ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre 3583

THESE

En vue de l'obtention du : ***DOCTORAT***

Structure de Recherche : Intelligent Processing & Security of Systems (IPSS)
Discipline : Informatique
Spécialité : Artificial Intelligence

Présentée et soutenue le 30/12/2021 par :

Randa ZARNOUFI

Techniques NLP Génériques et Systèmes D'apprentissage Automatiques pour l'Analyse du Texte Bruité : Cas de la Détection de la Cyberviolence

JURY

Fouzia OMARY	PES, Faculté des Sciences, Université Mohammed V, Rabat	Présidente
Azzeddine MAZROUI	PES, Faculté des Sciences, Université Mohammed I, Oujda	Rapporteur/Examinateur
Karim BOUZOUBAA	PES, Ecole Mohammadia d'Ingénieurs, Université Mohammed V, Rabat	Rapporteur/ Examinateur
Abdelhak MAHMOUDI	PH, Ecole Normale Supérieure-Rabat, Université Mohammed V, Rabat	Rapporteur/ Examinateur
Youness TABII	PH, Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes, Université Mohammed V, Rabat	Examinateur
Amine BOUT	PA, Faculté de Médecine, Université Sidi Mohamed Ben Abdellah, Fès	Invité
Mounia ABIK	PES, Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes, Université Mohammed V, Rabat	Directrice de Thèse

Année Universitaire : 2021/2022

Remerciement

La réalisation de cette thèse n'aurait été possible sans l'intervention de certaines personnes. Qu'elles trouvent ici l'expression de mes sincères remerciements pour leurs précieux conseils et leur contribution directe ou indirecte.

En premier lieu, je tiens à remercier Mme. Mounia ABIK, ma directrice de thèse. En plus de son encadrement pertinent, elle a toujours su m'encourager et m'orienter dans les moments difficiles. Grâce à son soutien, son aide précieuse, sa confiance inconditionnelle et sa disponibilité, j'ai pu progresser et arriver au terme de mes travaux. Elle restera pour toute ma carrière un exemple à suivre.

En deuxième lieu, j'exprime mes profonds remerciements à Mme Fouzia OMARY, Professeure à la Faculté des Sciences de Rabat et responsable de l'équipe IPSS qui m'a accueillie dans son équipe pendant mes années de recherche et qui a bien voulu présider le jury de ma thèse.

Mes sincères remerciements vont également à Mr. Azzeddine MAZROUI, Professeur à la Faculté des Sciences d'Oujda qui a accepté de juger ce travail, d'en être rapporteur et pour l'intérêt qu'il a porté à mon travail.

Je tiens à remercier Mr. Karim BOUZOUBAA, Professeur à l'Ecole Mohammadia d'Ingénieurs d'avoir eu la gentillesse d'accepter d'évaluer ce travail et d'en être rapporteur.

Je tiens à remercier très vivement Mr. Abdelhak MAHMOUDI, Professeur à l'Ecole Normale Supérieure de Rabat qui a accepté de juger ce travail et d'en être rapporteur.

J'adresse mes remerciements à Mr. Younes TABII, Professeur à l'Ecole Nationale Supérieure d'Informatique et d'analyse des Systèmes qui a accepté d'examiner ce travail et d'être membre de jury.

Je tiens à remercier également Mr. Amine BOUT, Professeur à la Faculté de médecine de Fès qui a accepté d'être membre de jury.

Mes remerciements les plus forts s'adressent à ma famille pour m'avoir encouragé. Sans elle, je n'aurais jamais pu achever ce travail et je ne serais pas présente en ce jour mémorable.

Enfin, à tous ceux que je n'ai pas pu citer, auxquels je réitère mes sincères remerciements.

إلى كل من شاركني هاته المسيرة

Résumé

Les médias sociaux (SM) débordent aujourd'hui de données textuelles, ces données peuvent être très utiles pour des applications de NLP et Text Mining; à titre d'exemple, celles conçues pour la détection de la cyberviolence qui est devenue un phénomène suscitant une intervention urgente. Cependant, ces textes générés par les utilisateurs des SM sont de nature bruitée ou non standard, ils contiennent des éléments bruyants dont : le *Code Switching* (CS), le *dialecte*, les mots mal orthographiés, les abréviations et les symboles. Tous ces aspects linguistiques rendent ce type de texte difficile à traiter avec les techniques *NLP* traditionnelles, et par conséquent, il faut les normaliser afin qu'ils prennent une forme standard.

Dans le cadre de cette thèse, notre contribution porte sur plusieurs axes. D'abord, afin de surmonter les problèmes susmentionnés, notre approche, de type *générique*, vise à la normalisation du texte bruité en exploitant *des ressources et des outils existants*. Le principal traitement consiste en la normalisation des phrases CS de plusieurs langues en une seule langue tout en préservant leurs sens. Ceci a été réalisé à travers une approche de type traduction automatique. Les autres opérations concernent la normalisation du dialecte et des expressions spéciales du SM ainsi la correction orthographique. Nous avons appelé l'ensemble de ces traitements '*Machine Normalization'*, ce processus sert de prétraitement précédant l'analyse du texte bruité. Puis, nous entamons le problème de la détection du *contenu violent* à partir du texte généré par les utilisateurs en ligne. Notre approche est basée sur les techniques *Ensemble Machine Learning*, que nous avons entraînées sur des caractéristiques liées aux caractères de la *personnalité* des utilisateurs, un choix inspiré des études faites en psychologie sur la cyberviolence. Cette approche a prouvé son efficacité devant les techniques Deep Learning dans un contexte de dataset déséquilibré et de taille réduite. Finalement, nous présentons notre *corpus annoté* pour la détection du contenu violent en Arabe Marocain.

Mots-clés : NLP, Analyse des Médias Sociaux, Normalisation du Texte Bruité, Détection des Contenus Violents, Caractères de la Personnalité Relatives à la Violence, Machine Learning.

Abstract

Today, Social Media (SM) user-generated text is the main resource for many Text Mining and NLP tasks. For instance, we can perform cyberviolence detection, which is a widespread online phenomenon calling for emergency response. This text, however, is of a noisy nature and does not follow the standard rules of writing. The main noisy elements composing this type of texts are: *Code Switching* (CS), *dialect*, misspelling, acronyms and symbols. All these linguistics forms avoid the processing of this text using classical *NLP* techniques which implies its normalization into a standard form.

We contribute to this thesis in several ways. First, to overcome the raised problems, we present our *language independent* approach for normalizing *noisy text* using *available tools and resources*. The main processing of our solution is CS sentence normalization from a mixture of language to one language while preserving its semantic. This processing has been achieved through a machine translation approach (MT-like) where the source languages and target one are *automatically* identified without human intervention. The other processing is the normalization of dialect and SM special lexicon in addition to spelling correction. We refer to these processing as '*Machine Normalization*', which can be used as a pre-processing step prior to the analysis of the noisy text. Second, we address the problem of detecting violent content from online user-generated text. We have based our approach on *Ensemble Machine Learning* models trained on features related to user *personality* following the findings of psychological studies on cyberviolence. This approach has proved to be more efficient than Deep Learning techniques in an unbalanced and small dataset context. Finally, we present our *dataset* annotated for violence content detection in Moroccan Arabic.

Key-words: NLP, Social Media Analysis, Noisy Text Normalization, Detection of Violent Content, Personality Traits Related to Violence, Machine Learning.

Contenu

Liste des Figures

Liste des Tableaux

Glossaire

AM: Arabe Marocain

BERT: Bidirectional Encoder Representations from Transformers

BOW: Bag Of Word

CBOW: Continuous Bag Of Words

CNN: Convolutional Neural Network

CS: Code Switching

CSMT: Character level SMT

DL: Deep Learning

LDA: Latent Dirichlet Allocation

LID: Language Identification

MA: Morphological Analyzer

MFT: Most Frequent Translation

ML: Machine Learning

MN: Machine Normalization

MT: Machine Translation

MVC: Multilingual Vertical Context

NC: Near Context

NLP: Natural Language Processing

NMT: Neural Machine Translation

OOV: Out Of Vocabulary

POS: Part of Speech

RBMT: Rule Based Machine Translation

RNN: Recurrent Neural Network

SM: Social Media

SMOTE: Synthetic Minority Over-sampling Technique

SMT: Statistical Machine Translation

SVM: Support Vector Machine

SVO: Subject Verb Object

TF-IDF: Term Frequency-Inverse Document Frequency

VSO: Verb Subject Object

WSD: Word Sense Disambiguation

Chapitre 1

Introduction

Durant ce chapitre d'introduction, nous présenterons le contexte ainsi que les motivations de cette thèse menée autour de la normalisation des textes bruités pour permettre la détection des traits de violence chez les utilisateurs en ligne. Ensuite, nous définirons l'acte de cyberviolence, le déclencheur majeur de ce travail, ses formes et ses manifestations. Puis, nous détaillerons les objectifs et la démarche de recherche suivie, pour finaliser ce chapitre par le plan de cette dissertation et un résumé des contributions.

1.1 Contexte

La récente prolifération mondiale de la technologie, en particulier d'internet, et de téléphonie mobile a participé largement à une utilisation sans précédent des médias sociaux. Aujourd'hui, internet est devenu un besoin nécessaire pour tout le monde. Surtout, avec l'arrivée du smartphone qui a facilité l'accès à ses services à travers des applications mobiles très conviviales et simples à utiliser sans se soucier des détails technologiques derrières. Cette facilité d'utilisation avec le faible coût des dispositifs tactiles ont accéléré la large expansion d'utilisation des plateformes des médias sociaux (SM).

Ces plateformes regroupent aujourd'hui des personnes de quatre coins du monde qui communiquent et partagent entre eux tout type de contenu : des textes, des images, des vidéos et aussi des audios, en toute liberté et sans limites. De plus, le contenu, personnalisé et très attractif, offert par ces plateformes a encouragé les utilisateurs à s'attacher à ces médias sociaux et à en devenir des fois très dépendants. En résultats, ces SM sont devenus débordants de données. Le nombre d'utilisateurs sur internet a augmenté de 83% de 2014 à 2019. Plus de 500 millions de tweets et 4 milliards de messages Facebook sont publiés chaque jour[1]. Ceci dit, ces médias sont devenus un moyen de communication important et surtout influent.

Les utilisateurs des SM profitent de cette opportunité pour exprimer et échanger librement, d'une façon bidirectionnelle, des contenus personnels ou informatifs concernant leurs états émotionnels, leurs avis et leurs opinions. Puisque le web est ouvert pour tout le monde, y compris les personnes souffrant de troubles psychologiques, alors, ces derniers, ayant toute la liberté d'expression, exploitent ces environnements pour diffuser leurs propres contenus. Du coup ils peuvent envoyer des contenus harassants à d'autres personnes en ligne en gardant l'anonymat pour ne pas être identifier.

De ce fait, la cyberviolence a fait son apparition et ne cesse d'accroitre, avec un nombre accru de victimes. Les effets négatifs de la cyberviolence sur ces victimes sont multiples. Ils peuvent souffrir de douleurs physiques et psychologiques importantes qui influencent négativement leur vie et leur bien-être. Par conséquent, un nombre important d'initiatives préventives ont vu le jour, comme l'organisation de campagnes de sensibilisation par les

[1] https://www.internetlivestats.com/

gouvernements et les ONG[2], ainsi que la création des numéros verts pour la réclamation des actes de violence aux autorités. Dans de nombreux pays développés, les politiques et les législations nationales qui interdisent toutes les formes de violence, y compris la cyberviolence, sont avancées. En plus, plusieurs recherches sur la santé électronique surtout celle mentale ont été conduites, tant en psychologie que dans le domaine computationnel, afin de cerner les actes de cyberviolence et limiter ces conséquences. Malgré tout, ces efforts restent insuffisants pour assurer une utilisation saine d'internet.

Au niveau computationnel, la technologie principalement employée est l'intelligence artificielle qui a influencé positivement l'avancement de différentes spécialités de soins de santé y compris la santé mentale. Dans ce sens, plusieurs applications ont été réalisées, comme par exemple, la détection de la dépression chez les utilisateurs en ligne (Cacheda et al., 2019), les psycho-chabots qui jouent le rôle de psychiatre électronique (e.g Woebot[3]) et l'initiative de Facebook pour la prévention du suicide[4] sur sa plateforme. Toutefois, le développement de cette technologie est très dépendant des données en qualité et en quantité. Dans le cas de la santé mentale, il va falloir collecter des données personnelles décrivant le comportement et la psychologie des individus. L'une des façons pour procurer de quantités massives de données sur les individus, est de recourir aux SM.

Les données offertes par les SM présentent plusieurs autres avantages aux chercheurs. Premièrement, le langage des médias sociaux est écrit dans des contextes sociaux naturels, souvent entre amis et connaissances (Park et al., 2015). Deuxièmement, les données peuvent être extraites gratuitement depuis plusieurs sources. Troisièmement, les utilisateurs des médias sociaux divulguent leurs informations personnelles à un rythme élevé et continu, puisque le sujet de discussion fréquent est eux-mêmes (Naaman et al., 2010). Finalement, les utilisateurs des SM présentent généralement leur vraie personnalité et non pas des versions idéalisées (Back et al., 2010; Hall and Caton, 2017).

Ainsi, le langage des SM est potentiellement une source très riche de données informatives sur la psychologie et la personnalité des individus. Par conséquent, pour détecter les actes de violence en ligne, nous pouvons exploiter les données diffusées dans ces

[2] Organisation Non Gouvernementale.
[3] https://woebothealth.com/
[4] https://www.facebook.com/safety/wellbeing/suicideprevention/

médias, lieu où des personnes souffrantes de troubles mentaux, tels que les auteurs de cyberviolence, sont engagées pour nuire à autrui.

Cependant, les textes rédigés par les utilisateurs des SM ne suivent pas les normes standards de rédaction. Ils peuvent contenir des mélanges de langues standard ou dialectale, des mots mal orthographiés, en plus d'un lexique spécial à ces SM. Ces formes linguistiques empêchent le traitement de ces textes avec les outils NLP (Natural Language Processing) et Text Mining[5] traditionnels. Ce qui nécessite tout d'abord la normalisation de ces textes pour qu'ils prennent une forme standard. Cette opération permettra ensuite l'analyse de ces textes pour en extraire les traits de violence de leurs auteurs.

Afin de mieux comprendre le phénomène de cyberviolence et ses différentes formes ainsi que l'ampleur de son impact sur la société, nous commençons d'abord par le définir.

1.2 Définition de la cyberviolence

Les SM sont un espace libre et ouvert, chacun peut partager, exprimer ses pensées et écrire ce qu'il veut sans aucun contrôle, parfois avec plus de liberté que dans la vie réelle, grâce à un certain degré d'anonymat offert par les plateformes SM. Par conséquent, les personnes cruelles utilisent le SM pour diffuser leur malveillance à d'autres personnes de leur réseau afin de se sentir puissantes, de se venger ou de tirer profit des autres (Kowalski et al., 2014).

Cette activité violente en ligne est appelée cyberviolence. Dans la littérature scientifique, les définitions sont nombreuses, mais généralement la cyberviolence peut être considérée comme un comportement d'un acteur qui a eu lieu en ligne par le biais de l'utilisation des systèmes informatique en particulier les ordinateurs, les téléphones mobiles et la technologie Internet (Slonje and Smith, 2008). Ce comportement est hostile et agressif, et peut également être offensant, obscène ou menaçant (Owen, 2016).

Jusqu'à ce jour, il n'existe pas encore de terminologie bien établie ou de typologie bien précise et stable des infractions considérées comme de la cyberviolence. De nombreux exemples de types de cyberviolence sont interdépendants ou se chevauchent ou consistent

[5] Text Mining est le processus d'analyse des textes (données non structurées) en vue d'en extraire des informations pertinentes.

en une combinaison d'actes[6]. Mais, malgré ce désaccord de termes utilisés, le concept de cyberbullying ou cyberharcèlement semble le plus dominant dans les études scientifiques (Berguer, 2015). Le cyberharcèlement implique un comportement agressif intentionnel de façon répétée à l'encontre d'une victime en présence d'un déséquilibre des forces (Paul et al., 2012). Toutefois, ce type d'agression ne couvre pas toutes les formes de violence en ligne, par conséquent dans notre étude nous considérons les actes de cyberviolence dans leurs globalités sans donner d'importance à des cas particuliers. La cyberviolence comprend de multiples formes d'agression, dont voici les plus communes :

- Harcèlement : insultes, menace, humiliation, troll[7], exclusion de groupes, flaming[8].
- Violation de la vie privée : diffamation, traque, dénigrement[9], usurpation d'identité, outing[10], la propagation de rumeurs.
- Propagation de la haine contre des groupes basé sur la race, l'ethnicité, la religion, le sexe, l'orientation sexuel, l'handicap.
- Abus sexuel et exploitation sexuelle : prostitution des enfants, violence dans les cyber-rencontres.
- Elle peut également impliquer des menaces directes ou des violences physiques : incitation à la violence, chantage.

Après avoir été harcelées, les victimes de la cyberviolence peuvent développer une variante de détresse psychologique et même physique, par exemple, dépression, anxiété, colère, repli sur soi, insomnie, troubles alimentaires, maladies physiques. Dans certains cas, ces victimes peuvent devenir des auteurs de cyberviolence, ou même envisager le suicide pour échapper à la souffrance morale (Sampasa-Kanyinga et al., 2014). Pour ces conséquences négatives, la cyberviolence a été décrite comme un "problème de santé publique émergent"

[6] https://rm.coe.int/t-cy-2017-10-cbg-study-fr-v2/1680993e65
[7] En argot Internet, un troll caractérise un individu ou un comportement qui vise à générer des polémiques. (Wikipédia).
[8] Le flaming consiste à envoyer un message violent (insultant, menaçant) et plus particulièrement une « salve » de messages à destination d'une personne ou d'un groupe de personnes.
[9] Le dénigrement consiste à décrédibiliser une personne, à porter atteinte à son image, à sa réputation.
[10] L'outing consiste à divulguer des informations intimes et /ou confidentielles sur une personne.
(Source : https://eviolence.hypotheses.org/187)

(David-Ferdon and Hertz, 2007).

Les chiffres existants sur la prévalence de la cyberviolence sont très parlants. Au niveau mondial, l'enquête menée par *ditch the label*[11] en 2017 montre que 17% des personnes ont été exposées à l'une des formes de cyberviolence. Selon UNESCO (2017) Les données disponibles sur la prévalence du cyberharcèlement suggèrent que de 5 à 21 % des enfants et des adolescents en sont victimes, les filles y étant plus exposées que les garçons.

Aux États-Unis, selon une enquête de 2013 sur les comportements à risque des jeunes, 15 % des enfants de la 9ème à la 12ème année avaient été harcelés en ligne. Les filles étaient plus de deux fois plus susceptibles que les garçons (21 % contre 9 %, respectivement).

En Europe, suivant une enquête portant sur 25 pays européens, il a été observé qu'entre 2010 et 2014, la proportion d'enfants et d'adolescents âgés de 9 à 16 ans ayant été exposés au cyberharcèlement était passée de 8 % à 12 %, en particulier chez les filles et les enfants les plus jeunes (UNESCO, 2017).

Au Maroc, d'après le Haut-Commissariat au Plan[12], 14%, ou près de 1,5 million de femmes sont victimes de violence électronique. Le risque d'être victime de ce type de violence est plus élevé parmi les citadines (16%), les jeunes femmes âgées de 15 à 19 ans (29%), celles ayant un niveau d'enseignement supérieur (25%), les célibataires (30%) et les élèves et étudiantes (34%). Une enquête faite par Microsoft[13] en 2012, a montré qu'au Maroc 40% (contre une moyenne de 37% dans 25 pays) des enfants âgés de 8 à 17 ans ayant répondu à l'enquête disent avoir été victimes de toute une série de cyberviolences.

Finalement, il est intéressant à noter que toutes les formes ou tous les cas de cyberviolence ne sont pas aussi graves et ne nécessitent pas forcément une solution pénale, mais ils peuvent être traités par une approche progressive et une combinaison de mesures préventives, éducatives, protectrices et autres.

Les grands acteurs du numérique, en particulier GAFAM[14] dépensent chaque année des

[11] https://www.ditchthelabel.org/wp-content/uploads/2017/07/The-Annual-Bullying-Survey-2017-1.pdf
[12] https://www.hcp.ma/Communique-du-Haut-Commissariat-au-Plan-a-l-occasion-de-la-campagne-nationale-et-internationale-de-mobilisation-pour-l_a2411.html
[13] https://tinbergens2.files.wordpress.com/2015/02/ww-online-bullying-survey-executive-summary-morocco_final-3.pdf
[14] GAFAM : Google, Apple, Facebook, Amazon et Microsoft.

millions d'euros pour réviser manuellement les contenus en ligne et supprimer les documents volumineux. Facebook a publié son rapport d'audit[15] sur les droits civils, qui explique sa stratégie de lutte contre les abus et les contenus agressifs. Le rapport affirme qu'il n'est pas possible de construire un système d'automatisation complet pour détecter le discours de haine et que la modération des contenus est inévitable. Malgré ces efforts, on leur reproche encore de ne pas en faire assez et on leur fait subir une pression croissante pour régler ce problème.

1.3 Objectifs et démarche de la recherche

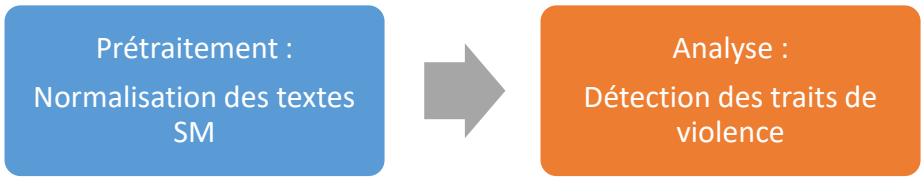

Figure 1.1. Etapes suivies pour la détection des traits de violence

L'objectif principal de cette thèse est la normalisation des textes bruités pour la détection des traits de violence à partir des écrits des utilisateurs en ligne. Afin d'atteindre ce but, nous avons tracé des objectifs partiels et de nature séquentielle (Figure 1.1).

Le premier objectif de cette recherche est le prétraitement des textes issus des réseaux sociaux. Ce prétraitement sert à la normalisation des formes non standards constituants les textes SM afin de les rendre standards, et donc, pouvoir les exploiter dans des tâches d'analyse, et d'en extraire des informations pertinentes pour une application cible donnée.

Le point de départ de cette première étape était de répondre aux questions :

- *Quels sont les types de bruits dont souffrent les textes des SM et qui bloquent leur traitement par les outils NLP classiques et comment pouvons-nous pallier ce problème et convertir un texte non standard à un texte standard ?*

- *Est-il possible de faire le traitement des textes bruités issues des réseaux sociaux sans recourir au développement des outils linguistiques fondés sur les règles, et non plus sur des techniques d'apprentissage basées sur les corpus annotés ?*

[15] https://www.theverge.com/interface/2019/7/2/20678231/facebook-civil-rights-audithate-speech-moderators

Les techniques basées sur les règles nécessitent l'exploration des règles linguistiques relatives à une langue donnée, ce qui fait que celles-ci soient dépendantes de la langue, et donc elles ne seront utiles que pour une langue particulière. Tandis que l'approche basée sur les corpus demande le développement de corpus contenants d'énormes quantités de données annotées. Cependant, la construction de tels corpus n'est pas une tâche facile compte tenu qu'elle est très coûteuse et prend du temps (ce type de corpus passe par la collecte, l'annotation et la validation par les experts).

Pour répondre à la question susmentionnée et remédier à ces limites, nous visons à bâtir une solution générique valable pour toutes les langues utilisant des outils et des ressources multilingues existants, ce qui permet de reproduire nos techniques facilement sans avoir à construire de nouveaux outils ou ressources. En outre, nous essayons de couvrir tous les phénomènes linguistiques qui participent au bruit perturbant et limitant l'application des techniques de NLP existantes. Dans ce sens, nous donnons des solutions appropriées pour traiter chaque type de bruit, en particulier à la normalisation du code switching, l'un des défis de la NLP, qui consiste à convertir des phrases écrites en plusieurs langues à une phrase monolingue. De plus, nous attaquons le problème de l'orthographie non standard des dialectes avec une solution extensible valable pour tout type de dialecte.

Après le prétraitement, le deuxième objectif vise l'analyse des textes afin de détecter les traits de violence de leurs auteurs.

Les questions de recherche à examiner ici sont les suivantes :

- *Est-ce qu'il y a un caractère commun caractérisant tous les types de cyberviolence ?*
- *Est-ce que les caractères de la personnalité sont des forts prédicteurs des traits de violence ?*

Afin de répondre à cette question de recherche, nous avons opté pour une technique reposant sur des approches supervisées de machine learning classique avec des caractéristiques d'apprentissage liées à des recherches en psychologie sur les caractères et la personnalité des auteurs des actes de violence en ligne. Ce choix de ML classique à la place des algorithmes de deep learning (DL) est relatif à notre volonté de détecter le comportement violent et dégager les caractéristiques de la personnalité du cyber-perpétrateur responsables

des actes de violence. Alors que les algorithmes DL ne sont pas interprétables et très gourmands en quantité de données annotées.

1.4 Plan de la thèse

Le reste de ce chapitre décrit la thèse et présente un résumé des principaux résultats des recherches conduites.

Le chapitre 2 introduit d'abord, un ensemble de techniques qui seront employées dans plusieurs tâches de cette thèse, puis il parcourt les différents travaux connexes réalisés autour des principales problématiques de la présente étude. En premier lieu, il couvre les travaux sur la normalisation des textes issus des médias sociaux qui sont de nature bruitée, notamment à travers la conversion du code switching, du multilingue au format monolingue, tout en respectant la sémantique des mots traduits et aussi la normalisation des translittérations dialectales. En deuxième lieu, ce chapitre présente les différentes techniques employées dans la littérature pour détecter le contenu violent dans les messages générés par les utilisateurs en ligne lié à leur comportement violent.

