur les corpus équilibré et déséquilibré binaire (accuracy jusqu'à 88,4% et 87%). Sur le corpus à trois classes, les résultats sont plus modestes (accuracy 80,4%) mais le rappel de la classe minoritaire atteint 0.363, ce qui est correct pour un modèle aussi simple. Sa capacité à intégrer les sous-mots et sa rapidité d'entraînement en font un modèle pratique et compétitif.

Tableau III-10. *Performances de BiLSTM+CNN sur les Corpus.*

Corpus	Précision	Rappel	F1-score	Accuracy
Équilibré 2 classes				**0.865**
Normal	0.883	0.861	0.872	
Offensif	0.845	0.869	0.857	
Avg	*0.865*	*0.865*	*0.865*	
Déséquilibré 2 classes				**0.879**
Normal	0.886	0.877	0.882	
Offensif	0.872	0.881	0.877	
Avg	*0.879*	*0.879*	*0.879*	
Déséquilibré 3 classes				**0.810**
Normal	0.892	0.834	0.862	
Abusif	0.793	0.875	0.832	
Offensif	0.420	0.379	0.398	
Avg	*0.810*	*0.810*	*0.809*	

Le modèle hybride BiLSTM+CNN combine les avantages des réseaux convolutifs et des réseaux récurrents. Il atteint une accuracy de 86,5% sur le corpus équilibré, 87,9% sur le déséquilibré binaire, et 81% sur le corpus à trois classes. Son rappel pour la classe minoritaire reste élevé (0.379), ce qui montre que la combinaison des deux architectures permet de mieux capter les structures complexes dans les données.

4.3. Comparaison des performances des modèles

Dans l'ensemble, les modèles de deep learning surpassent les modèles classiques de machine learning, surtout sur les corpus déséquilibrés. Les modèles CNN, BiLSTM, GRU et BiLSTM+CNN montrent une meilleure capacité à détecter la classe minoritaire, avec des rappels proches ou supérieurs à 0.35, là où les modèles classiques peinent à dépasser 0.25. Les modèles de machine learning restent cependant compétitifs sur le corpus équilibré, avec des performances comparables aux modèles profonds, notamment la régression logistique et Naïve Bayes. En revanche, sur les corpus avec déséquilibre, les architectures profondes, en particulier CNN et BiLSTM+CNN, sont plus stables et plus performantes.

4.4. Analyse des erreurs et biais potentiels

4.4.1. Analyse des erreurs selon les modèles

L'analyse des résultats met en évidence certaines tendances récurrentes dans les erreurs commises par les modèles. On constate que certains modèles sont plus résistants face aux déséquilibres de classes ou aux formulations ambiguës, tandis que d'autres ont tendance à se baser sur les classes les plus fréquentes.

Les modèles profonds comme montrent de très bonnes performances dans la détection des textes offensifs, même lorsque ceux-ci sont exprimés de manière complexe. Sur les corpus binaires, qu'ils soient équilibrés ou non, ces modèles atteignent souvent des scores supérieurs à 86 % en précision, rappel et F1-score. Leur efficacité s'explique par leur capacité à saisir des relations plus fines dans les textes, y compris lorsque le sens dépend de la structure ou du contexte local d'un mot ou d'une expression. À l'inverse, les modèles plus simples, basés sur des approches linéaires ou probabilistes, commettent davantage d'erreurs, surtout lorsqu'il s'agit d'identifier les textes appartenant à une classe peu représentée. Sur le corpus déséquilibré, par exemple, qui comporte trois classes, la majorité des erreurs concernent la confusion entre la classe minoritaire (souvent la classe 2) et la classe 0. Certains modèles vont même jusqu'à ignorer presque totalement la classe 2, ce qui affecte fortement leur capacité à bien la détecter et dégrade significativement leurs résultats sur cette partie.

Globalement, les modèles profonds se distinguent par leur stabilité et leur capacité à généraliser, alors que les modèles classiques montrent vite leurs limites face aux formulations ambiguës ou aux textes à faible contenu.

4.4.2. Analyse des erreurs selon les corpus

L'analyse des erreurs en fonction des corpus utilisés montre à quel point la structure et le contenu des données influencent la qualité des prédictions.

Sur le corpus déséquilibré à trois classes, les erreurs sont à la fois plus fréquentes et plus critiques. La classe 2, nettement minoritaire, est souvent confondue avec la classe 0. Même les modèles les plus performants rencontrent des difficultés à bien la détecter, avec des rappels qui tournent autour de 0.37 à 0.40. Ce type d'erreur s'explique par la proximité sémantique entre certaines classes, mais aussi par la présence d'expressions ambiguës, parfois interprétées différemment selon le contexte culturel ou social. Il arrive aussi que certaines insultes implicites ou déguisées soient considérées comme normales par les modèles, ce qui complique encore la tâche.