Le chapitre 3 fournit une vue détaillée sur notre contribution pour la normalisation des textes bruités, en commençant par un diagnostic des phénomènes linguistiques qui touchent la formation des mots et des phrases dans les réseaux sociaux et qui empêchent leur traitement par les outils classiques existants. Puis, il présente les différentes démarches suivies pour le prétraitement du texte bruité afin de le rendre exploitable par d'autres tâches NLP ou Text Mining qui visent son analyse. Ces démarches couvrent, la traduction du code switching de plusieurs langues vers une seule langue, fondée sur des théories linguistiques, tout en respectant l'aspect sémantique des phrases. Ce traitement est précédé par une étape d'identification de langues, de correction orthographique et de normalisation de dialecte ainsi que celle des différents types de lexiques spécifiques aux réseaux sociaux.

Le chapitre 4 décrit la démarche suivie dans la détection du comportement violent des individus en nous servant du contenu de leurs messages sur les réseaux sociaux. L'apprentissage machine supervisé a été utilisé en se reposant sur l'ingénierie des caractéristiques relatives à la personnalité, en particulier, les émotions ainsi que les Big Five traits de personnalité (l'ouverture, la conscience, l'extraversion, l'agréabilité et le

neuroticisme). Cette technique sera comparée avec l'apprentissage profond. Ce chapitre sert aussi à présenter le corpus de violence que nous avons construit durant ce travail.

Le chapitre 5 conclut cette thèse en résumant ses principales contributions et leurs résultats. En outre, il suggère des champs applicatifs potentiels des solutions présentées ici, et met en évidence les défis qui sont toujours ouverts ainsi que les futurs objectifs qui seront réalisés comme suite à cette thèse.

1.5 Résumé des contributions

Durant cette thèse nous avons réalisé plusieurs contributions que nous pouvons classer en deux catégories. La première est liée à la phase de prétraitement qui vise la normalisation des textes bruités, tandis que la deuxième concerne la phase d'analyse en vue de détecter les traits de violence chez les utilisateurs des SM.

1.5.1 Phase de prétraitement

Dans le chapitre 3 nous présentons trois contributions relatives à cette phase :

'Machine Normalization' (Zarnoufi et al., 2020b) était la 1ère contribution, il s'agit d'un prétraitement des textes issus des médias sociaux qui sont de nature bruitée pour les transformer sous format standard avant d'être utilisés dans des applications Text Mining par exemple. Le traitement principal dans ce travail concerne la normalisation du code switching. L'approche adoptée est celle de traduction automatique (machine translation like) hybride combinant les règles et les statistiques. Le processus passe par trois étapes : l'analyse, le transfert et la génération. L'objectif est de passer d'une phrase multilingue, trois langues pour notre cas, vers une autre monolingue.

Dans la deuxième contribution (Zarnoufi et al., 2019), nous avons proposé une technique hybride basée sur les connaissances (analyseurs morphologiques) et des modèles statistiques (modèles de langage). L'objectif de cette technique est l'identification de langue au niveau de chaque mot dans une phrase contenant du code switching, ainsi que l'identification du sens des mots spéciaux aux médias sociaux en utilisant des dictionnaires spécifiques, et finalement la correction orthographique multilingue des mots non reconnus.

Le troisième travail (Zarnoufi et al., 2020c) traite le problème de la normalisation du

dialecte de l'Arabe Marocain, qui n'a pas d'orthographie standard. Chaque mot, peut être écrit sous multiples formes de translittérations, en augmentant ainsi la complexité des tâches NLP. Pour uniformiser ces formes d'écriture, notre solution fonctionne en faisant la correspondance entre les formes canoniques des mots, fournies par un dictionnaire d'Arabe Marocain, et leurs différentes translitérations, les plus similaires, produites par trois modèles word embedding. Ces modèles étant, Word2Vec, CBOW et Skip-gram, et FastText entraînés sur un corpus de commentaires YouTube écrit en Arabe Marocain.

1.5.2 Phase d'analyse

Quant à cette phase, nous présentons les détails de deux principales contributions en chapitre 4.

Dans la première contribution (Zarnoufi et al., 2020a), nous avons travaillé sur l'identification du comportement violent des utilisateurs en ligne. Nous avons adopté l'approche d'apprentissage supervisé avec les algorithmes Ensemble Learning (Random Forest, Gradient Boosting, XGBoost et AdaBoost) avec, comme caractéristiques d'apprentissage, un ensemble de huit émotions (colère, peur, anticipation, confiance, surprise, tristesse, joie et dégoût). Pour l'extraction des caractéristiques, nous avons utilisé le lexique d'émotion EmoLex. En plus de la similarité lexicale, nous avons utilisé un modèle de word embedding pour effectuer la similarité sémantique et ainsi améliorer le processus d'extraction. Nous avons validé l'approche sur un corpus annoté d'harcèlement en ligne. Le modèle le plus performant était XGBoost.

Dans le deuxième travail (Zarnoufi and Abik, 2020), nous avons étudié la relation entre les Big Five traits de personnalité du cyber-auteur et son comportement violent pour montrer son impact sur la détection de la cyberviolence. Pour ce faire, nous avons utilisé un ensemble de lexiques associés au Big Five traits de personnalité (l'ouverture, la conscience, l'extraversion, l'agréabilité et le neuroticisme) et un dataset de tweets pré-labélisés d'harcèlement en ligne en plus des algorithmes Ensemble ML (Random Forest, Gradient Boosting, XGBoost et AdaBoost). Nous avons utilisé un lexique lié aux cinq grandes facettes de la personnalité que nous avons enrichies en utilisant le vocabulaire généré par le word embedding. Le modèle le plus performant était AdaBoost.

Chapitre 2

Fondements Théoriques et Etat de l'Art

Tout au long ce premier chapitre consacré à l'état d'art et les fondements théoriques, nous ferons un survole sur les travaux antécédents relatifs aux problématiques soulevées dans ce présent travail, et qui ont fait l'objet de nos principales contributions. En l'occurrence, les travaux qui s'intéressent aux différents aspects de la normalisation des textes issus des médias sociaux. Comme ceux réalisés sur la normalisation ou la traduction du Code Switching, ainsi que la désambiguïsation des sens des mots et la normalisation des dialectes. Nous présenterons également, les travaux liés à la détection du contenu violent sur les médias sociaux. Toutefois, nous aborderons tout d'abord les principales techniques dont reposent plusieurs traitements de cette recherche.

2.1 Fondements théoriques

Dans cette section, nous définissons les techniques que nous avons employées dans cette thèse. Notamment, les systèmes d'apprentissage automatique utilisés en classification, la traduction automatique ou machine translation (MT) et les techniques de représentation des mots. Mais, nous commençons d'abord par une brève définition du NLP.

2.1.1 Le NLP (Natural Language Processsing)

Le NLP ou le traitement automatique du langage naturel (texte, parole et image) est une discipline qui s'intéresse à l'étude et au développement des techniques qui visent l'automatisation des tâches relatives au langage. L'objectif principal du NLP est de faciliter la communication homme-machine ce qui permettra de déléguer plusieurs tâches à la machine. Le NLP se situe à l'intersection de plusieurs autres disciplines, nous citons, entre autres, l'intelligence artificielle, la linguistique et la science cognitive. Ces applications sont aujourd'hui partout présentes, en partant des détecteurs de spams et en passant par les moteurs de recherche et les traducteurs automatiques jusqu'aux assistants vocaux (Google Assistant, Alexa, Siri...). Les techniques adoptées par le NLP ont passé historiquement par les approches basées sur les règles, puis par les statistiques et finalement par l'apprentissage automatique.

2.1.2 Apprentissage automatique ou machine learning

Les techniques de machine learning (ML) permettent de construire des modèles mathématiques à partir de données. Chaque entrée (aussi appelée exemple ou échantillon) de ces données est représentée par un vecteur d'un ensemble de caractéristiques, où chaque caractéristique contient la valeur d'un attribut particulier des données. L'ensemble de ces entrées constituent un dataset, qui peut être vu comme une matrice de $n \times d$, avec n le nombre d'échantillons (les lignes), et d le nombre de caractéristiques (les colonnes).

Considérant le type de données, la construction d'un modèle ML peut être faite à travers un *apprentissage supervisé*, où chaque échantillon possède un label (valeur continue ou discrète), comme il peut être *non supervisé* si les labels ne sont pas disponibles. En cas d'apprentissage supervisé, lorsque les labels sont des valeurs continues, nous disons alors qu'il s'agit d'une tâche de *régression.* Lorsque les labels sont des valeurs discrètes (catégories), cette tâche est appelée classification.

Concernant la partie algorithmique, nous distinguons entre deux types de ML, le premier c'est le ML classique et le deuxième c'est le deep learning (DL).

2.1.2.1 Techniques ML classique

Quant aux techniques ML classique, cette tâche se base sur deux grandes étapes. La première est l'ingénierie de caractéristiques d'apprentissage (features engineering), où le choix de ces caractéristiques est fait sur la base d'expertise dans le domaine applicatif, suivi de l'extraction de ces caractéristiques à partir des données d'entrée. La seconde étape commence par l'apprentissage (training) du modèle sur une partie du dataset, suivi par un test de la capacité de prédiction du modèle généré, sur une partie de données non vues précédemment. Les algorithmes ML classique sont nombreux, nous citons à titre d'exemple : les SVM (Support Vector Machine) et les arbres de décision (Decision Trees).

a. Support Vector Machine

Une machine à vecteur support (SVM) (Cortes and Vapnik, 1995) est un algorithme d'apprentissage automatique supervisé, qui peut être utilisé à des fins de classification. Les SVM sont basés sur l'idée de trouver un hyperplan qui sépare au mieux un ensemble de données, étant linéairement séparables en deux classes. En d'autres termes, chercher un hyperplan maximisant la marge entre ces classes, ce qui réduit la possibilité d'erreur. Le même principe est utilisé pour la classification multi-classe, après avoir décomposé celle-ci en plusieurs problèmes de classification binaire. Si les données ne sont pas linéairement séparables, dans ce cas, l'algorithme projette ces données dans un espace de plus grande dimension avant de chercher un hyperplan de séparation dans cette nouvelle dimension.

b. Arbres de décision (Decision Trees)

Ils font partie des algorithmes ML les plus utilisés, ils sont à la base d'autres techniques plus performantes comme le Random Forest (voir section 4.3.2.2-b). Leur principe simule le

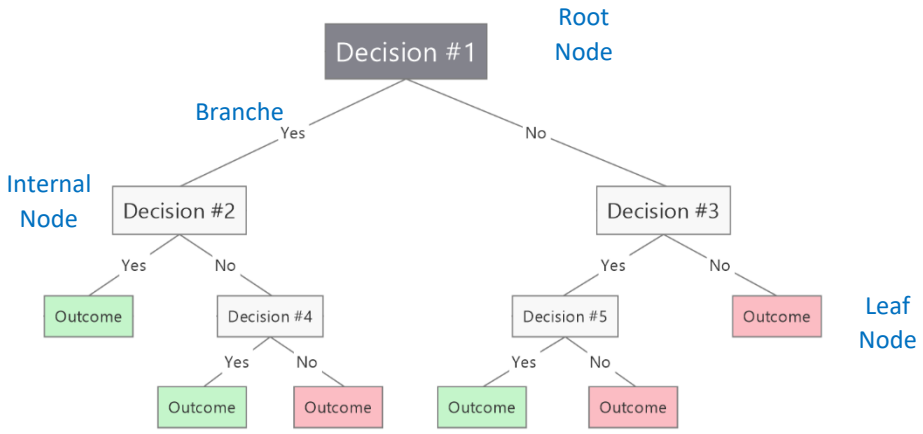

Figure 2.1 Architecture d'un Arbre de Décision

processus humain de prise de décision. Un arbre de décision est un arbre où chaque nœud évoque une caractéristique (règle de décision ou condition) en fonction de laquelle l'arbre se répartit en branches représentant une décision. L'extrémité d'une branche est un nœud feuille marquant un résultat (un label d'une valeur catégorique en cas de classification). L'idée générale est de créer un tel arbre pour l'ensemble de données et de produire un seul résultat par feuille (ou de minimiser l'erreur dans chaque feuille). Afin de créer un arbre, il faut d'abord trouver un nœud racine et puis les autres nœuds parmi les caractéristiques d'apprentissage. Pour ce faire, l'algorithme cherche la caractéristique qui classifie le mieux les données d'entraînement pour l'appliquer à la racine de l'arbre, puis il répète ce processus à chacune des branches.

2.1.2.2 Techniques DL

Regardant les techniques DL, ce sont des algorithmes multicouches qui essayent d'apprendre des représentations à partir de données. Le DL repose sur les réseaux de neurones (neurones artificiels connectés entre eux) en grande quantité organisés sous forme de couches. Les couches sont classées en trois niveaux : i) une couche d'entrée qui reçoit les données, ii) d'un ensemble de couches cachées dont chacune reçoit les sorties d'informations de la couche précédente et iii) d'une couche de sortie qui fournit les décisions.

Les connexions entre chaque couche et celle du niveau inférieur reposent sur une fonction d'activation et un ensemble de pondération. La valeur de chaque nœud de la couche cachée est transformée par une fonction non linéaire avant d'être transférée aux sommes

pondérées de la couche suivante. L'apprentissage se passe à travers un processus d'évaluation-correction de poids, répété plusieurs fois afin de minimiser une fonction de perte (loss function). Ce processus est appelé *backpropagation*.

Les architectures DL les plus connues sont les modèles seq-2-seq, étant des réseaux *CNN* et *RNN*, et les modèles *Transfomer*, leurs définitions seront données dans ce qui suit.

2.1.2.3 Convolutional Neural Networks (CNN)

L'architecture CNN ou convnets (Lecun et al., 1990) a été originalement inventée pour les applications de vision sur ordinateur (computer vision). Par la suite, le CNN a tracé son chemin avec succès dans les tâches NLP. Le CNN est composé de trois types de couches principales :

1- Couche convolutionnelle : elle utilise des filtres qui scannent l'entrée, suivant ses dimensions en effectuant des opérations de convolution dans le but d'extraire les caractéristiques de l'entrée. Les filtres sont les neurones de cette couche. La sortie de cette couche est appelée feature maps.

2- Couche pooling : elle effectue une opération de sous-échantillonnage sur les feature maps en réduisant leurs tailles. Elle prend la valeur moyenne ou bien maximale de l'entrée pour construire une autre feature map plus concise et moins redondante.

3- Couche fully-connected : elle est utilisée à la fin du réseau pour créer d'une part des combinaisons des caractéristiques reçus des couches précédentes, et d'autre part pour faire les prédictions. Elle opère sur une entrée préalablement aplatie[16] (flattened) où chaque neurone est connecté à tous les neurones.

Le CNN est utilisé dans plusieurs tâches NLP comme la classification du texte. Puisqu'un texte est constitué d'une séquence de données, il est alors considéré uni-dimensionnel (1D). Par conséquent, il peut être traité par les convnets 1D qui ont effectivement prouvé leurs performances dans ces tâches. Dans ce cas, le réseau CNN est considéré comme un extracteur de caractéristiques qui seront injectées dans d'autres types de réseaux neuronaux tel que le

[16] Flattening : convertir une matrice de données en un vecteur unidimensionnel pour les transmettre à la couche suivante.

RNN ou autres.

2.1.2.4 Recurrent Neural Networks (RNN)

Le RNN (Rumelhart et al., 1986) est un autre type de réseaux de neurones dont la particularité est qu'il possède une mémoire. Le RNN domine les problèmes d'apprentissage automatique impliquant des entrées en forme de séquences où l'ordre est signifiant. Cela revient au fait que le RNN permet de représenter des entrées séquentielles de taille arbitraire par des vecteurs de taille fixe, tout en tenant compte des propriétés structurées des entrées grâce à sa mémoire.

La fonction de mémorisation dans les RNN classiques est réalisée à travers des connexions en boucle (loops). Cette mémoire permet à ce type de réseau d'apprendre des séquences d'entrées plutôt que de traiter chaque entrée indépendamment. Les points de données sont traités un par un et non pas d'un seul coup, comme c'est effectué par les autres types de réseaux.

Le RNN appliqué au texte, il le traite mot par mot itérativement, en conservant une mémoire des mots précédents. Ce qui permet une meilleure représentation du sens des phrases. Ce processus est similaire à celui biologique suivi pendant la lecture.

Un RNN peut être vu sous forme de multiples copies du même réseau, chacune transmettant un message à son successeur. L'un des principaux atouts des RNN est leur capacité à relier des informations antérieures à la tâche en cours. Cependant, mis en fonctionnement, ces réseaux ne sont capables d'apprendre que des courtes dépendances. Pour surmonter cette limitation d'autres architecture RNN ont vu le jour, dont les plus répandus sont Long-Short-Term Memory (LSTM), Bidirectional LSTM (BLSTM) et Gated Recurrent Unit (GRU).

Les réseaux LSTM (Hochreiter and Schmidhuber, 1997) disposent de blocs de mémoire qui sont connectés en couches. La structure de ces mémoires leur permet de sauvegarder les informations les plus anciennes et de les réinjecter à un moment ultérieur, tout en évitant que les signaux ne disparaissent progressivement au cours du traitement. Ce qui rend ces réseaux capables d'apprendre des dépendances long terme en dépassant ainsi une limite des RNN classiques.

Les GRU (Cho et al., 2014) fonctionnent avec le même principe que les LSTM, mais elles sont légèrement simplifiées et donc moins coûteuses à faire fonctionner et aussi avec moins de pouvoir représentationnel.

Les BLSTM (Schuster and Paliwal, 1997) constituent une autre variante de LSTM fonctionnant en deux directions. Ils sont composés de deux réseaux LSTM, l'un opère sur la direction chronologique et l'autre sur la direction inverse (ordre antichronologique), puis ils combinent leurs représentations. Leur avantage est de pouvoir capter des patterns, peut-être omis, par le réseau unidirectionnel, et construire ainsi des représentations plus riches.

2.1.2.5 Transformer

Bien qu'ils soient performants, les modèles seq-2-seq (dotés du mécanisme d'attention) présentent certaines limites : La gestion des dépendances à longue portée (contexte long) et la nature séquentielle de leur architecture empêchant le parallélisme. Ces limites ont été surmontées par le concept de *Transformer*. L'idée derrière Transformer est de gérer

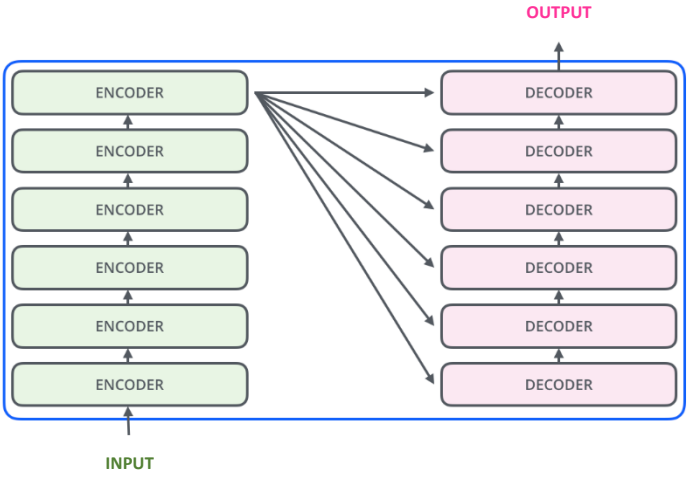

Figure 2.2 Architecture externe d'un Transformer

complètement les dépendances entre l'entrée et la sortie, en adoptant le mécanisme d'*attention* et le fonctionnement *récurrent*. Cette architecture a permis de gagner énormément du temps de calcul grâce à son traitement parallèle.

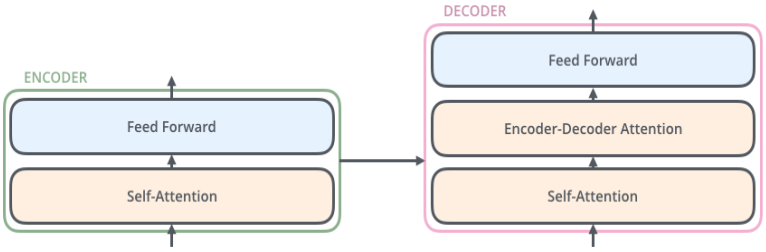

Figure 2.3. Architecture interne d'un Transformer

Les Transformers sont composés de plusieurs encodeurs et décodeurs identiques empilés les uns sur les autres (Figure 2.2), tous dotés d'auto-attention (Figure 2.3). Celle-ci permettra au décodeur de se focaliser sur les parties appropriées de la séquence d'entrée. Le fonctionnement du Transformer se déroule comme suit :

- Les word embeddings (voir section suivante) de la séquence d'entrée sont transmis au premier encodeur.
- Ils sont ensuite transformés et transmis à l'encodeur suivant.
- La sortie d'attention du dernier encodeur est transmise à tous les décodeurs (à leur propre couche d'attention).

2.1.3 Représentation distributionnelle des mots

Tout au long de cette recherche, nous avons employé des modèles de représentation distributionnelle des mots, en particulier le *word embedding*. Il est donc utile de donner un bref aperçu de ces représentations. Premièrement, nous exposons les modèles les plus répandus du word embedding. Par la suite, nous parlons brièvement de *contextual embedding*, étant la plus récente des techniques de représentations des mots.

Le word embedding est l'une des représentations les plus répandues du vocabulaire des documents. Il est capable de capturer le contexte d'un mot dans un document, ainsi que la similarité sémantique et syntaxique, la relation avec d'autres mots, etc. Le word embedding est en effet une représentation vectorielle dense de faible dimension d'un mot donné avec des valeurs réelles. L'idée principale de cette approche est que les mots apparaissant souvent dans un contexte similaire (entourés des mêmes mots voisins) ont tendance à être sémantiquement similaires.

Le word embedding permet de représenter chaque mot par un vecteur à N dimensions composé de valeurs en virgule flottante. Une autre façon de concevoir un embedding est de le considérer comme une table de correspondance (*lookup table).* Après la phase d'apprentissage, nous pouvons coder chaque mot en recherchant le vecteur dense auquel il correspond dans le tableau.

Parmi les applications les plus intéressantes des modèles word embedding, nous citons la saisie automatique. Dans de tel système, lorsqu'on fournit une partie de la phrase, le modèle prédit le mot suivant ou même un mot manquant dans la phrase. En plus cette représentation vectorielle de mots permet d'utiliser ces vecteurs dans un grand nombre de tâches du traitement de langage. En effet, nous pouvons alimenter des algorithmes de classification classiques (tels que SVM et Random Forest) en considérant ces vecteurs parmi les caractéristiques d'apprentissage du classifieur. Ces vecteurs sont aussi utilisés au niveau de la couche d'entrée des réseaux de neurones pour l'encodage des données. Nous pouvons encore les utiliser pour la mesure de similarité entre mots, phrases, ou documents.

Parmi les techniques les plus populaires du word embedding, nous citons le Word2Vec (Mikolov et al., 2013) qui a été développé chez Google. D'autres modèles similaires sont aussi très connus, en particulier Glove (Pennington et al., 2014) de Stanford et FastText (Bojanowski et al., 2017) de Facebook.

Très récemment, une nouvelle génération d'embedding a vu le jour, appelée *contextual embedding* dont le modèle le plus connu est BERT (Devlin et al., 2019) (encore chez Google). Ce dernier a battu les records des performances dans plusieurs tâches NLP.

2.1.4 Word Embedding

2.1.4.1 Word2Vec

C'est un simple modèle neuronal qui se sert d'un réseau neuronal peu profond. Il est capable de calculer la probabilité d'un mot, compte-tenu de son contexte. Il permet de générer des vecteurs de mots denses où les mots similaires sont proches dans un espace compressé de faible dimension. L'idée principale est d'entrainer un réseau de neurones avec une seule couche cachée où la matrice de poids de cette couche correspond aux vecteurs de mots. Chaque réseau est composé donc d'une couche d'entrée, d'une seule couche cachée et d'une couche décisionnelle de sortie.

Deux modèles d'apprentissage différents ont été introduits pour entrainer le word embedding : ce sont le modèle du « *sac de mots continus* », ou CBOW (Continuous Bag Of Word) et le modèle de « *saut de mots continu* » ou Continuous Skip-gram. Nous présenterons en détails l'architecture et le fonctionnement des deux modèles dans les sous-sections suivantes.

a. CBOW

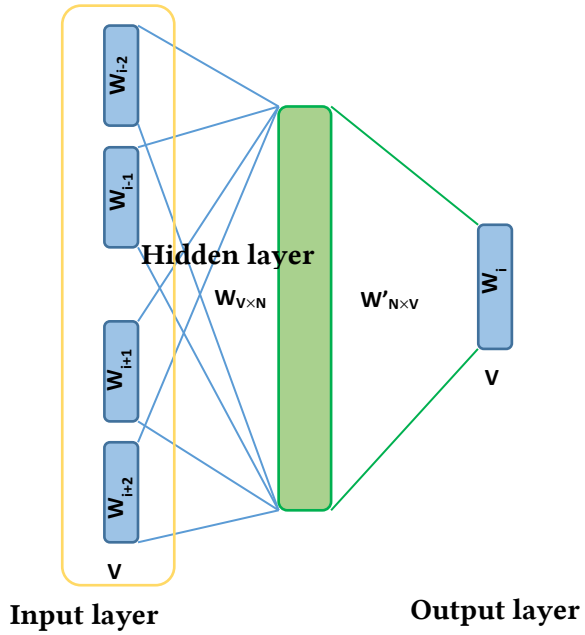

Figure 2.4. Architecture CBOW

Le modèle CBOW apprend à prédire le mot central en fonction de son contexte. Le contexte est constitué de quelques mots avant et après le mot (du milieu) concerné. En d'autres termes, l'entrée du réseau de neurones dans le cadre du CBOW prend une fenêtre autour du mot et essaie de prédire le mot central en sortie. Cette architecture est appelée modèle de « sac de mots continus », *sac de mots* signifie que l'ordre des mots dans le contexte n'est pas considéré, et *continu* signifie que ces mots possèdent une représentation vectorielle continue.

Dans cette architecture, l'entrée est composée de mots de contexte qui sont des vecteurs codés à chaud (one-hot[17]) de taille V (Figure 2.4). La couche cachée contient N neurones

[17] L'encodage One-hot permet de représenter un texte sous format numérique (vecteurs) adéquate aux algorithmes de machine learning. En principe, cette représentation associe chaque mot à un index unique du vocabulaire constituant un corpus. Pour un vocabulaire de taille $\boldsymbol{V}$, un vecteur d'un mot $\boldsymbol{w}$ (de taille V) sera

(dimensions). $W_{V\times N}$ est la matrice de poids qui fait correspondre l'entrée à la couche cachée (matrice dimensionnelle V×N). $W'_{N\times V}$ est la matrice de poids qui fait correspondre les sorties de la couche cachée à la couche de sortie finale (matrice dimensionnelle N×V). Les neurones de la couche cachée ne font que reproduire la somme pondérée des entrées dans la couche suivante. La sortie est à nouveau un vecteur de longueur V dont les éléments sont les valeurs de *softmax*[18] *qui est* un classifieur log-linéaire. Les vecteurs word embedding peuvent être extraits de la matrice des poids $W'_{N\times V}$.

Pendant son apprentissage, le réseau compare ses valeurs prédites avec celles réelles, puis il corrige les représentations vectorielles générées en utilisant la technique de rétropropagation.

b. Skip-gram

Le modèle de saut continu apprend à prédire les mots voisins à partir d'un mot central sur une fenêtre déterminée à l'avance. Il est appelé skip-gram parce que le modèle, en créant les instances d'apprentissage, peut sauter certains mots de contexte. Donc les mots du contexte ne sont pas forcément consécutifs. Cette idée a été proposée initialement par (Guthrie et al., 2006) dans l'objectif de pallier le problème de manque de données.

Dans le cas du skip-gram (Figure 2.5), l'entrée du réseau est constituée d'un seul vecteur codé à chaud qui correspond au mot cible. Ce vecteur sera multiplié par la matrice de poids associée à la couche cachée. La sortie de cette dernière est le vecteur de mot cible (word-vector). Ce vecteur sera transmis à la couche de sortie pour produire des distributions de probabilité (context-vectors), une pour chaque mot de contexte. Les mots prédits seront comparés aux mots de contexte réels, puis les vecteurs générés seront corrigés par rétropropagation.

encodé par un **0** sur toutes les dimensions sauf celle où apparaît le mot ***w*** qui sera à **1**, comme par exemple [0 0 0 0 0 1 0 0 0 0 0 0...0 0 0 0 0 0 0 0... 0]. Cette technique est efficace pour le stockage et le calcul, cependant, elle ne fournit aucune information sémantique ou syntaxique.

[18] Softmax est une généralisation de la fonction régression logistique pour la classification multi-classes. De nombreux réseaux neuronaux multicouches se terminent par une avant-dernière couche qui produit des scores à valeur réelle (positive, négative ou nulle) qui ne sont pas bien mis à l'échelle et donc difficilement interprétables. Le softmax convertit ces scores en une distribution de probabilité normalisée entre les valeurs 0 et 1. Pour cette raison, elle est souvent utilisée comme dernière fonction d'activation dans la couche finale d'un réseau neuronal.

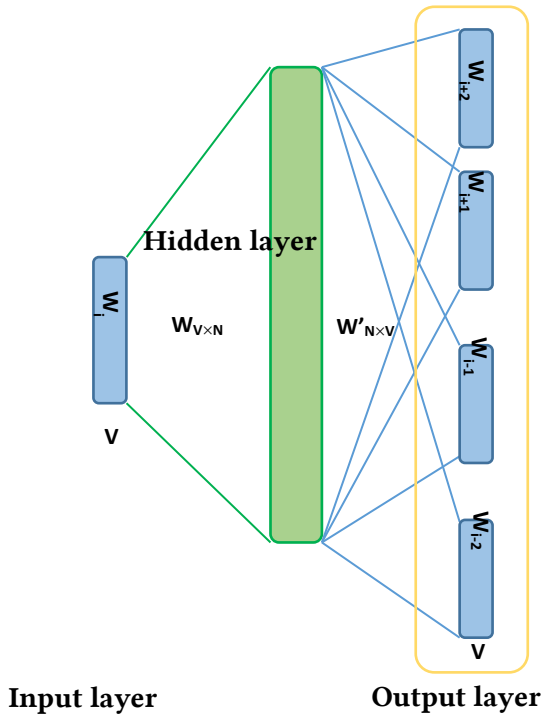

Figure 2.5. Architecture Skip-gram

Pour conclure, les performances de CBOW et Skip-gram sont en général similaires. Néanmoins chacun des deux modèles a ses propres avantages et inconvénients. Selon Mikolov, Skip-gram représente bien les mots rares, en plus, il est plus performant dans les tâches sémantiques. En revanche, CBOW est plus rapide et représente mieux les mots les plus fréquents, en outre, il est plus performant dans les tâches syntaxiques.

2.1.4.2 Glove

L'algorithme Global Vectors for Word Representation, ou GloVe, est une extension de la méthode Word2Vec dont l'objectif est d'améliorer l'apprentissage des vecteurs de mots. Glove fonctionne en combinant des techniques de factorisation matricielle telles que l'analyse sémantique latente (LSA). Ces techniques permettent d'utiliser les statistiques globales du texte (fréquence de co-occurrence des mots) avec l'apprentissage basé sur le contexte local (les mots du voisinage) tel que celui de Word2Vec.

Les modèles basés sur le contexte local s'adaptent très bien à la taille du corpus. De plus, ils ont démontré une efficacité importante dans plusieurs tâches NLP. En revanche, leur inconvénient majeur vient du fait que l'information sémantique apprise sur un mot n'est influencée que par les mots de son voisinage. Ce qui ne permet pas de profiter de la grande

quantité de répétitions (fréquence de co-occurrence) existante dans l'ensemble des données du corpus.

Plutôt que d'utiliser une fenêtre pour définir le contexte local, GloVe construit premièrement une matrice de co-occurrence pour l'ensemble du corpus. Cette matrice compte essentiellement la fréquence d'apparition d'un mot dans un contexte (le nombre de fois qu'un mot donné apparaît avec d'autres mots). Puis, puisque la taille de la matrice est importante, alors pendant son apprentissage, le modèle tente de la factoriser pour réduire sa dimensionnalité. Cette opération est effectuée tout en préservant les probabilités de co-occurrence des mots qui constitueront les vecteurs du word embedding.

Les techniques de factorisation matricielle capturent efficacement les statistiques globales du texte et leur entraînement est considérablement rapide. Toutefois, elles ne permettent pas d'incorporer facilement de nouveaux mots ou documents. En plus, elles ne sont pas adéquates lorsqu'il s'agit de mots ou documents volumineux.

2.1.4.3 FastText

FastText est un autre modèle de word embedding qui est aussi une extension du Word2Vec. Cependant, au lieu d'apprendre directement les représentations vectorielles du mot entier, FastText représente les mots sous forme de caractères n-grammes en plus du mot lui-même. Ce qui permet au modèle d'exploiter les informations des sous-mots pour construire le word embedding.

Par exemple le mot 'parabole' avec n=3 sera représenté (en considérant le début et la fin du mot) par <pa, par, ara, rab, abo, bol, ole, le>. Les crochets angulaires indiquent le début et la fin du mot pour distinguer les n-grammes d'un mot du mot lui-même. Donc par exemple, si le mot 'bol' fait partie du vocabulaire, il est représenté par <bol>. Cette technique est utile pour préserver le sens des mots plus courts qui peuvent apparaître comme n-grammes d'autres mots. En plus, cette information de sous-mots permet aux embeddings de comprendre les suffixes et les préfixes.

Pendant la phase d'entraînement, les représentations sont premièrement apprises à partir des caractères n-grammes, et les mots seront représentés par la somme des vecteurs n-grammes. Ensuite, un modèle de type Skip-gram est entraîné pour apprendre l'embedding.

FastText fonctionne bien avec les mots rares. Ainsi, même si un mot n'a pas été vu pendant l'entraînement, il peut être décomposé en n-grammes pour obtenir son embedding. Alors que Word2vec et GloVe ne fournissent aucune représentation vectorielle pour les mots qui ne figurent pas dans le vocabulaire généré par le modèle.

2.1.5 Contextual Embedding

Contextual embedding est un récent avancement dans la modélisation du langage. Les modèles précédents du word embedding fournissent une représentation unique figée pour chaque mot du vocabulaire quel que soit le contexte où il figure. Toutefois, le contextuel embedding produit pour chaque mot une représentation différente suivant le contexte de son utilisation. Plusieurs modèles ont vu le jour ces dernières années, à titre d'exemple nous citons, ELMo (Peters et al., 2018), GPT-3 (Brown et al., 2020) et BERT (Devlin et al., 2019). Ce dernier modèle reste le plus populaire aujourd'hui, il a été largement étudié et employé dans différentes tâches NLP.

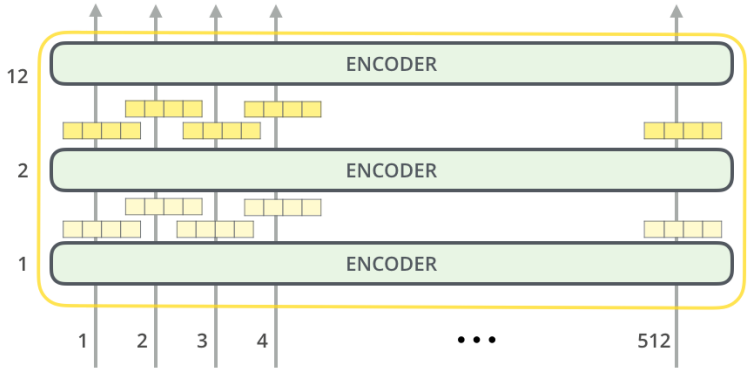

Figure 2.6 Architecture de BERT (version BERT-base avec 12 encodeurs). 512 c'est la taille maximale de la séquence d'entrée.

BERT (Bidirectional Encoder Representations from Transformers) a été entraîné sur un corpus de texte brut énorme composé de la version anglaise de Wikipedia (2.500M mots) et BooksCorpus (800M mots). L'architecture du modèle est constituée de multicouches de Transformers[19] (Vaswani et al., 2017) contenant seulement des encodeurs (Figure 2.6).

L'entraînement de ce modèle passe par deux phases, la phase du *pré-entraînement* (pre-training) et la phase de *l'adaptation à une tâche* (fine tuning). Pendant le pré-entraînement, le

[19] Un Transformer est un modèle neuronal d'architecture encoder-décoder intégrant les mécanismes d'attention pour transmettre une image plus complète de la séquence entière au décodeur en une fois plutôt que de manière séquentielle.

modèle est entraîné sur deux tâches. La première est la *Modélisation du Langage Masquée* (Masked Language Modelling (MLM)) où l'on masque aléatoirement un mot du texte en entrée. L'objectif étant alors de prédire le mot masqué. La deuxième tâche est la *Prédiction de la Phrase Suivante* (Next Sentence Prediction (NSP)) où le modèle doit décider si deux phrases en entrée sont consécutives. Pendant la deuxième phase, les représentations contextuelles générées par BERT sont utilisées comme entrée d'autres systèmes effectuant une tâche NLP spécifique.

Ce modèle a été utilisé dans plusieurs tâches NLP, par exemple celle de questions-réponses (Question Answering) ou la reconnaissance des entités nommées (Named Entity Recognition) et autres. Il a démontré une progression remarquable des performances par rapport aux modèles existants. Son succès revient en premier lieu à son encodeur bidirectionnel qui prend en considération le contexte se situant avant et après le mot. En deuxième lieu, son architecture permet le traitement parallèle des séquences d'entrée ce qui permet de gagner énormément du temps de calcul. En revanche, ce type de modèle nécessite une énorme quantité de données d'entraînement (texte brut), ce qui entraîne un long temps de calcul et nécessite un matériel très coûteux.

En résumé, Word2Vec et Glove traitent des mots entiers, et ne peuvent pas facilement traiter des mots qui n'ont pas été vus auparavant. FastText (basé sur Word2Vec) utilise des n-grammes de caractères, bien qu'il génère toujours un vecteur par mot, et il peut généralement traiter des mots non vus pendant l'apprentissage. Ces trois modèles génèrent une représentation statique (fixe) pour chaque mot sous forme d'un vecteur unique. Les différents sens du mot (s'il y en a) sont combinés en un seul vecteur pour cela ils sont indépendants du contexte (hors contexte). Toutefois BERT assure une représentation adaptée au contexte. Il génère des embeddings dynamiques qui permettent de générer plusieurs vecteurs pour le même mot, en fonction du contexte dans lequel le mot est employé.

2.1.6 Systèmes de traductions automatique (MT)

Le domaine des MTs a été largement étudié pendant de nombreuses années. Le système de traduction basé sur les règles RBMT (Rule Based Machine Translation) est le plus ancien

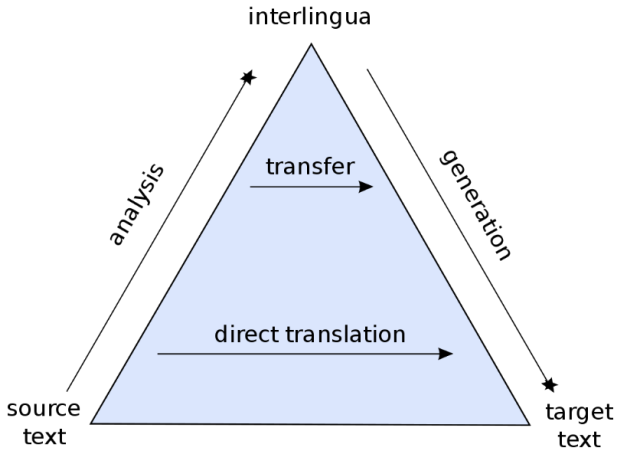

Figure 2.7. Triangle de Vauquois (source Wikipedia)

des systèmes MT (Hutchins, 1986). L'objectif de la RBMT est de **construire une traduction** d'une langue source vers une langue cible. Il existe trois types de systèmes RBMT comme c'est illustré sur la figure 3.1. Le premier type est basé sur la *traduction directe* (direct translation), elle consiste à traduire les phrases mot à mot avec une légère correction morphologique et syntaxique. Le deuxième type est basé sur le *transfert* (transfer based), cette approche applique des règles linguistiques pendant le processus de traduction. Le dernier type est basé sur *l'interlingua*, dont le principe est de faire correspondre la langue source à une représentation intermédiaire abstraite (interlingua : langue universelle) à partir de laquelle la langue cible est générée.

La RBMT repose sur l'utilisation de ressources lexicales telles que les *dictionnaires bilingues* en plus des *règles de structure*. Ces règles sont employées pour transférer la structure grammaticale du texte source dans le texte cible. En conséquence, la complexité de ce type de traducteur augmente avec le *nombre de règles,* quoique le *sens soit mieux préservé* surtout pour le type basé sur le transfert comme confirmé dans (Crego et al., 2014). Un autre avantage de cette approche est son indépendance du domaine du texte à traduire.

La traduction RBMT basée sur le transfert passe par trois étapes principales :

- Etape d'***analyse*** dont l'objectif est d'extraire les caractéristiques de la structure du texte de la langue source suivant différentes règles : règles lexicales[20],

[20] La lexicologie est l'étude des aspects des mots (origine, sens, forme) composant une langue.

morphologiques[21], syntaxiques[22] et sémantiques[23] (pour les langues source et cible). Dans les systèmes actuels de traduction, cette étape comporte également un module d'identification automatique de la langue source.

- Ensuite, vient le ***transfert*** du sens et des caractéristiques morphologiques des mots de la langue source vers la langue cible.

- Enfin, la ***génération*** de la forme de surface des mots dans la langue cible ainsi que le réordonnancement de ces mots dans leurs phrases.

La deuxième approche qui a émergé après la RBMT est la SMT (Statistical Machine Translation). L'objectif de la SMT est de **trouver la traduction la plus probable** d'une phrase source vers une autre cible. Cette correspondance entre phrase source et phrase cible se fait à l'aide de modèles statistiques (Brown et al., 1993). Le premier modèle, est le modèle de traduction, dérivé de l'analyse de corpus parallèle[24] bilingue, et le deuxième c'est le modèle de langue construit à partir de corpus monolingue. Pour obtenir une bonne qualité de traduction, la SMT a besoin de *ressources importantes* dans un domaine spécifique. Toutefois, les résultats de traduction fournis par les SMT sont *plus fluides* (Crego et al., 2014), ce qui constitue leur principal avantage devant les RBMT.

Plusieurs études se sont intéressées à la combinaison de la RBMT et de la SMT en une MT hybride afin d'acquérir les meilleures propriétés de ces deux techniques, comme l'indique cette étude (Costa-Jussà and Fonollosa, 2015). Des approches différentes ont été utilisées pour réaliser cette hybridation, qui peut être classée en hybridation guidée par la RBMT (Costa-Jussà and Centelles, 2015) ou par la MT basée sur les corpus (Antonova and Misyurev, 2014).

La NMT ou la MT neuronale est la plus récente des approches de traduction, elle a été introduite dans les travaux de (Sutskever et al., 2014) et (Cho et al., 2014). Les NMTs ont été basées principalement sur des modèles encodeur-décodeur de séquence à séquence de type RNN. Puis, elles ont été améliorées par l'ajout du mécanisme d'attention et d'autres techniques pour devenir les traducteurs d'état d'art (Koehn, 2020). Par rapport aux modèles

[21] La morphologie est l'étude de la formation des mots et de leurs variations.
[22] La syntaxe est l'étude de la construction de la phrase ou bien des règles de combinaison des éléments de la phrase.
[23] La sémantique est l'étude du sens de la phrase.
[24] Un corpus parallèle est un ensemble de couples de textes tel que, pour un couple, un des textes est la traduction de l'autre. Ces textes peuvent être alignés au niveau des paragraphes, phrases ou même des mots.

précédents, où la traduction passe à travers un pipeline de tâches séparées, les NMTs sont capables de transformer une phrase source en une phrase cible avec un seul large réseau. Cette approche a réalisé les meilleures performances de traduction pour plusieurs paires de langues (Stahlberg, 2020). Cependant, elle nécessite des *corpus parallèles volumineux* et souffre du problème des *mots rares* (Koehn, 2020).

2.2 Etat de l'Art

Afin de procéder à la détection du contenu violent dans les textes générés par les utilisateurs des médias sociaux, il va falloir passer par deux étapes principales. La première consiste en la préparation de ce texte de nature bruitée, à travers une chaine de traitements visant sa normalisation ou sa standardisation. L'objectif de la normalisation est d'uniformiser les écrits pour faciliter la tâche à la deuxième étape, celle-ci vise l'utilisation des techniques d'apprentissage automatique pour détecter l'existence des nuances de violence dans le texte. De nombreux travaux se sont intéressés d'une part, par la normalisation du texte bruité, qui constitue encore un défi pour les chercheurs en NLP, surtout lorsqu'il s'agit du phénomène de mélange de langues appelé Code Switching (CS) et du dialecte. Et d'autre part, par la détection des différentes formes de la cyberviolence. Dans ce qui suit, nous abordons les différents travaux connexes à ces deux dites étapes, la normalisation et la détection. Mais tout d'abord, nous présentons les différentes théories linguistiques relatives au CS sur lesquelles s'appuie notre approche de normalisation du CS.

2.2.1 Code Switching (Alternance Codique) en linguistique

Dans cette section, nous essayons de discuter les principales approches proposées pour la modélisation du CS. L'objectif vise à chercher un modèle général pour la représentation du CS afin de déterminer la meilleure direction de traduction, qui assure le transfert du sens. En d'autres termes, quelles langues allons-nous considérer comme source de la traduction et vers quelle langue cible allons-nous traduire ?

Ces approches ont été d'abord étudiées pour la parole. Toutefois, aujourd'hui la majorité des auteurs des réseaux sociaux utilisent le style de la parole à l'écrit. Donc, ces approches restent valables et peuvent être appliquées pour ce style de texte.

L'utilisation du CS est très répandue dans les communautés multilingues (les personnes capables de parler et d'écrire dans deux langues ou plus). L'alternance codique a été proposée pour la première fois par (Haugen, 1950) et elle est définie comme l'utilisation alternative de plus d'un code linguistique. Le CS peut être divisé structurellement en deux catégories principales (Poplack, 1980) :

- La première c'est l'*inter-sentenciel* qui se produit en dehors de la phrase, comme dans cette phrase. : « Tu vas nous créer des grands problèmes (Français). Daba tchouf chno ghadi yaw9a3 (dialecte Arabe Marocain). ».
- La deuxième catégorie est l'*intra-sentenciel qui* se produit à l'intérieur de la limite de la phrase, comme dans « L3az khouti (Arabe Marocain) nous irons tous au match (Français) let's go (Anglais) daba! (Arabe Marocain) »

Dans ce travail, nous nous intéressons au CS intra-sentenciel qui, vu sa complexité, constitue un véritable défi à la traduction. Selon (Hamers and Blanc, 1983), ce type de CS peut être représenté par une séquence de langues ou de codes alternés LX et LY, dans le cas de deux langues, comme dans l'exemple suivant:

CS = LX/LY/LX/LY ou **CS = LX/LX/LY/LX**.

Plusieurs études ont tenté de modéliser la grammaire du CS et les approches les plus connues sont les suivantes :

Modèle linéaire : parfois appelé le modèle de **Poplack** (Poplack, 1980), puisque ses études ont été les plus influentes dans cette approche. Dans ce modèle, la phrase CS est constituée de fragments alternés de différentes langues selon deux contraintes syntaxiques, la contrainte d'***équivalence***[25] et la ***contrainte de morphème libre***[26]**.** Cependant, ce modèle, soulignant l'importance de l'équivalence syntaxique des deux codes et l'ordre des

[25] Selon Polack, la contrainte d'équivalence signifie que la juxtaposition des constituants alternés est permise si elle ne viole pas les règles syntaxiques des deux langues respectives. Chaque segment doit se soumettre à la grammaire de la langue dans laquelle il est formulé (Ziamari, 2003).

[26] La contrainte du morphème libre interdit que les codes soient alternés après un morphème lié et seuls les morphèmes libres ont cette possibilité. Par exemple, dans l'énoncé '*partagite'* le morphème lié *'ite'* de l'Arabe Marocain a été préfixé par la racine du verbe Français *'partager'*. Ce cas est considéré un emprunt et non pas du CS. Par contre dans l'énoncé '*oustadna calme'* est accepté comme CS puisque le mot Français calme est un morphème libre.

syntagmes[27], ne tient pas devant de nombreuses formes de phrases CS, qui restent injustifiables par ses contraintes.

Modèle de gouvernement : ce modèle (Muysken, 1995) est dérivé de la théorie "X Bar" de Chomsky, (1970) et essaie de trouver une explication universelle de ce phénomène. Il accorde une importance considérable à la structure interne des constituants des phrases CS. En particulier, à la relation entre l'élément lexical et la syntaxe qui l'entoure, connue sous le nom de ***Contrainte du gouvernement***[28]. Cette contrainte a été améliorée par son auteur à plusieurs reprises, pourtant le modèle reste toujours incapable de lever l'ambiguïté relative au CS.

Les deux modèles susdits, reposant sur l'équivalence syntaxique des deux codes, ne sont pas en mesure d'expliquer tous les cas de CS ou à la rigueur la plupart d'entre eux. Donc ils souffrent de l'incapacité à la généralisation. Cependant, nous recherchons une approche générique pour les différentes occurrences de ce phénomène linguistique. C'est pourquoi nous présenterons la troisième et la plus générale approche de modélisation du CS qui est le *modèle Insertional.*

Modèle Insertional : le modèle dominant dans cette approche est le ***Matrix Language-Frame*** (Myers-Scotton, 1995), introduit à l'origine par Joshi (Joshi, 1985). Cette approche est dérivée des études neuro-linguistiques et psycholinguistiques. Ce modèle distingue les rôles des langues participantes dans les phrases CS. Il considère que la distribution des deux codes (langues) est asymétrique c'est à dire que les contributions des différents codes ne sont pas équivalentes. En outre, ce modèle affirme que le CS est une combinaison syntactico-sémantique de codes. Ces codes sont la *Matrix Language*, qui est la langue dominante ou la plus fréquente (appelée langue d'accueil dans d'autres littératures), et l'autre code est la *Embedded Language,* représentée par les termes étrangers insérés. Scotton définit la *Matrix Language* (ML) comme : "le ML est la langue qui fournit relativement plus de morphèmes pour le type d'interaction concerné que la ou les autres langues dans la même conversation" (Myers-Scotton, 1995).

[27] Syntagme : mot ou groupe de mots.
[28] La contrainte du gouvernement prend en compte la structure hiérarchique des langues. Selon cette contrainte, c'est la relation de gouvernance entre éléments syntaxiques qui permet ou non d'alterner entre codes. Autrement dit, elle impose un contrôle sur les classes (verbe, adverbe, particules...) des constituants d'un énoncé CS pour le considérer grammaticalement correcte.

Scotton a étendu son modèle avec les modèles 4-M et Abstract Level (Myers-Scotton and Jake, 2001) en incluant la production linguistique et les compétences linguistiques du locuteur comme facteurs d'influence du CS. Ces modèles indiquent que l'harmonie entre les codes utilisés, connue sous le nom de *congruence*, est liée aux trois niveaux linguistiques de la production du CS : les niveaux, *conceptuel*, *fonctionnel,* et *positionnel*. Ces niveaux doivent être partagés par les codes utilisés pour assurer leur correspondance sémantique.

En résumé, tous ces modèles tendent à généraliser la structure du CS et à souligner l'idée que le CS est basé sur des **processus cognitifs abstraits,** ce qui nécessite l'investigation des connaissances linguistiques qui sous-tendent le CS.

Quant à l'identification de la Matrix Language, cette dernière est liée à différents paramètres. Dans le modèle de base du MLF proposé par Scotton, l'identification de la ML repose sur le critère de la fréquence, ce qui signifie que la ML est la langue qui fournit le plus de morphèmes dans la phrase CS. Alors que le linguiste Bauman a proposé de définir la ML comme la langue du verbe conjugué : « the Matrix language (ML) on sentence level is the language of the inflexion bearing element of the tensed verb » (Boumans, 1998). Auparavant, Joshi avait identifié le ML comme la langue des premiers mots de la phrase CS. Finalement, dans ses derniers modèles, Scotton affirme que la ML dépend des compétences linguistiques du locuteur et qu'elle peut être définie comme la première langue du locuteur.