Sur le corpus binaise déséquilibré, les modèles obtiennent de meilleurs résultats. Les erreurs y sont principalement dues à une sous-estimation de la classe 1 (offensif), notamment pour les messages très courts, ironiques ou rédigés de manière peu conventionnelle. Malgré cela, les performances restent très satisfaisantes, avec une bonne précision et un rappel équilibré dans l'ensemble.

Enfin, le corpus binaire équilibré est celui qui produit les résultats les plus fiables. Les erreurs y sont rares, et les prédictions se rapprochent fortement des étiquettes réelles. Tous les modèles atteignent de très bons scores sur ce corpus, ce qui confirme que l'équilibre entre les classes et une annotation plus homogène facilitent grandement le travail des classifieurs.

4.5. Techniques d'augmentation des données

Pour traiter le déséquilibre du corpus déséquilibré à trois classes, nous avons utilisé SMOTE (Synthetic Minority Over-sampling Technique). Cette méthode génère de nouveaux exemples synthétiques pour la classe minoritaire afin d'équilibrer les effectifs dans l'ensemble d'entraînement. Nous n'avons pas appliqué SMOTE aux modèles de deep learning car ces modèles intègrent souvent déjà des techniques internes de régularisation et nécessitent un pipeline d'augmentation différent. De plus, SMOTE fonctionne mieux avec des vecteurs fixes, comme ceux produits par TF-IDF ou BoW avec les modèles classiques. Pour plus de clarté, les

résultats détaillés sur la classe minoritaire avec les résultats des meilleurs hyperparamètres obtenus précédemment sur ce corpus sont présentés dans un tableau comparatif.

Tableau III-11. *Résultats de la classe minoritaire avec SMOTE*

Modèle	Accuracy	Rappel	F1-score
SVM (linéaire)	0.793	0.371	0.390
SVM (RBF)	0.803	0.363	0.326
Régression Logistique	0.803	0.427	0.431
Naïve Bayes	0.799	0.556	0.462
SGD (hinge)	0.795	0.452	0.419
SGD (log)	0.794	0.476	0.423
Random Forest	0.784	0.250	0.354

Les modèles entraînés avec SMOTE ont permis d'améliorer légèrement la détection de la classe minoritaire sur le corpus à trois classes. Par exemple, pour la régression logistique avec C=10, le rappel de la classe minoritaire est passé de 0.266 à 0.427, soit un gain important. Naïve Bayes a également montré une amélioration avec un rappel atteignant 0.556 pour alpha=1.0. Le modèle SVM, en particulier avec un noyau RBF et C=10, a obtenu une accuracy de 80,3% avec un rappel plus élevé (0.363) pour la classe difficile. Bien que les performances globales restent proches de celles sans SMOTE, l'oversampling a réduit l'impact du déséquilibre entre les classes. Cependant, la détection des classes peu représentées reste encore limitée malgré SMOTE.

5. Conclusion

Ce chapitre a permis d'évaluer de manière expérimentale diverses approches pour la détection automatique de contenu offensif en bambara. Les résultats montrent que les modèles profonds, notamment BiLSTM et CNN, offrent de bonnes performances, même dans des contextes de déséquilibre. Toutefois, certains modèles classiques comme SVM ou Naïve Bayes s'est révélé particulièrement efficaces sur des corpus bien préparés, avec des vectorisations adaptées. La représentation TF-IDF s'est avérée plus performante que le BoW dans la majorité des cas. Malgré de bons résultats globaux, la détection des classes minoritaires reste un défi

important. Ces expérimentations confirment l'intérêt d'utiliser des modèles combinés ou d'appliquer des techniques de rééquilibrage comme SMOTE dans les travaux futurs.

Conclusion Générale

Ce mémoire a permis d'exploiter une problématique à la fois technique et sociale, notamment la détection automatique du contenu offensif dans une langue africaine peu dotée (bambara). Nous avons montré que, malgré les défis liés à la rareté des ressources numériques, à la diversité dialectale et à l'absence de normalisation, il est possible de développer des modèles de classification efficaces.