Jusqu'au aujourd'hui, la définition de la ML n'est pas encore précise chez les linguistes. En général, elle est considérée comme la langue la plus fréquente dans une phrase ou elle peut être la première langue de l'auteur. Si nous supposons que la ML est la première langue de l'auteur, dans ce cas nous aurons deux scénarios. Le premier est que cette langue peut être définie comme la langue maternelle. Cependant, nous pouvons observer que dans certaines communautés multilingues, la langue maternelle est rarement utilisée par rapport aux langues étrangères (par exemple chez les immigrés) (Veltman, 1988). Il est donc difficile de généraliser cette hypothèse. Le deuxième scénario est que la première langue peut être la langue la plus maîtrisée par l'auteur. Cependant, l'identification de cette langue nécessite une connaissance approfondie du profil web de l'auteur pour définir ses compétences linguistiques (afin de construire les processus cognitifs abstraits). Ce qui demande une grande connaissance des théories psycholinguistiques et sociolinguistiques, difficile à atteindre dans notre situation.

En résultat, selon le modèle Matrix Language Frame, la langue matrice (Matrix Language) conduit la sémantique de la phrase CS et contribue fondamentalement à sa construction de sens. Par conséquent, dans ce travail, nous considérons comme langue cible de notre MT, la langue matrice que nous définissons par la langue dominante ou la plus fréquente. Tandis que, les autres langues présentes dans la phrase CS, seront les langues sources.

Finalement, pour convertir une phrase CS d'une phrase multilingue en une autre monolingue tout en préservant sa sémantique, nous allons d'abord identifier les différentes langues composant cette phrase, en déduire ensuite celle dominante, et enfin convertir la totalité des morphèmes de la phrase source à la langue dominante.

2.2.2 Normalisation du Code Switching

Dans ce travail, nous nous intéressons à la normalisation des textes des SM. À cette fin, nous adoptons une approche de type MT (traduction automatique) pour transformer le texte non standard généré par les utilisateurs en une forme standard, ce qui va permettre ainsi son analyse par les techniques de Text Mining. Par conséquent, nous devons en premier lieu traiter les irrégularités du langage, et en particulier le phénomène de CS et l'utilisation du dialecte. Pour cela, la traduction des textes CS est parmi les tâches principales de notre recherche. Cette traduction sera de plusieurs langues vers une seule langue qui est la Matrix Language tout en préservant la sémantique de la phrase à travers des techniques spécifiques. Dans cette section, nous présentons un aperçu de la littérature sur les travaux connexes, qui s'intéressent à l'*identification de langue*, à la *normalisation de type MT,* à la *traduction du CS* et à la *traduction sémantique.*

2.2.2.1 Identification de langue standard et dialecte

La LID (Language Identification) est essentielle pour les systèmes MT, parce qu'elle permet de déterminer la langue source avant de procéder à la traduction. Pour le texte standard monolingue, la LID est considérée comme un problème résolu avec une précision proche de 100% (McNamee, 2005). Dans ce cas, les systèmes LID peuvent fonctionner au niveau phrase ou document à travers l'utilisation de plusieurs techniques comme par exemple, la fréquence des mots communs[29], les modèles n-grammes[30] et la classification

[29] Mots communs : conjonctions, prépositions, déterminants...
[30] N-grams : séquence de caractères ou de mots

supervisée. Cependant, pour les textes des SM, l'identification de la langue est loin d'être une tâche résolue (Baldwin, 2017), en raison de la taille réduite des textes et l'usage non standard des langues.

Vu la nature des phrases CS composées de mots multilingues, il est donc nécessaire d'identifier la langue au niveau token[31]. Ce qui rend cette tâche encore plus complexe. Dans la littérature, certains travaux se sont intéressés à ce sujet. Les approches les plus utilisées sont classées en deux catégories. La première est basée sur le corpus. Tandis que la deuxième catégorie est une approche hybride où des ressources de connaissances et des corpus sont combinés.

Dans l'approche basée sur le corpus, nous distinguons entre les techniques non supervisées et celle supervisées. Pour la technique non supervisée, la majorité des travaux reposent sur les techniques de modélisation thématique ou Topic Modeling en particulier la technique Latent Dirichlet Allocation[32] (LDA) (Blei et al., 2003). Dans (Voss et al., 2014), ils ont considéré les langues comme des thématiques et ont employé la LDA pour regrouper les tweets suivant ces langues. Quant au (Zhang et al., 2016) la LDA a été utilisée pour filtrer ou identifier les langues primaires parmi les autres langues présentes dans un corpus. Ces techniques restent valables en cas de CS inter-sentenciel puisqu'elles repèrent les langues au niveau phrase.

Concernant les techniques supervisées qui reposent sur des corpus annotés, dans (Samih et al., 2016), les auteurs ont utilisé la représentation des caractères et des mots avec un modèle RNN et le word embedding. Dans un autre travail (Jurgens et al., 2017), un modèle RNN de séquence à séquence basé sur les caractères a été utilisé pour détecter les variétés de dialectes et langues dans le CS. Encore un modèle de segmentation RNN (Mager et al., 2019) a été employé pour l'identification des sous-mots dans le CS intra-mots, ce modèle est approprié aux langues morphologiquement riches.

En ce qui concerne l'approche hybride, dans (Elfardy and Diab, 2012) un analyseur morphologique a été utilisé pour l'identification de l'Arabe standard moderne (MSA) et un

[31] Token: unité lexicale

[32] LDA : est un modèle probabiliste génératif qui permet de regrouper les mots d'un document par thème. La LDA assume que chaque document contient une combinaison de thèmes. Ces thèmes sont des clusters de mots connexes. Puisque les mots d'un document sont connus, les thèmes peuvent être estimés à travers un échantillonnage.

dictionnaire pour l'identification des dialectes. En outre, ils ont utilisé les règles de changement de son, pour explorer les variantes phonologiques possibles et enfin des modèles de langue n-grammes. Également, dans (Nguyen and Do, 2013), un mélange de techniques a été utilisé, y compris la recherche dans les dictionnaires, un modèle linguistique, et un modèle de régression logistique et CRF avec et sans contexte. Dans (Das and Gambäck, 2014), les auteurs ont appliqué plusieurs méthodes pour la LID qui reposent respectivement sur un modèle de caractères n-grammes, des dictionnaires et un classifieur SVM.

Notons que pour plus de détails sur les travaux liés à la LID, l'enquête (Jauhiainen et al., 2019) est très instructive.

En somme, pour appliquer la LID en cas de CS nous avons deux possibilités : si nous disposons d'un corpus annoté, nous pouvons adopter les techniques d'apprentissage supervisées. Dans le cas contraire, il faut opter pour une approche hybride.

2.2.2.2 Normalisation type MT ou MT-like

L'approche de normalisation de type MT (Saloot et al., 2015) vise à utiliser des techniques de la traduction automatique pour normaliser un texte contenant des formes d'utilisation du langage non standard. Elle a été adoptée par (Kaufmann and Kalita, 2010) qui ont utilisé l'approche de type SMT (Statistical MT) pour normaliser les tweets Anglais. Également, dans (Lopez Ludeña et al., 2012), un système de SMT a été utilisé pour convertir des textes SMS contenant des mots non standard vers une forme standard.

La CSMT (Character level SMT) étant une SMT fonctionnant au niveau caractères, a également été utilisée dans la normalisation et ses performances ont dépassé celles de la SMT au niveau mots. Dans (Simov et al., 2016), une CSMT entraînée sur des paires de segments de mots a été utilisée pour normaliser le corpus dialectal Suisse-Allemand.

Quant aux méthodes neuronales, elles n'ont pas été largement utilisées dans la normalisation des textes (même si elles sont beaucoup plus performantes en MT). Parce qu'elles nécessitent de grands ensembles d'entraînement, qui ne sont pas disponibles pour cette tâche. Pour remédier à cette limitation, (Lusetti et al., 2018) ont utilisé un encodeur-décodeur neuronal avec deux modèles de langage, le premier au niveau mots et le deuxième au niveau caractères. Ce qui leur a permis de surpasser les performances de la CSMT.

Dans ces études présentées, nous pouvons constater que la normalisation vise à la standardisation des tokens bruyants contenus dans des textes monolingues. Cependant, dans notre cas, nous nous intéressons à la normalisation d'un type de texte bruyant plus compliqué, à savoir un texte contenant des phrases CS, afin d'obtenir une forme de texte standard convenable au traitement de l'analyse. Ainsi, en plus de la normalisation des éléments bruyants, nous devons convertir le texte CS en un texte monolingue. Nous présentons ci-après des études qui s'intéressent à la traduction CS.

2.2.2.3 La traduction sémantique du Code Switching

a. Traduction du Code Switching

En général, les travaux connexes sur le CS sont limités. Nous pensons que cela est dû à deux raisons. La première est liée à la complexité de cette tâche, comme il est confirmé par (Çetinoğlu et al., 2016). La deuxième raison revient au fait que le CS a été étudié en grande partie dans le traitement de la parole, vu son utilisation massive dans les communications orales, en particulier chez les communautés multilingues. Alors que l'emploi du CS en texte écrit n'a pas été répandue dans le passé. Il a commencé avec les SMS, et est devenu très populaire après l'expansion de la libre expression dans les SM. Par conséquent, le problème de la traduction du CS n'a pas été largement couvert dans la littérature. A travers les paragraphes suivants, nous présenterons les travaux relatifs à la traduction du CS.

Dans l'étude faite par (Fung et al., 1999) et développée dans (Cheung and Fung, 2005), les auteurs ont décrit une méthodologie de traduction et de désambiguïsation des mots. Il s'agit d'une sélection lexicale pour les requêtes en langue mixte Anglais-Chinois en passant par trois étapes principales. La première étape consiste à générer des candidats à la traduction de mots appartenant à la langue secondaire vers la langue principale (la langue la plus fréquente) à partir d'un dictionnaire bilingue. La deuxième est la pondération de ces candidats en calculant les scores de l'information mutuelle de la fréquence de cooccurrence du mot cible et ses mots contextuels. Ces scores ont été calculés à l'aide d'un corpus monolingue de la langue principale. La dernière étape est la sélection de la traduction ayant le meilleur score d'information mutuelle. Bien que cette technique ne nécessite pas de corpus parallèles, pourtant, elle possède deux inconvénients. D'une part, l'information de cooccurrence de deux mots ne tient pas compte de la distance entre eux. Ce qui influence la précision de la désambiguïsation, comme le rapporte une autre étude (Gao et al., 2002). En outre, un aspect

important a été omis dans cette solution, c'est qu'elle ne considère que les lemmes des mots de contenu et ne tient pas compte des différentes formes flexionnelles, comme la forme conjuguée des verbes ou le pluriel des noms. Alors, pour élargir l'utilisation de cette solution, il est nécessaire d'employer une analyse morphologique pour extraire les lemmes des formes fléchies avant la traduction. Par exemple, pour les verbes conjugués, nous devons extraire leurs formes infinitives, et pour les noms au pluriel, leurs formes singulières, autrement, ces mots ne seront pas traduits.

Dans son travail, (Sinha and Thakur, 2005) a proposé une solution pour traduire un texte hinglish (CS entre Hindi et Anglais) en convertissant l'ensemble du texte en Hindi pur puis en Anglais pur. D'une part, l'approche adoptée est basée sur des règles linguistiques qui rendent cette solution dépendante de la langue. D'autre part, elle ne tient pas compte de la sémantique au niveau mot, puisque la traduction se fait sans tenir compte du contexte du mot.

Dans (Manandise and Gdaniec, 2011), les auteurs ont présenté une technique permettant de traduire des termes de langues mixtes et d'emprunt dans des phrases hispano-anglaises par l'utilisation d'une analyse morphologique. Ils ont constaté que les mots échangés sont surtout considérés comme des noms lors de l'analyse syntaxique, ce qui dégrade la performance de la traduction. De plus, l'analyse lourde des règles de structure rend la technique dépendante de la langue.

Dans leur étude (Van Gompel and Van Den Bosch, 2014), ils ont montré une méthodologie pour traduire des textes mixtes en utilisant la SMT (Statistical Machine Translation). La traduction a été faite d'une phrase ou d'un mot dans une langue maternelle L1 vers une phrase ou un mot dans une langue étrangère L2 en se basant sur un contexte L2. Pour la désambiguïsation de la traduction, un classifieur entraîné est utilisé pour faire correspondre un mot ou une phrase de L1 dans son contexte de L2 à leurs traductions appropriées en L2. Ils ont évalué leur approche de classification sur plusieurs paires de langues telles que l'Anglais-Espagnol, l'Anglais-Français et autres. La précision rapportée a été considérablement améliorée par rapport à la référence de base. Cette technique nécessite un corpus parallèle pour l'entraînement des classifieurs. Par conséquent, elle convient bien aux langues riches (ressources disponibles). En revanche elle ne peut être appliquée aux langues et dialectes faiblement pourvus en ressources.

Et récemment, (Dhar, 2018) a créé et augmenté un corpus parallèle existant liant entre le CS Hindi-Anglais et l'Anglais monolingue. Ce corpus a été ensuite injecté dans certains traducteurs existants pour améliorer leur traduction du CS Hindi-Anglais vers l'Anglais. La meilleure performance a été atteinte par Google Neural MT. Cette solution était basée sur la traduction de la plus large séquence de mots appartenant à la langue intégrée (embedded language). Toutefois, ces séquences ne sont qu'un cas de CS. En réalité, dans certaines phrases CS, nous pouvons trouver des mots isolés de la langue intégrée, entourés par des mots ML (langue dominante). Dans ce cas-là, nous aurons besoin d'autres techniques de désambiguïsation qui opèrent au niveau mot.

En conclusion, tout d'abord nous pouvons observer que toutes ces études se sont concentrées sur les textes et les langues standards. Ils n'ont pas abordé le cas des textes bruyants des SM, ni les dialectes, ni les langues moins dotées de ressources. Deuxièmement, la plupart des approches, soit pour la traduction soit pour la désambiguïsation, étaient dépendantes de la langue ou basées sur des corpus. Ce qui limite également leur application étendue pour tous les types de langues. Enfin, ils n'ont pas abordé la question de l'identification automatique des langues source et cible, à l'exception d'une tentative limitée dans le dernier travail cité. Alors que nous le considérons comme un traitement clé pour une traduction sémantique réussie

Par conséquent, dans cette étude, nous introduisons une solution pour traduire le CS en texte SM. Cette solution inclut l'identification automatique des langues sources (intégrées) et de la langue cible (dominante), sans être dépendante des langues impliquées.

Dans ce contexte, l'aspect le plus précieux de notre solution est la sémantique. Ceci dit, il est obligatoire de préserver le sens du mot dans son contexte pendant la traduction. La seule solution ici est d'inclure une étape de WSD (word sens disambiguation) dans notre traitement. Pour souligner l'intérêt de cette tâche et montrer les techniques employées, nous citons ci-après quelques études intéressées par la MT utilisant la WSD.

b. Désambiguïsation du sens des mots (WSD)

La polysémie est un phénomène typique qui touche les mots dans les différents langages naturels. Les mots polysémiques sont des mots qui possèdent plusieurs sens selon leur contexte d'utilisation. Ce phénomène pose des problèmes devant les tâches NLP qui

nécessitent de savoir le sens exact des mots et donc des phrases. La désambiguïsation du sens des mots (WSD) est la tâche qui consiste à associer le sens correct d'un mot dans un contexte donné (ensemble de mots qui entourent ce mot). Cette tâche est utilisée dans plusieurs traitements du langage naturel car elle n'est pas une fin en soi (Ide and Véronis, 1998). Par exemple, dans les systèmes de traduction automatique, le rôle de la WSD est fondamental pour garantir la qualité de la traduction. Elle est aussi présente dans les applications de la recherche d'information et la reconnaissance de la parole parmi d'autres. Plusieurs approches ont été développées jusqu'à présent que nous pouvons regrouper en deux catégories : les approches basées sur les connaissances, et celles basées sur les corpus.

Les approches WSD basées sur les connaissances *(knowledge based)*

Les méthodes fondées sur les connaissances exploitent les relations entre les mots qui sont fournies par des ressources lexicales externes (par exemple : dictionnaires, ontologies, thésaurus). En général, ces méthodes peuvent être classées en deux grandes catégories, à savoir celles fondées sur les *similarités* et celles fondées sur les *graphes*, aussi connues par *l'approche structurelle* (Navigli, 2009) :

- *Approche de similarité sémantique*

Cette première catégorie utilise les dictionnaires comme ressource, elle compare chaque sens d'un mot cible, fourni par un dictionnaire, avec les mots de contexte qui l'entourent. Le sens qui présente la plus grande similarité est supposé être le plus correct. Dans ces approches, les sens corrects sont déterminés individuellement pour chaque mot sans tenir compte des sens précédemment attribués. La mesure de similarité la plus connue dans ce type est celle de *Lesk* (Lesk, 1986) qui considère la similarité entre deux sens comme le nombre de mots en commun de leurs définitions au sein d'un dictionnaire. Et la mesure *Lesk étendu* (Extended gloss overlaps) introduite par Banerjee et Pedersen (Banerjee and Pedersen, 2003), qui élargit les glosses des mots comparés pour inclure aussi des définitions des sens reliés par des relations taxonomiques dans WordNet (Miller et al., 1990) : par exemple, hyperonymes (has-kind), hyponymes (kind-of), meronymes (part-of), holonymes (has-part), troponymes et aussi par les relations *attribute*, *similar-to*, *also-see*.

- *Approche structurelle*

Dans les méthodes structurelles dites aussi à base de graphes (Mihalcea et al., 2004;

Navigli and Lapata, 2010), un graphe est construit à partir de ressources lexicales. Il est composé de nœuds qui représentent les sens des mots et des arcs qui représentent les relations ou les interdépendances significatives entre eux. Cette structure de graphe est établie pour déterminer l'importance de chaque nœud et le sens correct correspond au nœud le plus important pour chaque mot. On trouve deux principales mesures de similarité dans cette approche. La première est à base de distance taxonomique (Rada et al., 1989) dont le principe est de compter le nombre d'arcs qui séparent deux sens dans une taxonomie. Pour une paire de mots se trouvant dans le même contexte, le sens ayant la plus courte distance entre ces mots est le plus approprié. La deuxième mesure se base sur le contenu informationnel (*information content*) partagé par les mots dans le contexte. Cette mesure introduite par (Resnik, 1995) détermine la spécificité du concept qui englobe les mots dans la taxonomie WordNet. Elle repose sur l'idée que, plus le concept commun de deux ou plusieurs mots est spécifique, plus ils sont supposés être sémantiquement liés. Le même concept a été proposé par (Jiang and Conrath, 1997; Leacock and Chodorow, 1998; Lin, 1998) ou encore Personalized PageRank (Page et al., 1999) pour le WSD (Agirre and Soroa, 2009) et Random Walk on Graph (Agirre et al., 2014), et plus récemment (Wang et al., 2019). Le modèle le plus performant est SyntagRank (Scozzafava et al., 2020) qui applique l'algorithme Personalized PageRank sur plusieurs bases de connaissances lexicales.

Les approches WSD basées sur les corpus

Ces approches nécessitent des corpus annotés ou non, elles sont de trois types, supervisée, non-supervisée et semi-supervisée :

- *Approche supervisée*

Ces approches sont les plus performantes dans cette tâche (Bevilacqua et al., 2021). Les méthodes supervisées sont des techniques d'apprentissage automatique (Florian et al., 2002) basées sur des corpus annotés manuellement relativement au sens. L'approche supervisée se déroule en deux phases, à savoir l'entraînement et la classification. La phase d'entraînement nécessite un corpus construit de mots polysémiques et des phrases contenant ces mots, ces derniers sont annotés par leurs différents sens, suivant leurs contextes d'utilisation, donc un dataset de la forme : (mot polysémique, contexte, sens). L'objectif de cette phase est de construire un modèle de classification par entraînement sur la base de ces exemples de phrases à l'aide des techniques d'apprentissage automatique. Pendant la phase de

classification, ce modèle tente de reconnaître les sens requis en fonction des mots voisins. Plusieurs techniques ont été utilisées dans cette approche, nous citons par exemple : Naïve bayes (Le and Shimazu, 2004), Support Vector Machines (Zhong and Ng, 2010) et les réseaux de neurones (Luo et al., 2018; Pasini and Navigli, 2020). Le système supervisé le plus performant est ESCHER (Barba et al., 2021) qui utilise BERT en combinant des données annotées avec des ressources lexicographiques. Bien que ces techniques soient indépendantes de la langue utilisée, leur inconvénient majeur reste la difficulté liée à la construction manuelle des corpus annotés.

– *Approche non supervisée*

L'objectif de cette méthode est de surmonter le problème de l'annotation des corpus contenant les mots avec leurs sens. L'idée provenant de cette approche dit que les mots qui partagent le même sens sont souvent entourés par les mêmes mots voisins (Yarowsky, 1993). Ces méthodes sont en mesure d'induire le sens des mots à partir du texte en regroupant les occurrences des mots ou des contextes dans des clusters (chaque cluster correspond à un sens d'un mot cible). Les clusters ont été construits à travers des algorithmes de clustering en fonction de la similarité sémantique entre mots basée sur des caractéristiques morphologiques particulières, ou la fréquence des cooccurrences parmi d'autres (Pedersen, 2007). Cette technique de clustering ne s'appuie pas sur des textes annotés pour l'entraînement et n'utilise ni dictionnaires ni thésaurus ou ontologies, ce qui constitue son avantage majeur. Toutefois, puisque les systèmes totalement non supervisés n'exploitent aucun dictionnaire, ils ne peuvent pas s'appuyer sur une référence commune des sens. Donc, l'évaluation des clusters retrouvés ne peut pas être automatique sans intervention humaine (Navigli, 2009), ce qui constitue leur principal inconvénient.

Pour pallier les limitations du clustering classique, d'autres solutions ont été construites sur la base des techniques du word embedding combiné avec des bases de connaissances. A titre d'exemple, des graphs comme WordNet (Chen et al., 2014), ou l'étiquetage des clusters par les hypernyms et des images (Panchenko et al., 2017). Récemment, le contextuel embedding a été introduit afin d'incorporer plus de caractéristiques contextuelles dans le sens embedding. Dans (Loureiro and Jorge, 2019) le modèle BERT a été utilisé avec le corpus SemCor (Miller et al., 1993) et WordNet. Cependant, ni les bases de connaissances ni les modèles de langues performants ne sont disponibles pour toutes les langues.

– *Approche semi-supervisée*

L'apprentissage semi-supervisé se trouve sur les frontières entre la désambiguïsation supervisée et celle non supervisée. Le concept de base de la méthode semi-supervisée consiste à annoter automatiquement des exemples non labellisés en utilisant un petit nombre d'exemples labellisés manuellement. Ce qui permet de produire un grand ensemble de données annotées qui peuvent être utilisées comme données d'entraînement pour un algorithme classique d'apprentissage supervisé. Nous distinguons deux approches WSD dans ce type. La première est basée sur le *bootstrapping* (Yarowsky, 1995) où un corpus est créé automatiquement à partir d'un petit nombre d'exemples annotés manuellement. Le corpus initial sera étendu suivant un processus d'entraînement itératif : un classifieur est entraîné sur les exemples annotés, puis utilisé pour labéliser les phrases d'un corpus non annoté. Le classifieur sera de nouveau re-entrainer sur la combinaison des deux corpus. Bien que cette technique requiert un petit corpus annoté, toutefois, le processus itératif risque de réduire la précision pendant les étapes suivantes. Alors, pour améliorer les performances de la désambiguation, d'autres approche ont été proposées comme (Yatabe and Sasaki, 2020; Yuan et al., 2016) qui utilisent des graphes pour étendre l'annotation des sens.

La seconde approche est basée sur l'utilisation des *proches monosémiques* proposée par (Leacock and Chodorow, 1998) et utilisée dans (Agirre and Martinez, 2004; Przybyła, 2017). L'idée repose sur la construction d'un corpus annoté, d'abord par l'extraction des proches monosémiques à partir d'une base de connaissance (souvent WordNet). Les proches monosémiques sont des mots synonymes du mot polysémique (mot cible) ayant un sens unique et donc n'ont pas besoin de désambiguation. Cette opération est suivie par l'extraction des exemples incluant ces synonymes à partir d'un corpus énorme comme le web. Cela permet de créer une collection d'exemples d'entraînement pour les sens des mots polysémiques. Les inconvénients majeurs de cette technique sont : le manque du proches monosémiques pour certains sens du mot cible, en plus du corpus qui doit être assez volumineux pour couvrir les différents sens.

Pour conclure, généralement, les approches non supervisées utilisent uniquement des connaissances lexicales, tandis que les approches supervisées utilisent des corpus annotés pour la WSD. En examinant la littérature, nous pouvons toutefois constater la tendance à combiner les connaissances lexicales et les corpus annotés dans les modèles de WSD

récemment développés (Bevilacqua et al., 2021). En termes de performance, les approches supervisées de la WSD sont les supérieures, parce qu'elles peuvent apprendre la correspondance entre certaines caractéristiques et le sens cible à partir d'un corpus annoté. Ces corpus doivent être de taille relativement importante afin d'assurer ces résultats. Cependant le manque de tels corpus coûteux, particulièrement pour les langues faiblement dotées de ressources, constitue leur principal inconvénient. D'autre part, les approches basées sur les connaissances ont permis de réaliser la WSD en combinant des informations contextuelles et des connaissances sémantiques indépendamment des corpus, en réalisant ainsi une large couverture des mots polysémiques. Ce qui a motivé leur développement rapide ces dernières années. Par conséquent, l'écart de performance entre les deux types d'approches a été réduit (Wang et al., 2019). De plus, les systèmes supervisés les plus récents tendent à incorporer les ressources lexicales pour améliorer leurs performances (Bevilacqua et al., 2021). Cependant, l'approche basée sur les connaissances est plus convenable que l'approche supervisée pour les systèmes pratiques de WSD (Dongsuk et al., 2018).