Les travaux menés ont mis en évidence l'importance de l'adaptation linguistique et culturelle dans le traitement automatique des langues locales. Les performances observées avec des modèles tels que BiLSTM, CNN, FastText, ainsi que des modèles classiques comme SVM et Naive Bayes, ont montré qu'il est possible d'obtenir de bons résultats même avec un corpus limité. Les modèles profonds, notamment BiLSTM et FastText, ont atteint les meilleures accuracy, tandis que CNN s'est distingué par sa stabilité sur les différents corpus. Les modèles classiques, quant à eux, se sont avérés efficaces sur les classes majoritaires, mais moins précis pour la classe minoritaire contrairement aux modèles profonds qui affiche une légère supériorité sur ce point. Ces résultats soulignent l'importance du choix des représentations textuelles, du calibrage des hyperparamètres, et de l'adaptation des modèles aux spécificités des langues peu dotées. De plus, les modèles de type Transformers, comme BERT, bien qu'exigeants en ressources, représentent une alternative prometteuse et peuvent être adaptés au bambara via des techniques de transfert d'apprentissage.

Au-delà des aspects techniques, ce travail souligne l'importance de ne pas exclure les langues locales dans les développements technologiques. Intégrer le bambara et d'autres langues africaines peu dotées dans les systèmes intelligents permettrait de rendre les technologies linguistiques plus accessibles, plus équitables, et plus représentatives des diversités culturelles. Il s'agit d'une avancée non seulement scientifique, mais aussi éthique. Ce travail ouvre plusieurs perspectives. Il serait pertinent, dans le futur, de :

- Constituer des corpus plus riches, diversifiés et représentatifs des usages réels du bambara ;
- Impliquer des locuteurs natifs ou linguistes pour affiner les annotations ;

- Explorer davantage les méthodes de transfert d'apprentissage et de BERT adaptées aux langues peu dotées ;
- Déployer ces modèles dans des systèmes de modération automatique dans les plateformes sociales pour contribuer à la lutte contre les discours offensifs en ligne.

Cependant, certaines limites doivent être reconnues. D'une part, la taille réduite du corpus et la variabilité des dialectes ont limité la capacité des modèles à bien généraliser. D'autre part, l'évaluation reste centrée sur des métriques classiques sans prise en compte des biais culturels ou contextuels. Il serait également intéressant d'intégrer des analyses qualitatives des erreurs faites par les modèles.

En conclusion ce mémoire s'inscrit dans une dynamique d'innovation inclusive et respectueuse des diversités linguistiques, en contribuant à une meilleure prise en compte des langues locales dans les outils technologiques modernes. Il constitue une étape parmi d'autres vers une plus grande justice linguistique dans les technologies numériques.

Références bibliographiques

[1] F. M. Adam, A. Y. Zandam, and I. Inuwa-Dutse, "Detection of offensive and threatening online content in a low resource language, 2023, doi: https://doi.org/10.48550/arXiv.2311.10541.

[2] H. Mubarak, A. Rashed, K. Darwish, Y. Samih, and A. Abdelali, "Arabic offensive language on Twitter: Analysis and experiments," in *Proceeding. Sixth Arabic Natural Language Processing Workshop (WANLP)*, Kyiv, Ukraine (Virtual), Apr. 2021, pp. 126-135. URL: https://aclanthology.org/2021.wanlp-1.13/.

[3] DIALLO, Abdoul Karim, "Détection automatique des contenus offensifs en Bambara sur les réseaux sociaux," mémoire de master, Université 8 Mai 1945 Guelma, Guelma, Algérie, 2023.

[4] Mnassri K, Farahbakhsh R, Chalehchaleh R, Rajapaksha P, Jafari AR, Li G, Crespi N. 2024. A survey on multilingual offensive language detection. PeerJ Comput. Sci. 10:e1934 DOI 10.7717/peerj-cs.1934.

[5] M. J. Díaz-Torres *et al.*, "Automatic Detection of Offensive Language in Social Media: Defining Linguistic Criteria to build a Mexican Spanish Dataset," in *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, Language Resources and Evaluation Conference (LREC 2020)*, Marseille, France, May 11-16, 2020, pp. 132-136.

[6] A. Ajvazi et C. Hardmeier, "A Dataset of Offensive Language in Kosovo Social Media," in *Proceedings of the 13th Conference on Language Resources and Evaluation (LREC)*, Marseille, France, Jun. 20–25, 2022, pp. 1860–1869.

[7] NASSIROU DAOUDA Aminou, "Catégorisation automatique du contenu offensif," mémoire de master, Université 8 Mai 1945 Guelma, Guelma, Algérie, 2022.

[8] Peeters, Christophe. Le discours de haine dans les médias : état du droit et sanctions applicables. Faculté de droit et de criminologie, Université catholique de Louvain, 2020. Prom. : Jongen, François. http:// hdl.handle.net/2078.1/thesis:24366.