Dans cette section, nous avons passé en revue les principales méthodes de désambiguïsation du sens des mots. Pour plus de détails sur la WSD et son histoire, les revues de littérature de (Ide and Véronis, 1998), (Navigli, 2009) et (Bevilacqua et al., 2021) sont très instructives.

c. Traduction sémantique à l'aide de la WSD

Depuis les débuts des systèmes de traduction jusqu'à aujourd'hui, leur défi primordial reste la conservation du sens sémantique des mots pendant la phase de transfert. Dostert, (1959) a affirmé que " la traduction est le transfert du sens d'une langue à une autre ", en plus Li, (2015) a souligné que " la sémantique est la solution aux problèmes fondamentaux " des systèmes de traduction. Afin d'assurer ce transfert de sens, l'intégration des techniques de WSD dans le processus de traduction a fait l'objet de plusieurs études en MT.

En SMT, la tâche de la WSD a été considérée comme une tâche de traduction de mots (Vickrey et al., 2005). Dans ce travail, les auteurs ont utilisé un corpus parallèle[33] pour entrainer un classifieur et incorporer ses résultats dans un modèle de traduction. Ils ont montré que la WSD peut améliorer la qualité de la MT. De même, (Carpuat and Wu, 2007;

[33] Les corpus parallèles sont également appelés corpus bilingues, composés de deux textes, l'un servant de langue source, et l'autre comme langue cible.

Chan et al., 2007) ont prouvé que la WSD ou la sémantique lexicale peut effectivement améliorer la précision de la SMT. Dans une autre étude (Apidianaki et al., 2012), les auteurs ont intégré les résultats d'un classifieur WSD pour la sélection des n meilleurs candidats de la traduction et la modélisation locale de la langue. Les résultats obtenus montrent que la WSD peut aussi améliorer la traduction des mots de contexte.

La WSD a également été étudiée pour la MT basée sur les règles (RBMT), dans (Tyers et al., 2012) les auteurs ont décrit un module de sélection lexicale basé sur des transducteurs à états finis, entraînés sur des règles de sélection lexicale à partir de corpus parallèles. Ce module a été intégré dans un système RBMT (Apertium) afin de le rendre rapide pendant la traduction et flexible pour les utilisations humaines. Selon Tyers, les résultats obtenus présentent une amélioration statistiquement significative de la qualité de la traduction.

Quant à la MT hybride, dans (Scherrer and Ljubeši´c, 2016), une méthode de sélection lexicale a été intégrée dans un module de désambiguïsation basé sur des règles d'un RBMT Espagnol-Quechua afin d'améliorer les choix de mots ambigus. Pour ce faire, ils ont utilisé les estimations de probabilité les mieux classées, fournies par des classifieurs entraînés sur des corpus bilingues. Nous mentionnons également une étude de (Simov et al., 2016) où les auteurs ont présenté une méthode de sélection lexicale par substitution de formes de mots Anglais par des synsets[34] ou des lemmes représentatifs Bulgares. À cette fin, après une phase de WSD basée sur des graphes, un alignement entre les WordNets Bulgares et Anglais a été réalisé pour générer des règles de substitution de mots et de génération de corpus, puis ce corpus a été utilisé pour entrainer un système SMT.

En conclusion, toutes les recherches citées couvrant les différentes approches de MT prouvent l'efficacité de la WSD pour préserver le sens des mots pendant la traduction. En outre, la majorité des techniques de traduction de CS, et aussi celles de désambiguïsation de la traduction des mots sont basées sur des corpus parallèles, dont la construction est très difficile, ou elles reposent sur les règles et donc dépendent de la langue.

Dans cette section, nous avons présenté les travaux réalisés autour de la normalisation du CS à travers les techniques de traduction. La section suivante décrit les techniques utilisées pour la normalisation du dialecte.

[34] Synsetes : ensemble de synonymes ou ensemble de mots qui partage le même sens.

2.2.3 Normalisation du dialecte

Les dialectes sont des langues utilisées principalement en communication verbale. Cependant, depuis l'avènement du SMS, le dialecte s'est introduit dans les communications écrites, puis il est devenu très répandu dans les médias sociaux. Ces langues ne possèdent pas une orthographie standard parce qu'elles ne sont pas utilisées comme langues formelles. Par conséquent, chaque auteur, dans les différents médias, écrit suivant sa prononciation et sa volonté. Du coup, pour un seul mot nous trouvons différentes écritures. Ce qui constitue un obstacle majeur devant le traitement automatique des textes, d'où vient la nécessité de normaliser ces écritures.

La normalisation du texte peut être considérée comme le successeur de la correction orthographique, sauf que dans le premier cas, le style d'écriture bruyant est souvent intentionnel (par exemple les abréviations), alors que dans le second cas, les fautes d'orthographe sont involontaires. Les premières approches de normalisation des textes étaient basées sur les règles en plus du modèle du canal bruyant ou 'noisy channel model' (Shannon, 1948) qui était principalement lié aux techniques de correction orthographique. Elles ont été utilisées conjointement pour la normalisation automatique. Parmi ces règles, on trouve des règles lexicales utilisant la distance d'édition[35] (Sidarenka et al., 2013), celle-ci peuvent détecter les fautes d'orthographe et les modèles d'expression régulière utilisés pour supprimer ou remplacer les répétitions de caractères, URL, hashtags, et logogrammes[36] inutiles. Elles peuvent également être utilisées pour la détection des mots spéciaux aux SM (Cotelo et al., 2015). En outre, les règles phonétiques utilisant les variantes de l'algorithme Soundex peuvent servir à normaliser les mots bruyants liés à des différences de prononciation (Eryigit and Torunoglu-Selamet, 2017). Le modèle de canal bruyant normalise les mots à travers la sélection des formes les plus probables selon le classement de leurs probabilités fourni par un modèle de langage. Il a été utilisé dans le cadre de l'apprentissage supervisé et aussi non supervisé (Cook and Stevenson, 2009). En général, ces approches capturent les différences entre les formes de surface d'un mot en détectant la similarité au niveau lexical entre ses formes formelles et informelles. Cependant, le niveau sémantique reste inaccessible

[35] La distance d'édition est le nombre d'opérations appliquées pour transformer une chaîne de caractères en une autre. Elle permet de mesurer la similitude lexicale entre les chaînes de caractères. La distance de Levenshtein (Levenshtein, 1966) est la mesure la plus utilisée, elle inclut les opérations d'insertion, de suppression et de substitution.

[36] Utilisation d'une seule lettre ou d'un seul chiffre pour représenter un mot ou une partie de mot.

puisque ces techniques ne sont pas capables de saisir le contexte des mots. Le problème ici, est qu'un mot informel peut être attribué à un mot formel, uniquement sur la base de sa forme lexicale sans tenir compte de sa signification, ce qui peut constituer une source d'ambiguïté.

Afin de pallier cet inconvénient, l'apprentissage supervisé, la traduction automatique et d'autres techniques ont été utilisés. Pour l'apprentissage supervisé, des caractéristiques telles que les n-grammes des caractères, le word embedding, le tag POS, les distances d'édition, les listes de recherche et autres sont utilisées avec des données labellisées comme dans (Van Der Goot and Van Noord, 2017). En outre, différentes architectures de réseaux de neurones ont été adoptées, comme le modèle LSTM (Long Short Memory) pour prédire la forme canonique du mot, en utilisant le mot lui-même et les mots qui l'entourent comme dans (Min and Mott, 2015). Dans un travail très récent, Muller et al. (2019) ont essayé d'utiliser BERT dans le but d'apprendre la normalisation lexicale de l'Anglais. Cette tâche a également été abordée comme une tâche de traduction, une technique considérée sensible au contexte, dont le but est de traduire un texte bruyant en texte standard en utilisant des corpus parallèles. L'approche SMT-like (Kaufmann and Kalita, 2010) et aussi la CSMT qui est une SMT au niveau caractères (Scherrer and Ljubeši´c, 2016) ont été utilisées. Cette dernière a donné de meilleurs résultats que la SMT au niveau des mots. Quant à la MT neuronale (NMT), Lusetti et al. (2018) ont employé un encodeur-décodeur neuronal avec un modèle de langage au niveau des mots et des caractères, cette technique a dépassé les performances de la CSMT. Néanmoins, ces techniques nécessitent une grande quantité de données annotées, ce qui est en soi une tâche complexe et très coûteuse.

Pour résoudre ces problèmes, Sridhar (2015) a été le premier à introduire la normalisation contextualisée de manière totalement non supervisée. Il a utilisé la représentation distribuée de mots ou le word embedding afin de saisir la similarité contextuelle qui permet de correspondre un mot bruyant à sa forme canonique, s'ils partagent la même représentation vectorielle. En d'autres termes, leurs vecteurs sont les plus proches les uns des autres parmi tout le vocabulaire. Puis, il a représenté le lexique résultant avec des machines à états finis (FSM). Finalement, le processus de normalisation est effectué en transduisant les mots bruyants à partir de la FSM. Les principaux avantages de cette technique sont, premièrement, la non nécessité de corpus labellisé. Sridhar a utilisé Twitter et des notes du service clientèle comme données d'entraînement. Cette technique est donc extensible et adaptable à n'importe quelle langue. Son deuxième avantage est la présence de la dimension contextuelle qui a été

un facteur clé de sa haute performance dans cette tâche dépassant les précisions de Microsoft Word et Aspell.

Ces qualités positives ont inspiré d'autres travaux sur la normalisation. Bertaglia and Nunes (2016) ont réalisé la normalisation du portugais en utilisant un modèle de word embedding entraîné sur des revues de produits et des tweets. Ils ont construit un dictionnaire faisant la correspondance entre les mots bruyants et ceux canoniques pour représenter le lexique. Ils ont mené des expériences sur l'argot sur Internet et la correction d'erreurs orthographiques. Les résultats obtenus ont dépassé les outils existants. Htait and Bellot (2018) ont adopté la même approche pour construire des dictionnaires de normalisation, et donc pallier le manque de ce type de ressources pour l'Anglais, le Français et l'Arabe. Ils ont employé des corpus de Twitter, et ils ont également atteint des performances importantes dans les trois langues.

Tous ces travaux ont été réalisés dans le cadre de la normalisation des langues standards. Pour les dialectes qui souffrent d'un manque de ressources, les travaux connexes sont très limités. Par exemple, Al-badrashiny et al. (2014) ont introduit un système qui génère une liste de toutes les translittérations possibles pour chaque mot d'une phrase d'entrée. Ils ont entraîné un transducteur à état fini sur des corpus parallèles de mots arabophones Egyptiens. L'objectif est d'aligner les caractères du dialecte Egyptien écrit en Arabizi (écriture latine) à l'écriture Arabe. Partanen et ses collègues (Partanen et al., 2019) ont utilisé le NMT au niveau caractères pour convertir le Finnois dialectal en Finnois standard. Le modèle a été entraîné sur un corpus composé de transcriptions d'enregistrements vocaux depuis 1950, et qui a été annoté manuellement. Le word embedding a également été utilisé pour construire un corpus comparable et un lexique du dialecte Algérien (Abidi and Smaili, 2018) par l'alignement d'un corpus extrait de YouTube. Le lexique construit associe entre les différentes formes de translittérations des mots dialectaux écrits en script Arabe et ceux en Latin.

En tant que dialecte, l'Arabe Marocain (AM) est une langue qui manque énormément de ressources et aussi d'outils nécessaires pour son traitement, et surtout pour la normalisation. Les seuls travaux effectués à cette fin sont : Tachicart and Bouzoubaa, (2019) et récemment Tachicart and Bouzoubaa, (2021). Dans le premier, les auteurs se sont servis d'un corpus de Facebook et YouTube pour analyser l'incohérence orthographique du dialecte AM utilisé en SM, écrit en script Arabe et aussi en Latin. Ils ont comparé leur corpus avec un dictionnaire

de référence qui a été préalablement construit par les auteurs et ils ont constaté que 35% de ce texte est bruyant. Ils ont conclu qu'un outil de correction orthographique est essentiel pour nettoyer et convertir les mots dialectaux en une forme d'écriture standard unique. Dans le deuxième travail, les auteurs ont construit, d'une façon non supervisée, un lexique qui peut servir à la normalisation d'orthographie de la AM. Ils se sont focalisés sur le dialecte écrit en script Arabe et la génération du lexique a été faite en se basant sur un modèle FastText en plus d'un vocabulaire de référence. Sur un sous-ensemble de 1200 mots référence qu'ils ont pu valider, ils ont eu un taux d'erreur de 8.65%.

2.2.4 Détection du contenu violent dans les réseaux sociaux

La cyberviolence entre pairs peut prendre plusieurs formes. Plusieurs études ont été menées pour l'identification de ces formes, en particulier les contenus abusifs (Founta et al., 2017; Nobata and Tetreault, 2016), l'agression (Conicet et al., 2018; Kumar et al., 2018), le cyberharcèlement (Dadvar et al., 2014; Yao et al., 2018), le discours de haine (Davidson et al., 2017; Shervin and Zampieri, 2017) et le langage offensif (Wiegand and Siegel, 2018). Pourtant, le cyberharcèlement (cyberbullying) reste la forme la plus couverte par ces études. En effet, il touche une catégorie très vulnérable, à savoir les adolescents, et constitue le principal problème dangereux qui menace le bien-être et la santé mentale des victimes et, du coup, leur réussite dans la vie. Par définition (Paul et al., 2012), le cyberharcèlement implique un comportement agressif intentionnel de façon répétée. Cependant, en analysant ces études, nous avons constaté que la majorité d'entre elles se souciaient d'identifier le comportement agressif des auteurs et négligeaient son caractère répétitif et intentionnel. Cela revient à la difficulté de tracer et suivre l'activité des cyberharceleurs qui agissent sous le couvert de l'anonymat en inventant des identités virtuelles dont ils peuvent facilement changer. Le point commun entre tous les travaux relatifs au cyberharcèlement est la détection du contenu agressif dans les textes rédigés par les acteurs de violence. Par conséquent, les approches adoptées par ces études restent applicables pour les autres formes de cyberviolence.

En générale, la réalisation d'une tâche de détection se passe soit à travers les techniques ML classiques, celles-ci nécessitent l'ingénierie des caractéristiques plus un algorithme qui se charge de la détection, soit à travers les techniques DL. Quant au DL, puisque les caractéristiques sont créées d'une façon autonome, donc la détection se base seulement sur l'algorithme utilisé. Cependant, le DL requiert de grande quantité de données annotées afin

d'assurer de bonne performance.

Le reste de cette section sera organisé comme suit, après un bref aperçu des travaux connexes en psychologie, nous présentons ci-dessous quelques travaux liés à la détection du contenu violent. Notre classification se base sur l'enquête élaborée par (Salawu et al., 2017), que nous prétendons être très structurée et informative.

2.2.4.1 La cyberviolence en psychologie

Il est de plus en plus reconnu que la mesure de la violence dans toute sa complexité et sa multi dimensionnalité chez les individus reste très difficile (Nansel et al., 2001). Mais les chercheurs ont réussi à trouver des caractéristiques psychologiques et émotionnelles communes chez les cyber-agresseurs. Parmi ces caractéristiques, nous trouvons les traits de personnalité, tels que les cinq grands traits de personnalité ou Big Five, à savoir l'ouverture, la conscience, l'extraversion, l'agréabilité et le neuroticisme, qui constituent un critère essentiel dans les études psychologiques sur la cyberviolence (Kowalski et al., 2014).

Les recherches menées montrent que les auteurs de ces actes sont associés à une mauvaise adaptation psychosociale (Gumpel and Sutherland, 2010). Par exemple, les individus qui n'ont pas la capacité de comprendre les émotions des autres (manque d'empathie) ont tendance à adopter des comportements plus violents (Kowalski et al., 2014). En outre, les auteurs de ces actes font preuve d'une faible adaptation, tant sur le plan social (isolement, manque de réussite dans la vie) qu'émotionnel (peur, anxiété, dépression), en plus des troubles de comportement comme l'agressivité (Gumpel and Sutherland, 2010).

D'autres travaux ont constaté que les jeunes développent des émotions négatives telles que la colère et la dépression en raison de divers facteurs de tension, et qu'ils exécutent des actes violents pour évacuer ces sentiments négatifs (Olweus, 1978). Finalement, le fait de commettre un acte de cyberviolence peut avoir des effets négatifs sur l'avenir de ses auteurs. Par exemple, il a été observé que le comportement criminel des anciens auteurs a quadruplé à l'âge de 24 ans (Peterson and Densley, 2017).

En conclusion, la cyberviolence est un problème de santé auquel nous devons faire face. Les chercheurs dans ce domaine affirment qu'il est nécessaire d'examiner les caractéristiques des auteurs, à savoir les traits externes (la psychopathie) et potentiellement les traits internes (e.g. le narcissisme) qui présentent relativement un nouveau domaine d'étude (Paul et al.,

2012).

Par la suite nous présentons les techniques computationnelles employées pour l'identification des actes violents.

2.2.4.2 Techniques computationnelles employées pour la détection des actes violents

Dans cette section, nous présentons les différentes caractéristiques notamment celles d'apprentissage ainsi que les techniques adoptées pour la détection des actes violents.

a. Revue des caractéristiques d'apprentissage

L'extraction de caractéristiques vise à identifier les éléments pertinents d'un contenu violent (par exemple, ce qui fait qu'un message ou un poste est considéré comme un contenu violent et pourquoi). Dans la littérature, les caractéristiques de la cyberviolence sont classées en quatre groupes principaux, à savoir les caractéristiques basées sur le contenu, les caractères psychologiques, l'utilisateur et les informations du réseau.

Caractéristiques basées sur le contenu

Nous définissons ce groupe de caractéristiques par les éléments qui constituent un document texte, y compris les caractéristiques lexicales telles que les mots clés (par exemple, les mots profanes) (Dinakar et al., 2011; Kontostathis et al., 2013). Cependant, l'utilisation de mots clés seuls, ne peut pas être très indicative. Par conséquent, d'autres chercheurs ont utilisé des caractéristiques syntaxiques telles que les pronoms et la ponctuation (Dadvar et al., 2014). D'autres caractéristiques telles que les n-grammes, les sacs de mots ou Bag Of Word[37] (BOW) et le Term Frequency-Inverse Document Frequency[38] (TF-IDF), sont aussi basées sur le contenu. Leur utilisation est très répandue parmi les études conduites dans ce sens (Dadvar et al., 2012b; Dinakar et al., 2011; Munezero et al., 2014; Nahar et al., 2013).

Caractéristiques basées sur l'utilisateur

Des caractéristiques telles que l'âge, le sexe, l'orientation sexuelle, la race et le style d'écriture sont considérées comme des caractéristiques spécifiques à l'utilisateur (Nahar et

[37] Bag of Words : est un ensemble de tokens qui n'ont pas un ordre spécifique. Contraire de liste ou séquence de tokens ou l'ordre est respecté.
[38] TF-IDF : est une méthode de pondération qui permet d'évaluer l'importance d'un terme contenu dans un document, relativement à un corpus de documents. Son objectif est de valoriser les termes rares et dévaloriser ceux fréquents comme les mots vides (prépositions, pronoms, articles...).

al., 2014; Squicciarini et al., 2015).

Caractéristiques basées sur les informations du réseau

Les réseaux SM peuvent également être une source de caractéristiques des systèmes de détection de la cyberviolence telles que le nombre de vues, le nombre de 'J'aime' (Like), le nombre de téléchargements et le nombre d'amis (Huang et al., 2014; Nalinipriya and Asswini, 2015).

Caractéristiques basées sur les caractères psychologiques

L'analyse des sentiments, des émotions ou de la personnalité peut avoir un fort impact sur la détection de la cyberviolence. Néanmoins, la plupart des travaux connexes se sont intéressés surtout aux sentiments (positifs, négatifs et neutres) comme dans (Nahar et al., 2012) et aussi aux émotions (Dinakar et al., 2011; Munezero et al., 2014; Xu et al., 2012) où des lexiques des émotions ont été utilisés avec la correspondance lexicale. Récemment des études ont commencé d'investiguer l'efficacité des caractères de la personnalité dans cette tâche (Balakrishnan et al., 2019; Rosa et al., 2019) et qui ont montré leur efficacité. Pour cela ce domaine nécessite encore un approfondissement.

Toutes ces caractéristiques ont été utilisées dans plusieurs travaux individuellement ou en combinaisons. Cependant les études montrent que la sélection des caractéristiques d'apprentissage permet d'améliorer les performances des classifieurs (Robinson et al., 2018). D'autre part, les caractéristiques psychologiques n'ont pas été suffisamment explorées alors qu'elles sont des facteurs primordiaux dans les études psychologiques sur la cyberviolence.

b. Revue des techniques de détection du contenu violent

Dans la littérature, les techniques de cyberviolence sont également classées en quatre grandes approches : l'approche basée sur le lexique, l'approche basée sur les règles, l'approche d'apprentissage supervisé et l'approche hybride.

L'approche basée sur le lexique

L'approche basée sur le lexique se compose généralement de trois étapes : Tout d'abord, un lexique est créé contenant des mots ou des phrases de nature nocive, puis un analyseur passe en boucle tous les messages, à la recherche de ces mots/phrases nocives. Enfin, les messages reçoivent un score de violence en fonction du nombre ou du poids des mots violents

qu'ils contiennent (Kontostathis et al., 2013).

L'approche basée sur les règles

La méthode basée sur des règles est liée à l'utilisation d'un ensemble de règles inspirées des connaissances des experts humains dans le domaine d'intérêt. Par exemple, (Mahmud et al., 2008) ont utilisé un analyseur syntaxique pour trouver des relations entre les éléments de la phrase. De plus, (Bretschneider et al., 2014) ont défini un ensemble de règles pour détecter des patterns de relation entre des mots profanes et une référence à une personne.

L'approche de l'apprentissage supervisé avec les ML classiques

Cette approche vise à trouver des patterns implicites de caractéristiques dans des messages précédemment labellisés. C'est l'approche la plus adoptée dans les systèmes de détection de la cyberviolence. Dans leur travail Dadvar et al. (2012a) ont utilisé le sexe des utilisateurs en ligne comme caractéristique pour entrainer un classifieur SVM. Puis, dans (Dadvar et al., 2012b) ils ont ajouté une nouvelle caractéristique qui mesure les états émotionnels des victimes après un épisode de cyberviolence pour améliorer le processus de détection. D'autres techniques ont été également utilisées, comme dans les travaux (Balakrishnan et al., 2019; Chatzakou et al., 2017), où ils ont entraîné un classifieur Random Forest et ont obtenu de bonnes performances dans la détection du cyberharcèlement. Pour le même objectif, Al-garadi et al. (2016) ont entraîné un classifieur Random Forest et un autre LibSVM, ce dernier modèle a été le plus performant.

L'approche hybride

De nombreux chercheurs ont essayé de combiner des approches en intégrant les connaissances humaines aux algorithmes d'apprentissage machine, afin de tirer parti des deux techniques, comme cela a été fait dans (Dadvar et al., 2011; Sheeba and Vivekanandan, 2013).

L'approche de l'apprentissage supervisé avec les Techniques Deep learning

Récemment, l'apprentissage profond ou deep learning (DL) a attiré une grande attention dans plusieurs tâches de NLP et de Text Mining par ses performances supérieures aux algorithmes de ML classiques. Le DL a été utilisé de manière significative ces dernières années dans la détection de la cyberviolence. Dans le cas de (Badjatiya et al., 2017), ils ont attaqué le problème de la détection de discours de haine, et spécifiquement la détection du racisme et

du sexisme, en appliquant différentes architectures DL. Ces architectures comprennent les réseaux de neurones convolutionnels (CNN), les réseaux récurrents à mémoire court et long terme (LSTM) et FastText, combinés à de nombreuses caractéristiques telles que TF-IDF et BOW. Leur classifieur LSTM avec un word embedding aléatoire a obtenu des performances nettement améliorées par rapport aux techniques de base. Dans (Gambäck and Sikdar, 2017), des modèles CNN ont été utilisés pour détecter le discours de haine sur Twitter. Comme embedding, Ils ont employé des vecteurs aléatoires (one-hot encoding), ainsi que le word embedding et ils ont également concaténé des modèles CNN basés sur les mots et d'autres basés sur les caractères pour classer 6909 tweets en 4 classes. Tommasel et al. (2018) a présenté une approche pour la détection automatique des agressions basée sur la combinaison des modèles SVM et DL afin d'analyser un large éventail de caractéristiques : les caractères de mots, les mots, le word embedding, ainsi que des caractéristiques relatives aux sentiments et l'ironie. Leurs résultats montrent que la détection de l'agression est une tâche assez complexe, surtout lorsqu'elle est exprimée implicitement dans le texte (comme dans l'ironie et le sarcasme).

Des travaux récents cherchent à tester la généralisation des méthodes DL à d'autres datasets et domaines. Agrawal and Awekar (2018) testent l'apprentissage par transfert ou 'transfert learning' pour vérifier si les connaissances acquises par des modèles DL (CNN, LSTM, BLSTM et BLSTM avec attention) sur un dataset peuvent être utilisées pour améliorer les performances de la détection du cyberharcèlement sur d'autres datasets extraites de différentes plateformes SM. Ils ont utilisé trois datasets : Wikipedia, Formspring et Twitter. Les auteurs montrent que le transfert de modèles entraînés sur Wikipedia donne de bonne performance sur Formspring et vice versa. Par contre, le transfert de Twitter vers les deux autres domaines est peu performant. Dans une étude similaire, Dadvar and Eckert (2018) reproduisent les mêmes techniques en effectuant un transfert learning de Twitter vers un dataset YouTube montrant une hausse en performance.

D'autres travaux s'étaient intéressés à la détection de la violence dans plusieurs langues. Par exemple, dans (Ranasinghe et al., 2019), les auteurs identifient le langage offensif dans les tweets et les messages Facebook en Allemand, en Anglais et en Hindi. Le système utilise un prétraitement minimal et s'appuie sur le word ainsi que le contextuel embedding pour l'encodage du texte d'entrée. Ils ont expérimenté différentes architectures DL, et les meilleurs résultats ont été obtenus par l'architecture BERT. Dans un travail très récent (Samghabadi et

al., 2020), BERT a été utilisé dans une approche multi-tâche, pour détecter la misogynie et l'agressivité pour trois corpus différents Anglais, Hindi et Bengali, ce qui leur a permis d'aboutir à de très bonnes performances.

En résumé, parmi les cinq approches qui ont été présentées précédemment, l'apprentissage supervisé est l'approche dominante dans les études sur la cyberviolence. Les techniques DL restent les plus performantes, mais nécessitent des corpus annotés et volumineux. Il convient également de noter que les chercheurs ont montré un grand intérêt pour l'utilisation des techniques ML classiques et d'autres provenant d'autres domaines tels que la linguistique (via les règles et les lexiques), le NLP et la psychologie comportementale. Ainsi, nous pensons que la combinaison de l'apprentissage automatique avec les connaissances des experts en psychologie (comme les caractères de la personnalité) a de grandes chances de concevoir des systèmes qui permettront de prédire les comportements nuisibles des utilisateurs avec une grande précision surtout en cas de dataset de taille limitée.

2.3 Conclusion

Dans ce chapitre, nous avons procédé à une revue de la littérature portant sur les différentes approches utilisées dans les différentes tâches NLP que nous aborderons dans les chapitres suivants. En particulier, les travaux relatifs à la normalisation du Code Switching, du dialecte, et du langage spécifique aux réseaux sociaux. Ce qui servira comme un prétraitement du texte d'entrée afin de le préparer pour la tâche d'analyse. Nous avons également essayé de couvrir les différentes approches s'intéressant à la détection du contenu violent dans les textes SM. D'autres détails seront donnés pour chacun des traitements spécifiques durant les chapitres à venir.

Chapitre 3

Normalisation des Textes Bruités

Dans ce premier chapitre, nous présenterons les différents prétraitements requis pour normaliser les textes bruités avant de passer à leur analyse. En particulier, la normalisation du Code Switching à travers une technique de traduction hybride tout en détaillant comment nous avons remédié aux autres phénomènes linguistiques responsables du bruit.

Les travaux élaborés dans ce chapitre ont fait l'objet des publications suivantes :

- Randa Zarnoufi, Walid Bachri, Hamid Jaafar, Mounia Abik, "MANorm: A Normalization Dictionary for Moroccan Arabic Dialect Written in Latin Script", COLING/WANLP, Barcelona, Spain (Online December 2020). https://www.aclweb.org/anthology/2020.wanlp-1.14/

- Randa Zarnoufi, Hamid Jaafar, and Mounia Abik. 2020. Machine Normalization: Bringing Social Media Text from Non-Standard to Standard Form. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 19, 4, Article 49 (April 2020), 30 pages. *https://doi.org/10.1145/3378414*

- Zarnoufi R., Jaafar H., Abik M. (2019) Language Identification for User Generated Content in Social Media. In : Rocha Á., Serrhini M. (eds) Information Systems and Technologies to Support Learning. EMENA-ISTL 2018. Smart Innovation, Systems and Technologies, vol 111. Springer, Cham. https://doi.org/10.1007/978-3-030-03577-8_73

3.1 Introduction

Une grande partie de la population mondiale est quotidiennement connectée et très active dans les SM. Cette communauté produit une énorme quantité de données. Ces dernières, surtout textuelles constituent une mine d'or pour le développement de nombreuses applications NLP ou de texte en général (Farzindar and Inkpen, 2018). Cependant ces textes générés par les utilisateurs des SM ne suivent pas les normes d'écriture standards et contiennent des éléments hétérogènes, tels que les symboles, les abréviations, l'orthographe incorrecte ou inhabituelle, le dialecte, et le Code Switching (CS) (section 3.2.1). Afin d'exploiter ces données textuelles dans des applications telles que, l'analyse des sentiments, l'exploration des opinions ou l'extraction des traits de personnalité, nous devons premièrement s'attaquer à leur nature bruitée. Les techniques NLP existantes ne sont pas en mesure de traiter ce type de texte en raison de sa complexité, et d'ailleurs la plupart de ces techniques sont conçues pour les langues standards et les usages standards (Farzindar and Inkpen, 2018).

La principale préoccupation des chercheurs en NLP reste la compréhension et la révélation de la sémantique derrière les textes. Un tel objectif ne peut pas être facilement atteint vu les défis engendrés par la nature des textes SM, d'où vient la nécessité d'homogénéiser l'ensemble de ces textes pour qu'ils prennent une forme standard avant d'entamer leur analyse. Cet acte nous permettra de gagner plus d'informations sémantiques qu'une simple analyse de mots séparés, indépendants les uns des autres, avec des langues et des formes différentes. De ce fait, l'objectif principal de ce travail consiste à normaliser le texte SM contenant un mélange de langues et d'éléments non standards.

Outre la complexité du traitement du CS, la nature bruitée du texte SM est également liée à l'utilisation de lexique spécial, de dialecte, ainsi que de mots mal orthographiés. Ces utilisations non standards des langues rendent le processus de normalisation de plus en plus compliqué. Dans la littérature, la normalisation des textes SM a été principalement liée à la correction des fautes d'orthographe et à l'identification et au remplacement du lexique spécial du SM par ses équivalents standards. Cependant, la normalisation peut toucher d'autre phénomènes linguistiques. Ainsi, nous avons élaboré une solution plus étendue, qui inclut, en plus des traitements classiques, la normalisation du CS ainsi que celle du dialecte.

Puisque, nous ciblons les approches basées sur les corpus, qui sont utilisées pour la

détection des traits de violence, alors, la normalisation en tant que prétraitement, peut contribuer à l'amélioration des performances d'un tel système, comme indiqué dans (Almeida et al., 2016).

Dans ce chapitre, nous présentons notre solution pour la résolution de ces problèmes de normalisation des textes SM. Cette solution se base sur l'approche MT-like ou de *type traduction automatique*, c'est une approche hybride combinant les règles linguistiques avec des techniques statistiques. Elle comprend la conversion des phrases CS du multilingue au monolingue avec un processus de **traduction-désambiguïsation** comme principale traitement, et ce afin d'assurer le transfert de la sémantique pendant la traduction. De plus, cette solution intègre un ensemble de techniques destinées aux autres formes non standards composant les textes SM comme la normalisation du dialecte et la correction orthographique. Nous appelons notre solution ***Machine Normalisation*** (MN).

Parmi les critères que nous exigeons être vérifiés par notre solution est la généricité. Ce critère permettra à notre solution d'être appliquée sur toute langue possédant les ressources nécessaires. Pour cela, nous avons conçu un système indépendant de la langue utilisée, où nous exploitons et adaptons des ressources et outils existants qui conviennent à notre tâche sans en développer de nouveaux. Ce qui constitue l'un des grands avantages de notre solution.

Dans les systèmes de traduction, afin de préserver le sens à travers la traduction, le choix des langues source et cible est crucial. De ce fait, nous avons abordé l'identification des langues source et cible dans le cas de la CS en se basant sur des théorèmes linguistiques, pour déterminer les différentes langues source et cible. Notre objectif est la définition de la meilleure direction de traduction qui garantit le transfert du sens au niveau de la phrase. En plus, nous avons intégré une étape WSD (la désambiguïsation du sens du mot) pendant la traduction afin de préserver le sens au niveau du mot. Cette WSD est fondée sur une approche basée sur les connaissances, qui est le ***Lesk adapté*** (Banerjee and Pedersen, 2002)

La principale contribution dans ce chapitre consiste en une solution de ***normalisation*** pour convertir le texte du SM de la ***forme non standard à la forme standard*** et par conséquent, permettre son utilisation par des applications de NLP et Text Mining. Cette solution comprend trois traitements : l'***identification des langues source et la langue cible*** basée sur une théorie linguistique, la ***normalisation du dialecte***, et la ***traduction-***

désambiguïsation du **CS**. En plus, nous attaquons aussi d'autres formes d'utilisation non standard de la langue.

Dans les sections suivantes nous discuterons d'abord, les différents phénomènes linguistiques qui caractérisent les textes du SM. Puis nous introduirons notre approche de la normalisation du texte bruité et notamment le CS et le dialecte. Par la suite, nous décrirons les expériences conduites pour valider notre solution et les résultats obtenus.

3.2 Identification des différents phénomènes linguistiques caractérisant les textes du SM

Le langage formel utilisé dans les documents officiels respecte les règles standards d'expression et d'écriture. En revanche, les écrits générés par les utilisateurs dans les réseaux sociaux où la liberté d'expression et d'écriture règnent ne suivent aucune norme. Soit au niveau lexical soit au niveau syntaxique et même des fois sémantique (le sens de certaines expressions n'est pas compréhensible que par certaines catégories de personne). Pour ces raisons, ce type de texte est appelé *texte bruité.*

A cause de leur nature, les textes bruités alternent les performances des systèmes NLP engendrant de grands défis à ce type de tâche. Il est donc indispensable de nettoyer ou de normaliser ces textes avant de les analyser. A cet effet, nous devons tout d'abord identifier les différents types de bruit qu'ils contiennent. En examinant la nature des différents phénomènes linguistique caractérisant ce texte, nous avons repéré : le Code Switching, la variation orthographique des dialectes, l'utilisation d'un lexique spécial aux SM et l'orthographie incorrecte. Dans cette section, nous détaillerons chacun de ces phénomènes.

3.2.1 Code Switching

Une partie importante de la population mondiale communique quotidiennement dans plus d'une langue, que ce soit à l'oral ou à l'écrit. Ces personnes multilingues alternent des mots de deux ou plusieurs langues dans les mêmes phrases. Ce phénomène linguistique est appelé "Code Switching" ou *alternance codique* (la langue est considérée comme un code). Dans la littérature, de nombreux travaux se sont intéressés au CS dans le traitement de la parole, mais pour le texte, nous ne trouvons que quelques études connexes. Cela peut s'expliquer premièrement, par la complexité du traitement de texte CS et deuxièmement par

sa récente expansion due à l'utilisation importante du SM.

Le principal obstacle qui s'impose à la traduction des textes CS, demeure le manque de ressources linguistiques, telles que les dictionnaires bilingues et les corpus parallèles bilingues adaptés aux textes CS. Ce problème touche particulièrement les langues non standards ou les dialectes qui sont en réalité les langues les moins dotées de ressources.

Afin de vérifier l'efficacité des MTs dans la traduction des phrases CS, nous en avons examiné différents types : les traducteurs basés sur les règles (Forcada et al., 2011), les MTs data driven ou guidées par les données qui sont ceux basés sur les statistiques (Koehn et al., 2007), et plus récemment ceux basés sur les algorithmes neuronaux (Johnson et al., 2017). Premièrement, nous avons testé les MT basées sur les règles (e.g. Systran[39]), puis les MT à base de statistique (e.g. Google Translate et Microsoft Translator), et finalement les MT neuronales (e.g. Google Translate, Microsoft Translator [40], Pure Neuronale[41] MT, DeepL[42]).

Pour tous ces types de MT, les traductions ont été effectuées en activant la détection automatique de la langue source et en sélectionnant le Français comme langue cible. Un exemple de résultats de ce test est présenté sur le Tableau 3.1. L'exemple montre la traduction de la phrases CS « daba il m'arrive en mémoire une deep discussion dans le film the dark knight. » contenant des mots de l'Arabe Marocain (daba), du Français (il m'arrive en mémoire) et de l'Anglais (deep, the dark knight). Il faut noter que pour Google Translate et DeepL, ils ont commencé récemment à traduire le CS entre le Français et l'Anglais en utilisant des techniques neuronales.

D'après les résultats de ce test, nous avons constaté que ces systèmes peuvent atteindre une bonne précision pour les langues et les textes standards, cependant ils sont incapables de traiter les textes CS. La plupart d'entre eux rejettent ou translitèrent les mots non reconnus (en particulier pour les dialectes et les abréviations).

[39] https://translate.systran.net/#/translation
[40] Les versions récentes de Google et Microsoft utilisent des algorithmes neuronaux pour certaines langues.
[41] Pure neural MT est une nouvelle MT de Systran : https://www.systransoft.com/systran/translation-technology/neural-machine-translation-nmt/
[42] DeepL est une MT neuronale basée sur la large base de données de traduction Linguee : https://www.deepl.com/translator

Tableau 3.1. Exemple de traduction d'une phrases CS par les systèmes MT les plus connus.

Traducteur	**Langue Détectée**	**Traduction**
Traduction En français (par un humain)	Arabe Marocain (daba) Français (il m'arrive en mémoire une, dans le film) Anglais (deep, the dark knight)	maintenant je me souviens d'une discussion profonde dans le film le chevalier noir.
Systran	Non détectée	Aucun résultat.
Google Translate	Français	daba il m'arrive en mémoire une discussion profonde dans le film le chevalier noir.
Microsoft Translator	Français	daba il m'arrive en mémoire une deep discussion dans le film the dark knight.
Pure Neural MT (PNMT)	Non détectée	Aucun résultat.
DeepL	Anglais	daba it happens to me in memory a deep discussion in the movie the dark knight

Nous avons également remarqué que la principale limitation des traducteurs data driven devant les textes CS est l'identification automatique des différentes langues sources et la langue cible. Si le texte source contient plusieurs langues, ils ne parviennent pas à les détecter. En général, les MTs peuvent détecter automatiquement une seule langue source dans une phrase CS, c'est celle la plus fréquente. Ce qui en résulte que ces MTs effectuent la traduction des textes d'une seule langue source vers une langue cible. Par conséquent ces systèmes ne conviennent pas à la normalisation des textes CS. D'autre part, ces systèmes de traduction ne sont pas des sources ouvertes libres, donc nous ne pouvons pas les exploiter gratuitement.

En conclusion, tous les systèmes de traduction présentés dans cette section ne sont pas adaptés à notre cas d'étude. Vu la nature du CS nous aurons besoin d'une **détection de *plusieurs langues sources*** vers **une *langue cible.*** De plus, pour permettre l'intégration du système MN dans un pipeline de traitement de texte sans aucune intervention humaine, nous aurons besoin d'une solution de détection de langue complètement ***automatique***.

3.2.2 Lexique et abréviations spéciales aux SM

Les abréviations ont vu le jour à cause du nombre limité de caractères imposé par certaines SM plateformes (e.g. 😁 Twitter : 140 caractères par tweet). En plus des

abréviations des mots ou des expressions (**cv** au lieu de **ça va**), nous marquons également l'utilisation de symboles à base de ponctuation (**:(** au lieu de triste), de hashtags (#...), d'URL (https://...), et finalement d'émoticônes comme

3.2.3 Autres formes non formelles

Les SM constituent une foire aux formes d'écriture non standards comme la répétition des lettres utilisée pour marquer un effet verbal (e.g. ouiiiiiiiiiiiiiiii), l'omission de ponctuation et de majuscules. Nous citons aussi la fragmentation de phrase (utilisation de phrases courtes) qui vient de remplacer les phrases complètes et l'utilisation des homophones indifféremment (e.g. er, é, ez à la fin des verbes).

3.2.4 Orthographie incorrecte

Parmi les sources de bruit dans les SM, nous trouvons souvent des textes contenant des fautes d'orthographe. Les erreurs d'orthographie peuvent être divisées en deux catégories selon leur origine (Hládek et al., 2020). Les erreurs cognitives (également appelées orthographiques ou consistantes) : Elles sont causées par des handicaps comme la dyslexie, la dysgraphie ou d'autres problèmes cognitifs. Ces erreurs peuvent être aussi commises par une personne en phase d'apprentissage d'une nouvelle langue dont il ne connait pas les règles et l'orthographie correcte. Dans la deuxième catégorie nous trouvons les erreurs typographiques (également appelées conventionnelles) causées par les erreurs de frappe, qui sont souvent liées aux restrictions techniques du matériel utilisé (e.g. clavier de petite taille) comme le montre l'exemple dans le Tableau 3.2.

Tableau 3.2. Exemple des erreurs d'orthographie suivant leur origine.

Type d'erreur	**Exemple**
Cognitive	Il a été connecter hier a 7h.
Typographique	Quanf est ce que vous allez passer.

3.2.5 Variation orthographique des dialectes

L'utilisation de dialectes est un autre phénomène qui s'ajoute à la complexité de ce type de contenu. Le dialecte est une langue non standard utilisée principalement dans la communication verbale. Sa forme écrite est une translittération phonétique de mots parlés

qui ne suit aucune orthographie standard puisque chaque utilisateur improvise sa propre orthographie. Par conséquent, pour chaque mot, nous trouvons un mélange d'orthographies. Par exemple, en Arabe Marocain, le mot zwin (joli) peut être écrit de toutes les façons suivantes : zwine, zwiin, zouin, zouine.

En conclusion, tous ces phénomènes linguistiques illustrés dans cette section soulignent l'importance du prétraitement de ce genre de texte, pour qu'on puisse l'analyser et d'en extraire des informations pertinentes. Dans la section suivante, nous présenterons en détail notre approche suivie pour normaliser les textes issus des SM.

3.3 Approche

3.3.1 Introduction

La phase de prétraitement a pour but de normaliser les textes du SM, et plus particulièrement la normalisation du CS. Pour ce faire, nous avons adopté une normalisation de type MT (MT-like) comme approche. C'est une approche sémantique hybride combinant une MT peu profonde à base de règles (RBMT ou Rule Based Machine Translation) et des techniques statistiques. Cette solution est indépendante de la langue et repose sur l'utilisation des ressources et outils linguistiques existants, sans en développer de nouveaux. Par conséquent, elle peut être adaptée à d'autres langues en fonction de la disponibilité des ressources spécifiques nécessaires. Elle comprend également une phase de désambiguïsation de la traduction en plus de l'identification automatique des langues source et cible.

Le reste de cette partie est dédié à une présentation détaillée de l'architecture du système *Machine Normalization* et ses différents modules.

3.3.2 L'architecture de l'approche

Notre système de normalisation est fondé sur une approche de type MT. Pour cette dernière, nous avons choisi une hybridation guidée par la RBMT, qui commence par les différentes phases de traitement de la RBMT et suivie par un module statistique. L'architecture générale du système de normalisation est résumée dans la Figure 3.1. Les principales étapes de développement du système y sont représentées.

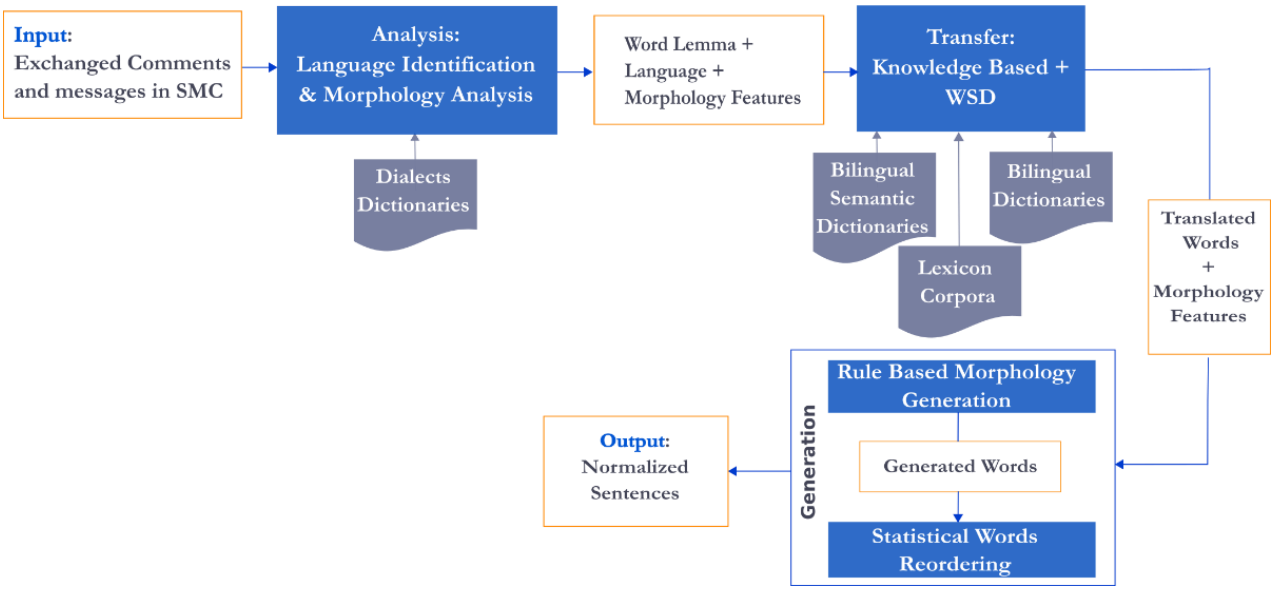

Figure 3.1. Architecture générale du système 'Machine Normalization' de type MT

Le choix de cette MT hybride est lié, d'une part, à la complexité de la structure des phrases CS générées par les utilisateurs des SM. Surtout avec la présence des éléments bruités et du dialecte qui n'a pas de forme normalisée, ce qui ajoute encore de la complexité à cette tâche. La nature de cette structure ne permet pas de faire une analyse syntaxique approfondie, puisque les outils d'analyse existants ne tolèrent pas le CS. Donc, nous sommes incapables d'appliquer la totalité des étapes de traitement de la RBMT. D'autre part, le manque de corpus bilingue nécessaires aux approches basées sur les corpus, en plus de la grande difficulté de leur construction, nous empêche d'adopter l'approche SMT et aussi la NMT.

Notre processus de traduction se déroule en trois étapes principales :

1. L'*analyse,* elle comprend l'identification des langues source et cible en plus de l'analyse morphologique pour extraire le lemme[43] et les caractéristiques linguistiques de chaque mot. Puis, nous passons à la correction orthographique pour traiter les mots mal orthographiés et la normalisation du dialecte.

2. Le *transfert,* il se base sur un traitement de traduction-désambiguïsation.

[43] Le lemme est l'unité autonome constituante du lexique d'une langue. C'est une suite de caractères formant une unité sémantique et pouvant constituer une entrée de dictionnaire. Par exemple, le lemme d'un nom pluriel est son singulier, d'un verbe est son infinitif, etc

3. La *génération,* elle comprend deux étapes. La première est la génération morphologique qui construit la forme flexionnelle des lemmes traduits, en appliquant les caractéristiques morphologiques préalablement extraites lors de l'étape d'analyse. La deuxième étape est le réordonnancement des mots, où nous utilisons un modèle de langue pour corriger la position des mots traduits dans la phrase en fonction de la structure de la langue cible.

Afin de surmonter l'obstacle de l'indisponibilité des corpus appropriés, nous avons adopté des techniques basées sur les connaissances dans la majorité des traitements de notre système. Pour ce faire, nous avons employé des dictionnaires et des corpus lexicaux spécifiques, en particulier pendant les deux phases d'analyse et de transfert. Cependant, au stade de la génération, où nous aurons un texte monolingue (après la traduction des mots vers une seule langue), nous exploiterons des approches statistiques.

L'objectif principal de cette phase de prétraitement étant la normalisation du texte CS, par conséquent, nous nous focalisons sur l'étape de transfert qui permet de convertir le texte CS en texte monolingue. Alors que, l'étape de génération reste facultative en fonction des exigences de l'application cible. Ainsi, durant cette étape, nous effectuons une simple génération morphologique et un simple réordonnancement de mots. Ce choix revient au fait que la précision de ces tâches n'est pas si critique pour les techniques d'apprentissage automatique qui seront employées dans notre application cible.

Dans cette phase de prétraitement, nous présentons également notre solution pour la normalisation du dialecte Arabe Marocain (AM) en utilisant la similarité sémantique à l'aide des modèles word embedding en plus de la similarité lexicale. Nous avons exploité trois modèles de word embedding dont nous avons combiné les résultats pour former un dictionnaire de normalisation MANorm. Dans le dictionnaire qui en résulte, nous trouvons la correspondance entre chaque mot du corpus et la forme de mot la plus similaire du dictionnaire AM.

La traduction des mots de la langue source à la langue cible, nous oblige à assurer un transfert correct de leurs significations. Pour ce faire, nous devons s'attaquer au problème de l'ambiguïté liée à la polysémie des mots. Les mots polysémiques sont des mots qui ont des significations multiples en fonction de leur contexte d'utilisation. Cet aspect linguistique génère une ambiguïté sémantique qui peut inférer le sens d'une phrase. Ce qui produit des

résultats erronés notamment pour les systèmes de traduction.

Pour résoudre ce problème, nous devons inclure la désambiguïsation du sens des mots (WSD) pendant la traduction, comme c'était signalé dans les travaux précédents. Pour cette raison, nous introduisons un système de désambiguïsation de la traduction pour les textes CS basé sur l'algorithme de ***Lesk Adapté*** (Banerjee and Pedersen, 2002), qu'on a combiné avec un contexte vertical multilingue. L'algorithme de Lesk a été critiqué par les chercheurs sur la WSD pour son explosion computationnelle (Agirre et al., 2014), surtout lorsqu'il s'agit de large contexte. Cependant, il reste une solution appropriée pour les langues moins dotées de ressources comme le cas du dialecte et les structures compliquées comme le CS. En plus, aujourd'hui, grâce à l'infrastructure Big Data, le stockage et le traitement de données volumineuses ne posent plus de problème.

Dans la suite de ce document, nous présenterons la totalité des traitements de notre système avec un focus sur la phase de transfert où, nous expliquerons les détails de notre principale contribution, à savoir, la **traduction-désambiguïsation** du texte **CS.**

3.3.2.1 Données d'entrée

L'entrée est composée de phrases issues des commentaires et messages des médias sociaux en particulier, Facebook, Twitter et YouTube. Les données de Facebook ont été extraites avec son API à partir d'une discussion publique au début du projet avant les restrictions et le renforcement de confidentialité établis par Facebook. Quant aux données de Twitter ainsi que celles de YouTube, nous les avons extraites à travers leurs APIs appropriées qui sont toujours ouvertes.

3.3.3 Analyse : Identification des langues et analyse morphologique

L'approche adoptée pour la MT est une hybridation guidée par la MT basée sur les règles. Ce type de MT inclut trois phases : l'analyse, le transfert et la génération. Par conséquent, le premier traitement à réaliser est celui d'*analyse,* qui comprend l'identification de la langue (LID) et l'analyse morphologique (MA). A la fin de ce traitement, nous serons capables d'identifier les langues source et aussi cible en exploitant le résultat de la LID.

Nous avons déjà signalé dans l'état d'art que, la LID est indispensable pour les systèmes MT. En fait, elle permet de définir automatiquement la langue source. Pour notre cas, elle

nous permettra de définir les différentes langues sources et aussi la langue cible de la traduction. Concernant les textes standards monolingues, la LID n'est plus un problème, puisqu'elle opère au niveau de phrase ou document. Cependant, pour les textes SM, l'identification de la langue constitue encore un grand défi. En raison de la taille réduite des textes et de l'usage non standard des langues. De plus, la structure des phrases CS nécessitant l'identification de la langue au niveau de chaque token, rend cette tâche particulièrement complexe.