[9] Chong, P. T., Othman, N. F., Abdullah, R. S., Anawar, S., Ayop, Z., & Ramli, S. N. (2021). Cyberbullying detection in Twitter using sentiment analysis. *International Journal of Computer Science and Network Security, 21*(11), 1–10.

[10] « Cyberharcèlement : qu'est-ce que c'est et comment y mettre fin ? » Consulté le: 12 avril 2025. [En ligne]. Disponible sur: https://www.unicef.org/fr/mettre-fin-violence/mettre-fin-intimidation-en-ligne.

[11] L. de l'Éducation, « Cyberbullying: signification, types et réglementations », L'Hebdomadaire de l'Éducation. Consulté le: 12 avril 2025. [En ligne]. Disponible sur: https://educ-hebdo.fr/787-cyberbullying-signification-types-et-reglementations/.

[12] « Cyberintimidation », Gouvernement du Québec. Consulté le: 22 avril 2025. [En ligne]. Disponible sur: https://www.quebec.ca/famille-et-soutien-aux-personnes/violences/intimidation/cyberintimidation.

[13] L. R. Huesmann et L. D. Taylor, « The Role of Media Violence in Violent Behavior », *Annu. Rev. Public Health*, vol. 27, nº 1, p. 393-415, avr. 2006, doi: 10.1146/annurev.publhealth.26.021304.144640.

[14] S. Modha, P. Majumder, and T. Mandl, "Filtering Aggression from the Multilingual Social Media Feed," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, Santa Fe, New Mexico, USA, August 2018, pp. 199-207.

[15] H. K. Sharma, T. P. Singh, K. Kshitiz, H. Singh, et P. Kukreja, « Detecting Hate Speech and Insults on Social Commentary using NLP and Machine Learning », vol. 4, nº 12.

[16] M. M. Khan, K. Shahzad, et M. K. Malik, « Hate Speech Detection in Roman Urdu », *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 20, nº 1, p. 1-19, janv. 2021, doi: 10.1145/3414524.

[17] F. A. Jafri, K. Rauniyar, S. Thapa, M. A. Siddiqui, M. Khushi, et U. Naseem, « CHUNAV: Analyzing Hindi Hate Speech and Targeted Groups in Indian Election Discourse », *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, p. 3665245, mai 2024, doi: 10.1145/3665245.

[18] K. Sreelakshmi, B. Premjith, B. R. Chakravarthi, et K. P. Soman, « Detection of Hate Speech and Offensive Language CodeMix Text in Dravidian Languages Using Cost-Sensitive Learning Approach », *IEEE Access*, vol. 12, p. 20064-20090, 2024, doi: 10.1109/ACCESS.2024.3358811.

[19] M. Das, S. Banerjee, P. Saha, et A. Mukherjee, « Hate Speech and Offensive Language Detection in Bengali », 7 octobre 2022, *arXiv*: arXiv:2210.03479. doi: 10.48550/arXiv.2210.03479.

[20] F. M. Adam, A. Y. Zandam, et I. Inuwa-Dutse, « Detection and Analysis of Offensive Online Content in Hausa Language », 26 avril 2024, *In Review*. doi: 10.21203/rs.3.rs-4266465/v2.

[21] S. H. Muhammad *et al.*, « AfriHate: A Multilingual Collection of Hate Speech and Abusive Language Datasets for African Languages », 15 janvier 2025, *arXiv*: arXiv:2501.08284. doi: 10.48550/arXiv.2501.08284.

[22] S. Das, A. Dutta, K. Roy, A. Mondal, et A. Mukhopadhyay, « A Survey on Automatic Online Hate Speech Detection in Low-Resource Languages », 28 novembre 2024, *arXiv*: arXiv:2411.19017. doi: 10.48550/arXiv.2411.19017.

[23] M. Diallo, C. Fourati, et H. Haddad, « Bambara Language Dataset for Sentiment Analysis », 5 août 2021, *arXiv*: arXiv:2108.02524. doi: 10.48550/arXiv.2108.02524.

[24] « Intelligence Artificielle et le Natural Language Processing (NLP) », OpenStudio. Consulté le: 18 avril 2025. [En ligne]. Disponible sur: https://www.openstudio.fr/metiers/intelligence-artificielle/natural-language-processing-nlp/

[25] J. JVC, « Introduction au Machine Learning avec Python », Data Transition Numérique. Consulté le: 29 mars 2025. [En ligne]. Disponible sur: https://www.data-transitionnumerique.com/machine-learning-python/

[26] « Machine Learning Explained: Understanding Supervised, Unsupervised, and Reinforcement Learning ». Consulté le: 29 mars 2025. [En ligne]. Disponible sur: https://www.linkedin.com/pulse/machine-learning-explained-understanding-supervised-ronald-van-loon

[27] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann, 2012.