Les approches de la LID les plus utilisées sont de deux types, l'approche basée sur les corpus et l'approche hybride. Dans notre solution, nous suivons une technique basée sur les connaissances présentée dans le travail de (Elfardy and Diab, 2012). Cette solution consiste à utiliser un analyseur morphologique pour les langues standards et un dictionnaire pour les dialectes. Concernant l'analyse morphologique (MA) des langues sources, nous utilisons une technique proposée par (Solorio and Liu, 2008) où un ensemble d'analyseurs morphologiques sont utilisés. Ils ont atteint une précision de 85,80% sur un corpus en Spanglish, en utilisant le POS[44] tagging avec l'identification de la langue. Le MA est également utilisé pour extraire les caractéristiques morphologiques de chacun des mots dans une phrase CS. Ces derniers seront employés dans la phase de transfert comme un critère de sélection au cours du processus de la WSD. Ces caractéristiques seront aussi appliquées pendant la phase de génération afin de reproduire la forme surfacique des mots traduits.

En outre, cette technique permet de détecter les mots hors vocabulaire (OOV) qui peuvent être des mots mal orthographiés ou des variantes de mots dialectaux. Ces mots hors vocabulaire seront traités par un correcteur orthographique. Un autre avantage de cette technique est qu'elle est extensible, en d'autres termes, nous pouvons ajouter ou supprimer

[44] POS : part of speech c'est partie de discours ou classe de mots, c'est une catégorie linguistique de mots communément définie par son comportement syntaxique : nom, verbe, adjectif, adverbe...

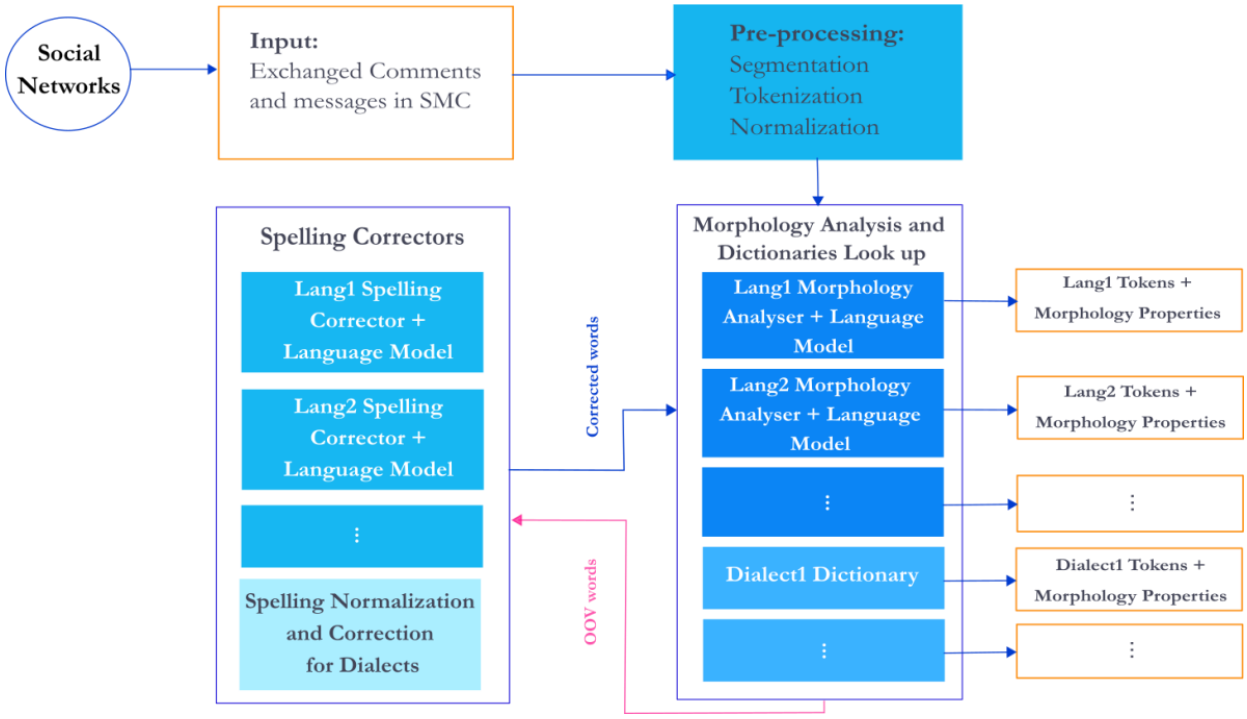

Figure 3.2. Les différents modules de l'étape d'Analyse

des MA ou des correcteurs orthographiques selon les besoins (en fonction des langues incluses dans le CS).

Dans cette section, nous présentons les étapes des traitements LID et MA qui visent à définir respectivement, la langue et les caractéristiques morphologiques des mots dans une phrase CS. L'architecture de notre LID et également le MA pour le CS est présenté sur la Figure 3.2.

3.3.3.1 Prétraitement

L'objectif du prétraitement est de préparer le texte pour les phases suivantes. Ce processus commence par la segmentation des phrases afin de les séparer. Ensuite, la tokenisation pour séparer les mots dans chaque phrase. Enfin, la normalisation, cette dernière étape vise à traiter les effets de la parole, par la suppression des caractères répétés plus de deux fois dans un mot (par exemple : ouiiiiiiiiii devient ouii, goooooood devient good, greaaaaaat devient greaat). La normalisation touche également la conversion des caractères du texte en minuscules et la suppression des hashtags et des URLs.

Nous signalons que nous n'avons pas supprimé la ponctuation et les émoticônes, car elles sont aussi importantes pour d'autres tâches comme l'analyse des sentiments. En outre, elles peuvent être facilement remplacées par leur signification dans la langue cible. Pareil, pour les entités nommées[45], qui ne sont pas traitées, puisqu'elles n'influencent pas le résultat de la LID.

Toutes les étapes précédentes préparent le texte à la phase d'analyse morphologique qui sera présentée dans ce qui suit.

3.3.3.2 Analyse morphologique

Afin d'identifier la langue de chaque token dans une phrase CS, nous avons utilisé un ensemble d'analyseurs morphologiques (MA) pour les langues standards et des dictionnaires pour les langues non standards ou dialectes. Nous notons que tous les outils et ressources utilisés seront présentés dans la section d'évaluation.

Notons qu'avant de procéder à l'analyse morphologique, nous vérifions d'abord si les tokens contiennent des chiffres. Ces tokens appartiennent souvent à des expressions spéciales du SM ou ils peuvent être des mots dialectaux. Dans ce cas, nous vérifions s'ils figurent dans le lexique spécial du SM ou dans le dictionnaire dialectal. Sinon, ce sont des mots mal orthographiés qui doivent être examinés par le correcteur orthographique du dialecte.

La LID fonctionne de telle façon que nous faisons passer chaque token par l'ensemble des MA pour les langues standards (une MA pour chaque langue concernée). Une fois qu'il est reconnu par l'un d'eux, nous lui attribuons comme langue celle de l'analyseur. Dans (Solorio and Liu, 2008), ils ont utilisé une cascade de MA, mais nous avons remarqué que les mots communs entre différentes langues peuvent être détectés par le premier MA et donc incorrectement étiquetés par sa langue. Par conséquent, plutôt que d'utiliser des MA en cascade, chaque token accède à l'ensemble des MA en parallèle. Si l'un des jetons a été reconnu par plus d'un MA, dans ce cas, un ensemble de modèles de langue n-grammes (pour toutes les langues concernées) résoudra ce conflit en sélectionnant la langue dans laquelle ce

[45] Une entité nommée est toute chose que nous pouvons référencer par un nom propre. (Jurafsky, Speech and Language Processing, 2019, p.334)

mot est le plus fréquemment utilisé.

Quant au dialecte, nous recherchons chaque mot probablement dialectal (rejeté par les MAs) dans le dictionnaire du dialecte. Ce dictionnaire est organisé selon les catégories POS. Alors, nous recherchons le mot dans les différentes catégories. Si nous le trouvons dans une catégorie, nous lui attribuons comme langue celle du dialecte en plus de la catégorie POS.

Il est à noter que nous considérons les mots d'emprunt utilisés dans le dialecte, comme des vocabulaires appartenant à ce dialecte. Par exemple, "tconnecta" en Arabe Marocain est emprunté du Français "connecté" et adapté avec l'utilisation des règles de phonologie et de morphologie de l'Arabe Marocain. Ces mots sont considérés comme des néologismes en Arabe Marocain. Ils sont surtout utilisés dans le langage des réseaux sociaux. Leur détection se fait à l'aide d'un lexique spécial collecté à partir des SM.

Ce bloc de traitement fournit en sortie la langue de chaque token et ses propriétés morphologiques : le lemme du mot + la langue + POS (nom, verbe, adjectif, adverbe ...) + genre + nombre + personne + temps du verbe (voir exemple dans3.3.6.3). Ce traitement est effectué d'une part pour réduire les candidats à la traduction lors du transfert et augmenter ainsi la précision du transfert lexical. D'autre part, ces propriétés extraites seront très utiles dans la phase de génération. A la fin de ce processus, tous les mots non reconnus par les MA seront vérifiés par les correcteurs orthographiques.

3.3.3.3 Correction orthographique

En général, le processus de la correction orthographique passe par trois étapes. La première étape c'est la détection des erreurs, suivie par la génération des candidats, et enfin le classement et la sélection des candidats. Les correcteurs open source les plus utilisés sont Aspell[46] et Hunspell[47]. Nous pouvons distinguer cinq types d'erreurs causées par les différentes opérations d'insertion, de permutation, de suppression, de répétition et de remplacement.

Les mots non reconnus par les analyseurs morphologiques et aussi par les dictionnaires sont appelés mots OOV (Out Of Vocabulary) ou mots hors vocabulaire. Ces OOVs sont très

[46] http://aspell.net
[47] http://hunspell.github.io/

probablement des mots mal orthographiés ou des variantes de mots dialectaux. Ils doivent donc, être traités en premier lieu par les correcteurs orthographiques. Ceci est fait à travers une multitude de correcteurs, un pour chacune des langues utilisées dans les phrases CS. L'objectif est de rechercher une forme correcte pour le mot OOV et de trouver ainsi sa langue.

Chaque mot OOV est examiné par tous les correcteurs qui proposent une correction pour ledit mot. Une fois que le mot est reconnu par un correcteur, il est étiqueté par sa langue. Dans le pire des cas, nous le gardons inchangé et nous lui attribuons le tag "NA" comme étiquette de langue. Nous avons utilisé également des modèles de langue n-gram pour traiter les mots multi-reconnus, de la même manière que pendant la phase d'analyse morphologique. Après la correction, chaque token reconnu retourne à son MA approprié (en fonction de sa langue) afin de l'analyser une seconde fois, et d'en extraire les caractéristiques morphologiques.

Les correcteurs orthographiques utilisés pour les langues standards sont basés sur le correcteur orthographique statistique Norvig[48]. Ce correcteur, connu par sa simplicité, est basé sur un modèle linguistique statistique qui propose en sortie la plus probable correction du mot entré. Le fonctionnement de ce correcteur passe par trois étapes. Premièrement, l'algorithme génère pour chaque mot incorrect l'ensemble des candidats possibles. Pour cela, il applique sur ce mot les différentes opérations d'édition, à savoir : l'insertion, la permutation, la suppression, la répétition et le remplacement. De plus, il limite la liste des candidats aux mots connus figurant sur un dictionnaire. Puis, il répète cette procédure une seconde fois afin d'obtenir des candidats avec une distance d'édition plus grande (pour les OOVs avec deux erreurs). Finalement, il utilise un modèle de langage uni-gram entraîné sur un large corpus, qui associe chaque mot à sa probabilité d'apparition. Le mot candidat ayant la probabilité la plus élevée est considéré comme la bonne correction.

Il est à noter que nous avons employé l'algorithme de Norvig dans sa version de base c'est à dire avec un modèle de langage uni-gram. Bien qu'un modèle de langue plus contextualisé, par exemple 3-grams, donnera des résultats meilleurs. Mais, vu la nature multilingue des mots composant les phrases CS, nous ne pouvons pas élargir le contexte à ce stade de traitement.

[48] https://norvig.com/spell-correct.html

Nous signalons que, la normalisation des abréviations dans notre système est basée sur l'utilisation des dictionnaires spécialement conçus à partir de lexique des SM. Ce traitement se fait avant la correction orthographique afin de détecter et de remplacer en premier lieu les abréviations existantes (par exemple remplacer 'mdr' avec 'mourir de rire') par leurs formes standards. Cette opération concerne toutes les langues détectées.

Quant au dialecte qui ne possède pas d'orthographie standard, nous avons construit un dictionnaire de normalisation qui sera présenté dans la section 3.3.4 suivante. Nous l'avons construit pour le dialecte Arabe Marocain, mais la technique reste extensible aux autres langues.

3.3.3.4 Identification des langues sources et de la langue cible.

Comme nous l'avons évoqué auparavant, la langue cible de notre traducteur est la langue dominante (langue matrice). Nous la définissons par : la langue la plus fréquente parmi les langues présentes dans une phrase CS. Par conséquent, son identification par rapport aux autres langues est liée au nombre de mots appartenant à chacune des langues utilisées. Pour cela, nous comptons le nombre de mots appartenant à chaque langue détectée. Puis, nous sélectionnons comme langue cible, celle qui atteint le maximum, et nous considérons, comme langues sources, les autres langues détectées.

3.3.4 Normalisation du dialecte : cas de l'Arabe Marocain

L'utilisation du dialecte tel que l'Arabe Marocain dans les communications écrites augmente encore la complexité des tâches de NLP. Le dialecte est une langue verbale qui n'a pas d'orthographe standard, ce qui conduit les utilisateurs à improviser l'orthographe pendant qu'ils écrivent. Ainsi, pour un même mot, on peut trouver de multiples formes de translittérations. Par conséquent, il est obligatoire de normaliser ces différentes translittérations en une forme de mot canonique.

Pour atteindre cet objectif, nous avons employé des modèles word embedding générés à partir d'un corpus de commentaires YouTube. En outre, en utilisant un dictionnaire de dialecte Arabe Marocain (AM) qui fournit les formes canoniques, nous avons construit un dictionnaire de normalisation que nous appelons ***MANorm***[49]. Nous avons mené plusieurs

[49] MaNorm est disponible sur : //github.com/MAProcessing/MANorm

expériences qui ont démontré l'efficacité de MANorm dans la normalisation des dialectes.

Depuis l'avènement du Short Messaging Service (SMS), l'Arabe Marocain a été introduit dans les communications[50] écrites des utilisateurs et aujourd'hui, dans les médias sociaux, ce phénomène se généralise (Caubet, 2017). Le AM est la langue maternelle de la plupart des Marocains, il a été utilisé dans les SM pour exprimer librement et spontanément des émotions et des pensées avec d'autres pairs (Hall, 2015). Cette langue n'a pas d'orthographe standard puisqu'elle n'est pas utilisée comme langue formelle. Par conséquent, chaque utilisateur de médias sociaux écrit selon sa propre orthographe. La variabilité de l'écriture est due à la diversité de la prononciation des individus en fonction de leurs différents référents, régionaux et culturels (Boukous, 1995). Ainsi, pour un même mot, on trouve des orthographes différentes. Par exemple, le mot "chkoun" (qui), possède cinq autres translittérations différentes ("chkoune", "chkon", "chkone", "chkou", "chkoon"). Ce problème constitue un handicap majeur pour de nombreuses tâches de NLP (Han and Baldwin, 2011). Pour surmonter ce problème, la normalisation peut être utilisée comme un prétraitement préalable à la tâche principale de la NLP. Ce prétraitement a prouvé son efficacité dans l'analyse des sentiments (Htait et al., 2018), l'analyse des dépendances (Van Der Goot et al., 2020) et également dans l'étiquetage POS (Bhat et al., 2018; Van Der Goot and Cetinoglu, 2021).

Pour les langues standards, en général, la tâche de normalisation vise à faire correspondre chaque mot hors vocabulaire à une forme correcte ou standard parmi un ensemble de mots standards candidats (n⟶1). Contrairement au dialecte, qui est une langue non standard, du coup, les mots ne possèdent pas une forme standard. Cette tâche commence par considérer une translittération des phonèmes des mots (les mots sont écrits tels que parlés) comme la forme canonique. Ensuite, nous essayons de capturer toutes ses formes de translittération possibles (1⟶n).

Dans ce travail, nous suivons cette approche pour la normalisation des dialectes AM. Tout d'abord, nous avons construit un dictionnaire de mots AM que nous considérons comme référence des formes canoniques des mots. Ensuite, nous avons construit des modèles de word embedding entraînés sur un corpus de commentaires YouTube en AM. Puis, nous avons exploité ces modèles pour extraire les mots les plus similaires (sémantiquement) de chaque

[50] Jusqu'en 1998, l'écriture en MA n'était pas encore reconnue, à l'exception de quelques essais issus des cours d'alphabétisation et de quelques chansons comme le " Melhoun " et quelques ressources linguistiques (Jaafar, 2012).

entrée du dictionnaire. Finalement, nous avons appliqué des mesures de similarité lexicale pour sélectionner toutes les formes des mots les plus proches de la forme canonique. Le résultat est un dictionnaire de normalisation qui établit une correspondance entre chaque translittération de mot AM et sa forme canonique. Nous appelons MANorm ce dictionnaire de normalisation construit pour l'Arabe Marocain.

3.3.4.1 Formes des mots en dialecte Arabe Marocain

Le dialecte AM est principalement dérivé de l'Arabe (Arabe classique et Arabe standard) à environ 86% et d'un mélange d'autres langues, à savoir le français 11,72%, le tamazight 0,39% et l'espagnol 0,06% (Tachicart et al., 2016). Comme mentionné précédemment, nous nous intéressons à la normalisation du dialecte AM, en particulier le dialecte écrit en alphabet latin (également appelé Arabizi). La première apparition de cette écriture remonte au début des SMS, au début des années 2000, lorsque les téléphones portables n'avaient pas encore de clavier arabe et que certains téléphones ne pouvaient pas afficher les messages écrits en écriture Arabe, alors que l'écriture latine était accessible dans tous les téléphones. Pourtant, jusqu'à ce jour, il y a encore des gens qui conservent ce type d'écriture même si les claviers arabes sont largement disponibles.

Le AM écrit en alphabet Latin est la translittération de phonèmes principalement d'origine Arabe, il s'agit donc d'une forme de parole plus que d'écriture. Cette écriture utilise des consonnes latines qui imitent les consonnes et voyelles arabes comme équivalent des diacritiques.

Dans le but d'identifier les différents phénomènes que la normalisation de dialecte doit couvrir, nous avons sélectionné quelques commentaires depuis notre corpus composé de commentaires YouTube. Nous avons ensuite analysé les formes de mots pour déterminer les sources de variation lexicale où nous avons identifié cinq catégories :

- *Variantes de voyelles pour le même phonème : cela est principalement dû aux différences de prononciation entre les régions. Par exemple, "a" et "e" peuvent être utilisés de manière interchangeable comme dans "bayan" et "bayen" (clair). Nous avons également observé que les voyelles peuvent être omises dans certains cas comme dans "m3alqa" et "m3lqa" (cuillère).*

- *Substitution de lettres par des chiffres : certains chiffres sont utilisés à la place de lettres pour représenter des graphèmes arabes quand leur forme graphique est proche d'une lettre en script Arabe. Par exemple, l'utilisation de "9" au lieu de "ق" [q] et de "7" au lieu de "ح" [ḥ]. Les cas détaillés sont présentés sur le tableau 3.3.*

- *Gémination[51] : est fréquente en Arabe, elle est représentée par des doubles consonnes marquées par certains utilisateurs et négligées par d'autres. Par exemple, "m3allam" (habile) peut s'écrire "m3alam".*

- *Concaténation de mots : les mots de certaines expressions spécifiques sont combinés pour former un seul mot. Par exemple, "hamdo li allah" (Dieu merci) peut s'écrire par exemple "hamdoulillah" ou "hamdollah" ou "hamdouallah".*

- *Agglutination de mots : AM est principalement dérivé de l'Arabe, qui est une langue très flexionnelle ou agglutinante où les affixes sont combinés avec le mot principal. Par exemple, l'expression "wlidatou" (ses enfants) est la concaténation de "wlidat" (enfants) + "ou" (suffixe utilisé pour marquer une possession, équivalent à, les siens). En outre, dans le dialecte AM, l'agglutination est également utilisée pour combiner des particules avec le mot principal, comme dans "fl7ayat" (dans la vie), la lettre "f" (dans) est une préposition concaténée avec "l" (la) un article défini et "7ayat" (vie) un nom.*

Ces caractéristiques linguistiques du dialecte écrit sont la source principale des variations et de la non-uniformité du texte AM. Dans les prochaines sections, nous présenterons notre solution pour normaliser les formes des mots dialectaux et donc augmenter l'uniformité de ce texte.

3.3.4.2 Extraction des données

Les données utilisées ont été recueillies à partir de commentaires vidéo de YouTube extraits à l'aide de l'API de YouTube. YouTube est la plateforme de SM la plus populaire au Maroc, utilisée par 46,39% de la population[52] . Ces vidéos ont été sélectionnées à l'aide de mots-clés liés à divers sujets tels que la politique, le sport, l'art, la cuisine, la comédie et autres. Par exemple, pour la cuisine, nous utilisons des mots clés comme "**مغربية شهيوات**", "**بسطيلة**",

[51] La gémination est le redoublement d'un phonème ou d'une syllabe (le Robert).
[52] https://gs.statcounter.com/social-media-stats/all/morocco (15/06/2021)

"الفيلالي حليمة". L'objectif est de capturer un large éventail de mots de différents domaines et d'assurer ainsi une large couverture du vocabulaire de l'Arabe Marocain. Le corpus collecté contient environ 500K phrases.

3.3.4.3 Prétraitement des données

Avant de commencer les étapes de normalisation, nous avons effectué un ensemble de prétraitements pour préparer notre corpus à la génération des modèles word embedding. Le premier consiste à sélectionner les commentaires en caractères latins, parce que le corpus brut contient également des commentaires en caractères arabes. Puis, les commentaires dupliqués ou ne contenant que des chiffres ou un seul mot sont supprimés et toute répétition de plus de trois lettres est ramenée à deux lettres. En outre, nous avons supprimé toutes les ponctuations, les hashtags, les URL, les mentions at, les émoticônes, les symboles et les séquences de chiffres complètes. Nous avons également remplacé les chiffres à l'intérieur des mots par leurs équivalentes lettres pour correspondre à celles utilisées dans le dictionnaire des dialectes (voir tableau 3.3). Sauf pour "7" (équivalent de "ح") et "3" (équivalent de "ع") qui n'ont pas d'équivalents dans l'alphabet latin. Finalement, nous avons réduit en minuscule l'intégralité du texte. Après ces prétraitements, le corpus obtenu devient composé de 160 651 phrases et de 242 277 mots uniques.

Tableau 3.3. Règles de conversion des chiffres en lettres de l'alphabet latin

Chiffres / Lettres	Lettres équivalentes
2	a
6	t
4 / 8	gh
5 / x	kh
9	q

3.3.4.4 Normalisation des dialectes

Notre système de normalisation de dialecte est basé sur deux étapes. A savoir, l'extraction des translittérations et leurs sélections, en partant de trois modèles de word embedding et d'un dictionnaire de dialecte AM qui servira de lexique pour la forme canonique des mots. Pour chaque mot canonique de ce dictionnaire, nous extrayons d'abord les mots les plus

similaires sémantiquement du vocabulaire produit par les modèles de word embedding. La similarité sémantique vise à sélectionner les voisins les plus proches de chaque mot canonique en fonction de leur contexte. Ensuite, à partir de ces mots triés, nous sélectionnons les mots les plus similaires lexicalement au mot canonique. La similarité lexicale vise à sélectionner les translittérations les plus similaires au mot canonique en termes de forme de surface. Nous décrirons en détail ces étapes de traitement dans les sections suivantes.

3.3.4.5 Dictionnaire des dialectes

La forme canonique envisagée pour notre tâche de normalisation était d'abord basée sur une collection de dictionnaires de noms, verbes et adjectifs en dialecte AM (Jaafar, 2012). Ce dictionnaire comprend 14548 entrées que nous avons converties à partir d'une transcription IPA (International Phonetic Alphabet) adaptée pour correspondre au script latin utilisé dans les médias sociaux (les différentes conventions adoptées sont énumérées dans le tableau A.1 de l'Annexe A). Ensuite, nous avons étendu ce dictionnaire pour inclure certains mots spéciaux aux médias sociaux (par exemple "tagini" (tag me)). Cependant, en vérifiant le dictionnaire collecté, nous avons constaté que seulement 16% de ses mots sont présents dans le vocabulaire des modèles d'embedding. En fait, actuellement, la majorité de ces mots sont rarement employés dans les médias sociaux régis par la jeune génération. Pour surmonter ce problème, nous avons collecté de manière semi-automatique un ensemble de mots à partir du vocabulaire généré par les modèles d'embedding. Tout en nous concentrant sur les mots utiles qui peuvent capturer d'autres translittérations. Nous mentionnons que nous considérons les mots empruntés à d'autres langues comme de la néologie et nous les incluons comme nouvelles entrées dans le dictionnaire MA. Par exemple, le mot "stationi" (garer) est emprunté au français "stationner". Le dictionnaire final est de taille 2502 mots canoniques.

3.3.4.6 Génération des modèles Word Embedding

Nous utilisons trois modèles de word embedding, à savoir Word2Vec CBOW (sac de mots continu) et Skip-gram et le troisième est FastText. Selon Mikolov et al., (2013), les deux architectures CBOW et Skip-gram fonctionnent bien dans les tâches sémantiques. En outre, Skip-gram est efficace lorsqu'il s'agit de présenter des mots rares, contrairement à CBOW qui convient plus aux mots fréquents. FastText est également convenable à la modélisation des mots rares. Par conséquent, une combinaison de ces modèles permettra d'étendre la couverture des différents types de mots (rares et fréquents). Alors, si nous combinons leurs

résultats, nous pourrons capturer plus de translittérations pour chaque mot canonique. Quant à la configuration de ces modèles, nous avons fixé le nombre minimum d'occurrence de chaque mot dans le corpus à deux occurrences. L'objectif est de capturer les formes de mots rares. Et comme taille pour la fenêtre de contexte, nous avons choisi sept mots des deux côtés, gauche et droite. Ce choix est influencé par la taille courte des phrases composant les commentaires YouTube.