[28] « Apprentissage automatique supervisé ». Consulté le: 22 avril 2025. [En ligne]. Disponible sur: https://www.datacamp.com/blog/supervised-machine-learning

[29] « Introduction à l'apprentissage automatique : ce que vous devez savoir | Serverspace ». Consulté le: 12 avril 2025. [En ligne]. Disponible sur: https://serverspace.io/fr/about/blog/introduction-to-machine-learning/

[30] « Machine Learning: Algorithms, Real-World Applications and Research Directions ».

[31] S. Abro, S. Shaikh, Z. Hussain, Z. Ali, S. Khan, et G. Mujtaba, « Automatic Hate Speech Detection using Machine Learning: A Comparative Study », *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, n° 8, 2020, doi: 10.14569/IJACSA.2020.0110861.

[32] S. Boussouf, "La Reconnaissance du Langage Offensant dans le Contenu Arabe en Ligne," Master's thesis, Département Informatique, Université de Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj, Bordj Bou Arréridj, Algeria, 2024.

[33] T. J. Sejnowski, « The unreasonable effectiveness of deep learning in artificial intelligence », *Proc. Natl. Acad. Sci.*, vol. 117, n° 48, p. 30033-30038, déc. 2020, doi: 10.1073/pnas.1907373117.

[34] « Réseau de neurones artificiels - Définition et Explications », Techno-Science.net. Consulté le: 12 avril 2025. [En ligne]. Disponible sur: https://www.techno-science.net/glossaire-definition/Reseau-de-neurones-artificiels.html

[35] Gourav, « Introduction to Artificial Neural Networks », Analytics Vidhya. Consulté le: 12 avril 2025. [En ligne]. Disponible sur: https://www.analyticsvidhya.com/blog/2021/09/introduction-to-artificial-neural-networks/

[36] S. Kadri, "Chapitre 3. Apprentissage Profond," *Introduction à l'Intelligence Artificielle*. M'sila, Algérie : Université Mohamed Boudiaf de M'sila, Faculté des Mathématiques et de l'Informatique, 2020-2021.

[37] L. Alzubaidi *et al.*, « Review of deep learning: concepts, CNN architectures, challenges, applications, future directions », *J. Big Data*, vol. 8, n° 1, p. 53, mars 2021, doi: 10.1186/s40537-021-00444-8.

[38] C. Zhou, C. Sun, Z. Liu, et F. C. M. Lau, « A C-LSTM Neural Network for Text Classification », 30 novembre 2015, *arXiv*: arXiv:1511.08630. doi: 10.48550/arXiv.1511.08630.

[39] U. Singh, « Comparing Transformer Architectures for Sentiment Analysis: A Study of BERT, GPT, and T5 », vol. 11, n° 3.

[40] « GloVe: Global Vectors for Word Representation ». Consulté le: 12 avril 2025. [En ligne]. Disponible sur: https://nlp.stanford.edu/projects/glove/

[41] « fastText - Meta Research », Meta Research. Consulté le: 19 avril 2025. [En ligne]. Disponible sur: https://research.facebook.com/blog/2016/08/fasttext/

[42] « Modération de contenu pour l'intelligence artificielle ». Consulté le: 14 mai 2025. [En ligne]. Disponible sur: https://www.innovatiana.com/post/content-moderation-for-ai

[43] « Analyse de sentiments », *Wikipédia*. 10 mars 2025. Consulté le: 10 mai 2025. [En ligne]. Disponible sur: https://fr.wikipedia.org/w/index.php?title=Analyse_de_sentiments&oldid=223752469

[44] M. A. Nedioui, "Techniques d'apprentissage automatique pour l'analyse et la fouille des sentiments dans les réseaux sociaux," Thèse de doctorat, Département de l'Informatique, Université Mohamed Khider – Biskra, Biskra, Algérie, 2021.

[45] M. Laurelli, « Adaptive Meta-Domain Transfer Learning (AMDTL): A Novel Approach for Knowledge Transfer in AI », doi: https://doi.org/10.48550/arXiv.2409.06800.

[46] S. Sarwade, « La PNL simplifiée - Partie 2 - Types de techniques de vectorisation », Geekflare France. Consulté le: 31 mai 2025. [En ligne]. Disponible sur: https://geekflare.com/fr/nlp-simplified-vectorization-techniques/

[47] R. Kassel, « Hyperparamètres : Qu'est-ce que c'est ? À quoi ça sert ? », DataScientest. Consulté le: 28 mai 2025. [En ligne]. Disponible sur: https://datascientest.com/hyperparametres-tout-savoir.