3.3.4.7 Mapping entre forme normalisée et translittération

Le processus de normalisation (Figure 3.3) comprend trois étapes, il commence par une phase de similarité sémantique suivie par une similarité lexicale, pour finir avec une combinaison des sorties des trois modèles word embedding.

a. Similarité sémantique

Dans la première étape, le système extrait les voisins les plus similaires de chaque mot du dictionnaire AM (forme canonique). Cela est fait sur la base du score de similarité sémantique, calculé à travers la distance cosinus entre les vecteurs dimensionnels de chaque mot du modèle et chaque mot canonique. Nous avons utilisé la classe most_similar du framework Gensim, dont la fonction est d'extraire la liste des mots similaires. Nous précisons que nous avons fixé le paramètre taille de la liste à vingt. En effet, nous avons constaté que les valeurs inférieures capturent peu de mots et que les valeurs supérieures capturent beaucoup de bruit. La même constatation a été faite par Htait et al., (2018).

b. Similarité lexicale

La deuxième étape de ce processus est la similarité lexicale, qui vise à extraire les mots les plus proches du mot canonique (fourni par le dictionnaire MA) selon sa forme de surface :

- *D'abord, nous mesurons la similarité lexicale entre chaque mot canonique et l'ensemble des mots extraits lors de l'étape précédente.*

- *Ensuite, nous sélectionnons les mots qui ont un score de similarité supérieur à une valeur seuil.*

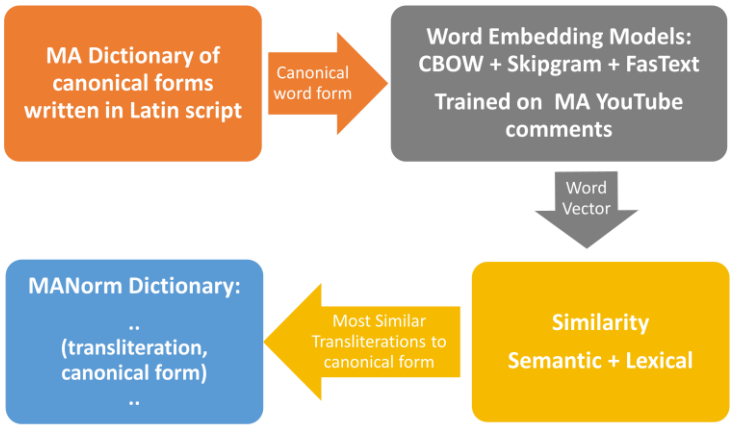

Figure 3.3. Processus de normalisation du dialecte AM

Cette valeur seuil a été définie empiriquement et nous l'avons fixée à 70%. Après plusieurs expériences (plus de détails seront donnés dans la section d'évaluation), nous avons observé que plus le seuil est faible, plus la couverture est grande, et par conséquent, de nombreux mots indésirables seront sélectionnés. Cependant, des valeurs seuils plus élevées éliminent une partie considérable des translittérations de mots, particulièrement dans le cas d'agglutination de mots. Par exemple, il est impossible de capturer et normaliser "lmagrib" (Maroc) en "maghrib" si le seuil est supérieur à 70%.

Le calcul du score de similarité lexicale repose sur différentes mesures. La première est basée sur le matching des séquences avec la suppression des voyelles et des doubles consonnes. Nous avons supprimé les voyelles des mots parce que, comme nous l'avons mentionné, la plupart des variations d'écriture sont liées à l'utilisation de voyelles différentes pour le même mot. Par exemple, le mot "ya3tek" (te donner) peut être écrit de 13 manières différentes : ya3tek, ye3tek, yaatik, ya3tik, yatek, yaatek, yaetik, y3atik, ya3tike, ytik, yi3tik, ya3atik, ye3tik. Par conséquent, en supprimant les voyelles et en conservant les consonnes, nous pourrons saisir une grande partie des translittérations. De plus, nous avons réduit à une, toutes les doubles consonnes consécutives servant à marquer la gémination, parce que leur utilisation n'est pas toujours respectée. A titre d'exemple, le mot "allah" (dieu) peut être écrit "alah" par certains utilisateurs.

La deuxième mesure utilisée pour la similarité lexicale est le matching des séquences avec le Soundex adapté aux règles phonétiques du dialecte AM, comme c'est montré sur le Tableau 3.4.

Tableau 3.4. Règles phonétiques MA utilisées pour le Soundex

Phonèmes	**Conversion**
b, f, m, p, v, w	1
d, t, l, n	2
s, z	3
j, y, ch	4
r, kh, gh	5

Les dernières mesures utilisées sont celles employées dans des travaux connexes. Notre objectif est de montrer l'efficacité de chacune des mesures à capturer les différentes translittérations d'un mot. Dans le Tableau 3.5, nous énumérons les différentes formules adoptées pour les fonctions de scoring dans ces études précédentes. Les performances de ces mesures seront données dans la section d'évaluation.

c. Combinaison des modèles

Finalement, nous avons exécuté le même processus de normalisation pour les trois modèles d'embedding. En analysant les résultats obtenus, nous avons bien constaté que chaque modèle peut capturer un ensemble de translittérations différentes. Ce qui confirme que ces modèles ont des portées différentes. Par conséquent, nous avons décidé de fusionner le lexique produit dans chaque cas et donc bénéficier d'une couverture plus large des translitérations. Les combinaisons résultantes (translittération, forme normalisée) sont ensuite regroupées pour former un seul dictionnaire de normalisation MANorm.

Tableau 3.5. Fonctions de scoring des similarités sémantiques et lexicales utilisées dans les approches de normalisation basées sur le word embedding

Approches	**La similarité sémantique**	Similarité **lexicale**
Sridhar (Sridhar, 2015)	Similarité cosinus $= \frac{\sum_{i=0}^{D} u_i \times v_i}{\sqrt{\sum_{i=0}^{D}(u_i)^2 \times \sum_{i=0}^{D}(v_i)^2}}$	$\textbf{lexical similarity}(s_1, s_2) = \frac{\textbf{LCSR}(s_1, s_2)}{ED(s_1, s_2)}$ $\textbf{LCSR}(s_1, s_2) = \frac{\textbf{LCS}(s_1, s_2)}{Max\ Lenght(s_1, s_2)}$ LCSR = Longest Common Subsequence Ratio (Melamed, 1995) LCS = Longest Common Subsequence ED[53] = Distance d'édition entre deux chaînes de caractères.
Bertaglia (Bertaglia and Nunes, 2016)	Similarité cosinus (même formule)	$\textbf{lexical similarity}(s_1, s_2) = \begin{cases} \frac{\textbf{LCSR}(s_1, s_2)}{MED(s_1, s_2)}, & if\ MED(s_1, s_2) > 0 \\ \textbf{LCSR}(s_1, s_2), & \textbf{otherwise} \end{cases}$ $\textbf{LCSR}(s_1, s_2) = \frac{\textbf{LCS}(s_1, s_2) + \textbf{DS}(s_1, s_2)}{Max\ Lenght(s_1, s_2)}$ $MED(s_1,s_2) = ED(s_1,s_2) - DS(s_1,s_2)$ MED = Distance d'édition modifiée DS = Symétrie diacritique entre s_1 et s_2 N.B : Comme les diacritiques n'existent pas dans notre cas donc, DS = 0 et la formule de similarité lexicale sera la même que celle de Sridhar.
Htait[54] (Htait et al., 2018)	Similarité cosinus (même formule)	Matching de séquences avec un score de 50%

3.3.5 Transfert

Une fois l'étape d'analyse est accomplie, nous disposerons donc des langues des différents mots, de leurs caractéristiques morphologiques, et aussi de la langue cible de traduction. Nous pouvons alors, passer au deuxième traitement dans le pipeline de MT qui est le transfert.

[53] Pour l'anglais, le calcul de la distance d'édition a été modifié pour trouver la distance entre le squelette consonantique des deux chaînes s1 et s2.

[54] Notre implémentation était basée initialement sur le code ouvert fourni par Htait : https://github.com/OpenEdition/NormAFE

L'objectif de cette phase est de chercher la bonne traduction du mot source dans la langue cible, en procédant à une sélection lexicale. Afin d'effectuer cette sélection tout en préservant le sens des mots pendant la traduction, nous avons procédé comme suit :

1. Nous avons adopté une approche basée sur les connaissances utilisant des dictionnaires sémantiques bilingues et des corpus spécifiques.

2. En plus de ces ressources, nous avons introduit le processus de désambiguïsation du sens des mots (WSD), qui a montré son efficacité dans des tâches similaires.

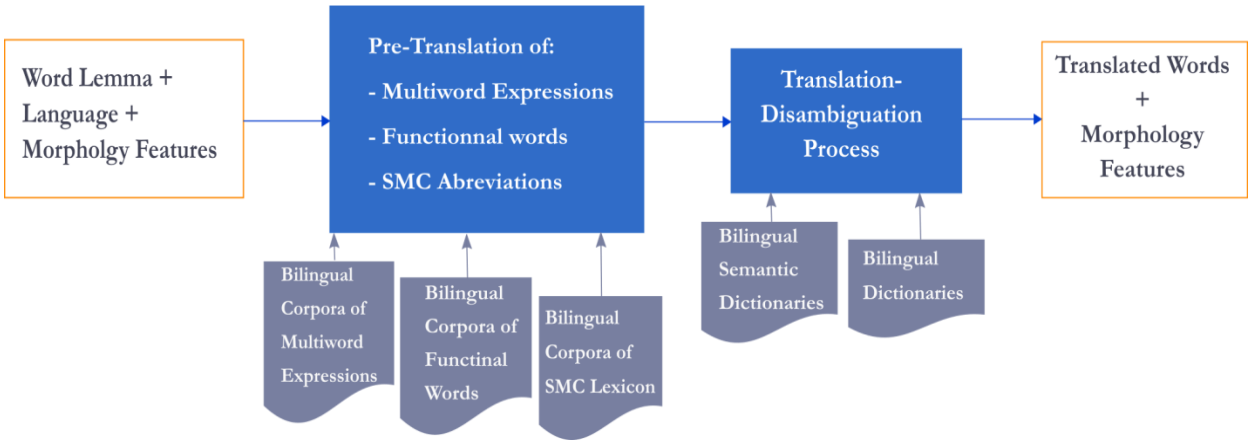

Figure 3.4. Étapes du processus de transfert

Le processus de transfert comprend donc, deux traitements en un, à savoir la traduction et la WSD. Chaque mot sera traduit et désambiguïsé dans son contexte car le *seul moyen d'identifier la signification d'un mot polysémique* est à travers son contexte (Ide and Véronis, 1998). L'approche adoptée pour la désambiguïsation est basée sur la connaissance, où nous utilisons des ressources lexicales en plus de l'algorithme **Lesk adapté** pour le scoring des sens, ainsi que les propriétés du discours (voir section 3.3.5.2). Grâce à ces propriétés, nous avons pu exploiter un nouveau contexte que nous avons appelé **contexte *vertical multilingue*** *ou* ***Multilingual Vertical Context (MVC)***. Les étapes du transfert sont illustrées sur la Figure 3.4.

3.3.5.1 Désambiguïsation du sens des mots (WSD)

La WSD vise à identifier le sens d'un mot polysémique, en se servant des mots de voisinage qui constituent son contexte. Dans les systèmes de traduction automatique, la WSD

s'avère primordiale pour assurer la qualité de la traduction. Dans notre cas, la WSD est utilisée pour identifier le sens approprié d'un mot, puis sa traduction appropriée afin de préserver la sémantique des phrases lors du transfert.

L'objectif de notre solution est de couvrir toutes les différentes langues intégrées dans les phrases CS, que ça soit langues standards ou dialectes. Comme c'est connu, la plupart des ressources lexicales ne sont disponibles que pour quelques langues, notamment l'Anglais. Tandis que les langues moins dotées souffrent de la pénurie de données. Ce qui constitue une limite face à l'application de certaines techniques de WSD. En effet, l'absence de corpus appropriés nous a empêché d'adopter des approches statistiques. Ainsi que, l'absence d'une hiérarchie structurée de connaissances (c'est-à-dire d'ontologie), nous a empêché d'appliquer les approches structurelles.

Vu ces contraintes, nous avons adopté une approche basée sur la connaissance pour la tâche de WSD, qui est l'algorithme *Adapted Lesk*. Introduit à l'origine par Lesk (Lesk, 1986), dans sa version originale, Lesk affirmait que le sens du mot peut être identifié en comparant les définitions ou les glosses[55] de chacun de ses sens, avec ceux de tous les mots environnants (contexte) en utilisant des dictionnaires (Machine Readable Dictionary). Le sens partageant le plus grand nombre de mots avec les glosses des mots de contexte sera attribué à ce mot.

Depuis son apparition jusqu'à aujourd'hui, plusieurs extensions de cette technique ont été proposées pour améliorer les performances de la WSD, telles que Simple (Kilgarriff and Rosenzweig, 2000), Adapté (Banerjee and Pedersen, 2002), Amélioré (Basile et al., 2014) et d'autres.

Adapted Lesk a beaucoup profité de WordNet (Miller et al., 1990), en élargissant la comparaison des sens avec d'autres relations sémantiques fournies par WordNet (hyperonyme, hyponyme, méronyme...). Ce qui peut améliorer la précision de la désambiguïsation. Alors que la technique de Lesk était basée sur les dictionnaires monolingues traditionnels, où la comparaison des sens se faisait uniquement entre les glosses. Ce qui influence négativement l'identification des sens. Adapted Lesk compare les glosses,

[55] Les glosses sont les définitions fournies pour un terme dans un dictionnaire.

l'hyperonyme[56], l'hyponyme[57], l'holonyme[58], le méronyme[59], le troponyme[60] et les attributs de chaque mot cible avec ceux de ses mots de contexte. Ensuite, il calcule le score de *relation* (Relatedness score) pour sélectionner le sens ayant le score le plus élevé.

Dans notre algorithme, nous avons gardé le même principe que celui de Adapted Lesk. Mais, étant donné que nous effectuons une tâche de traduction, nous avons recouru donc à des dictionnaires bilingues au lieu de dictionnaire monolingue utilisé en principe avec cet algorithme. Quant à la pondération ou la mesure de relation entre mots, nous avons utilisé la formule de score de recouvrement donnée dans Extended Gloss Overlaps (Banerjee and Pedersen, 2003). Il est à noter que la métrique choisie n'affecte pas le résultat de la WSD, comme il a été prouvé par (Vasilescu et al., 2004). Les détails des techniques suivies seront présentés dans la section Méthode, alors que le choix des mots du contexte et sa relation avec les propriétés du discours sera expliqué ci-après.

3.3.5.2 Propriétés du discours et contexte vertical multilingue

Le contexte se compose des mots environnants du mot cible (le mot à traduire) ou selon Navigli, (Navigli, 2009) : *"context(w) is the bag of all content words in a context window around the target word w."*. Qui veut dire qu'un contexte(w) est le sac de tous les mots du contenu [61]dans une fenêtre de contexte autour du mot cible w". Le mot et son contexte ont fait l'objet de plusieurs études qui indiquent que le mot cible et ses mots voisins construisent conjointement la même signification. C'est ce que signifie la propriété du discours "*one sense per collocation*", ou un sens par collocation, présentée par Yarowsky(Yarowsky, 1993).

Dans la même perspective, nous nous sommes intéressés aux mots de contexte et nous avons essayé de les employer efficacement pour mieux identifier la signification de chaque

[56] L'hyperonyme est une catégorie générale regroupant des sous-catégories.
[57] L'hyponyme est le contraire de l'hyperonyme, c'est une sous-catégorie appartenant à une catégorie supérieure plus générique. Par exemple, les lions, les tigres et les pumas sont tous des hyponymes de la catégorie des félins (qui est donc leur hypéronyme dans cet exemple). Ensuite, les félins sont un hyponyme de l'hypéronyme "Les animaux".
[58] L'holonyme est un mot qui vérifie une relation partitive hiérarchisée avec un autre mot. Par exemple, corps est un holonyme de bras et maison est un holonyme de toit.
[59] Le méronyme est le contraire de l'holonyme, c'est un terme qui désigne une partie d'un second terme. Par exemple, bras est un méronyme de corps, de même que toit est un méronyme de maison.
[60] Le troponyme c'est un terme qui marque la présence d'une relation "manière" entre deux verbes. Par exemple, grignoter est troponyme de manger.
[61] Les mots de contenu (content words) sont les mots qui portent une signification. Ce sont les mots qui constituent les classes POS : noms, verbes, adjectifs et adverbes. Leur contraire sont les mots de fonction (particules) qui sont employé pour lier entre les mots de contenu.

mot cible. Pour cela, nous avons utilisé un contexte particulier que nous avons appelé contexte *vertical multilingue* (MVC). Nous avons qualifié ce contexte par multilingue, puisqu' il contient des mots de différentes langues. En effet, dans ce contexte, nous avons gardé aussi les mots qui ne seront pas traduits (les mots dont la langue est celle cible), parce que leur sens pourra être utile à l'identification du sens des mots cibles. Ainsi, si nous les excluons du contexte, nous risquons de perdre certains des indicateurs significatifs de sens. Finalement, puisque nous faisons l'extraction du contexte depuis toutes les occurrences du mot cible, non seulement au niveau de la phrase, mais aussi au niveau du document (conversation). Nous appelons alors, ce contexte "contexte vertical" (Figure 3.5).

Ce contexte élargi ne peut pas générer une perte du sens du mot. Ce sens est toujours préservé pendant la même discussion selon l'heuristique "un sens par discours" affirmée par Gal (Gale et al., 1992) : *"a word is consistently referred with the same sense within any given discourse or document."*. Ce qui veut dire qu'un mot est constamment référencé par le même sens dans un discours ou un document donné.

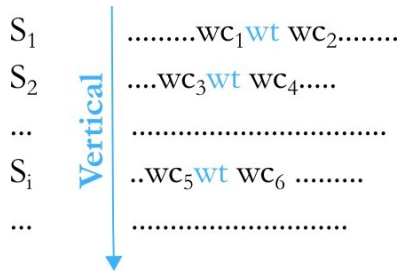

Figure 3.5. Illustration du contexte vertical multilingue

La Figure 3.5 montre une illustration du contexte vertical multilingue où, $\mathbf{S_i}$ représente une phrase du texte, **wt** le mot cible et $\mathbf{wc_i}$ est un mot de contexte. Ce contexte forme un ensemble ***Wc*** composé de tous les mots du contexte $\mathbf{wc_i}$, ainsi :

Wc = {wc1, wc2, wc3, wc4, wc5 ...}.

Les deux phrases suivantes représentent un exemple d'un contexte vertical multilingue :

- *il ne veut pas un **healthy diet** définitivement.*
- *il doit suivre son **diet** alimentaire.*

Où w_t = diet (anglais) et W_c = {healthy (anglais), *définitivement (*français), suivre (français), alimentaire (français)}.

En conclusion, à travers l'utilisation du MVC, nous élargissons le contexte d'un mot cible. Ce qui sera très utile pour déterminer le sens le plus proche de ce mot et donc préserver sa sémantique pendant la traduction.

Il est à noter qu'une idée proche de la nôtre a été présentée comme un algorithme pour la WSD basé sur les Field Association Schemes[62] ou les schémas d'association de champs (Wang et al., 2011) pour résoudre l'ambiguïté du contexte. Toutefois, la solution dépend de la construction des schémas et de leur mise à jour. Cela rend la solution lourde et difficile à maintenir, pour les textes CS en particulier où différentes langues sont mixées dans la même phrase.

3.3.5.3 Méthode.

Etant donné que la Langue Matrice est la langue dominante qui conduit le sens de la phrase CS, alors, la traduction sera faite à partir de mots étrangers vers cette langue cible.

Tout d'abord, nous commençons par une phase de pré-traduction afin de traduire directement un ensemble d'expressions multi-mots[63] et aussi les abréviations. Ainsi, nous pouvons éviter la traduction incorrecte de ces expressions. À cet effet, nous remplaçons d'abord, les mots spéciaux ou les séquences de mots utilisés dans les SM par leurs traductions équivalentes. Nous nous servons pour cela d'un lexique d'abréviations de réseaux sociaux. À tire d'exemple, l'abréviation OMG, qui signifie en anglais : Oh My God (en français : oh mon dieu) sera remplacée par sa signification dans la langue cible. Le même traitement est effectué pour les expressions multi-mots. Nous rappelons que, toutes ces traductions sont effectuées directement à l'aide de corpus bilingues pour toutes les langues standards et les dialectes utilisés. Concernant les mots fonctionnels (articles, prépositions, conjonctions, pronoms...), nous les traduisons en utilisant un traducteur open source basé sur les règles.

Ensuite, pendant le processus de traduction-désambiguïsation, nous localisons d'abord les mots étrangers à traduire. Puis, nous extrayons leur contexte à partir des mots environnants depuis toutes les occurrences. Nous précisons que, nous avons choisi un

[62] Le schéma d'association de champs est une technique structurelle utilisée pour la désambiguïsation de contexte. Il part du principe que la signification des mots est liée à des sens globaux, tels que des champs ou des domaines. Elle tente de construire une connaissance à base d'association de champs composée de termes (mots de contexte) et de leurs champs spécifiques associés.
[63] Les expressions multi-mots sont les mots composés et les idiomes.

contexte court avec une fenêtre de ±1 mots autour du mot cible (un mot de la droite et un autre de la gauche). Ce contexte court est étendu avec le contexte vertical, comme mentionné auparavant. Par la suite, nous collectons les différentes traductions du mot cible à partir d'un dictionnaire sémantique bilingue.

Concernant les mots du contexte, nous collectons leurs traductions ou synonymes suivant leurs langues sources (algorithme Annexe B). Puis, nous calculons les scores de relation entre les glosses, les exemples, les hyperonymes et les hyponymes de chaque mot cible et ceux de chaque mot du contexte selon l'algorithme Extended Gloss Overlaps (Banerjee and Pedersen, 2003). Enfin, nous sélectionnons la traduction qui atteint le score maximum. La Figure 3.6 illustre comment nous procédons au recouvrement.

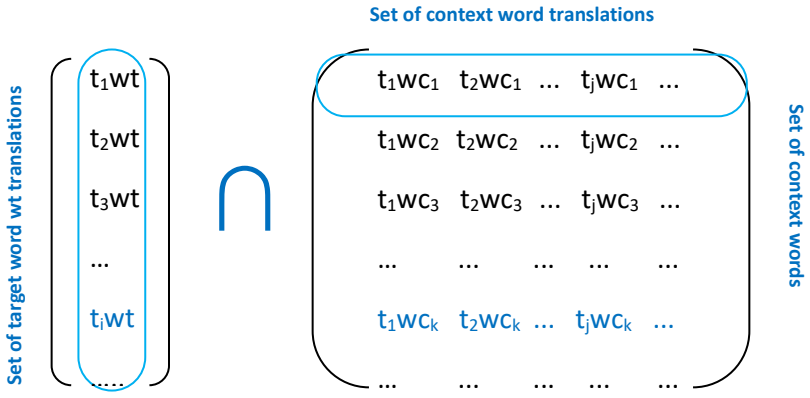

Figure 3.6. Recouvrement entre les traductions de mots cibles et les traductions de mots de contexte

Comme le montre cette figure, les t_iwt (à gauche) sont les différentes traductions du mot cible *wt*. Tandis que, les t_jwc_k (à droite) sont les différentes traductions de chaque mot de contexte wc_k. Précisément, les colonnes correspondent aux mots du contexte wc_k et les lignes correspondent aux différentes traductions de chaque mot du contexte. L'intersection représente le recouvrement entre l'ensemble des traductions de *wt* et l'ensemble des traductions des mots de son contexte.

Il est à noter qu'afin d'assurer une meilleure performance, nous avons limité les candidats à la traduction aux N traductions les plus fréquentes (Most Frequent Translations, MFTs). Ce choix sera détaillé dans la section d'évaluation.

Le score de recouvrement entre chaque traduction du mot cible t_iwt et les traductions de

tous les mots de son contexte t_jwc_k est calculé comme suit :

$$TransScoret_iwt = \sum_{k=1}^{|Wc|} \sum_{j=1}^{N} Relatedness\left(t_iwt, t_jwc_k\right) \quad (3.1)$$

Avec 1≤ i, j ≤ N

où, W_c est l'ensemble des mots de contexte. La traduction sélectionnée t_iwt est celle qui maximise ce score.

La formule de relation que nous avons utilisée (formule 3.2) est la même que celle définie dans Banerjee and Pedersen (Banerjee and Pedersen, 2003). En outre, afin d'enrichir davantage la comparaison, nous ajoutons des exemples de sens :

$$\begin{aligned} Relatedness(twt, twc) \quad (3.2)\\ &= score\left(gloss(twt), gloss(twc)\right) \\ &+ score\left(hyp(twt), hyp(twc)\right) + score(hypo(twt), hypo(twc)) \\ &+ score(hyp(twt), gloss(twc)) + score(gloss(twt), hyp(twc)) \\ &+ score(gloss(twt), \exp(twc)) \\ &+ score(\exp(twt), gloss(twc)) \\ &+ score(\exp(twt), \exp(twc)) \end{aligned}$$

où, *score* représente le nombre de recouvrements entre chaque paire utilisée. Nous notons qu'avant de commencer le calcul des scores, nous avons supprimé les mots fonctionnels puisqu'ils ne portent aucune information sémantique. De plus, pour améliorer encore le résultat de la désambiguïsation, nous avons sélectionné les traductions en fonction de l'étiquette "POS" (nom, verbe, adjectif et adverbe). Cela est justifié d'une part, par le fait que cet étiquetage des parties de discours constitue la première étape de la WSD, comme il a été montré par (Wilks and Stevenson, 1997). D'autre part, en appliquant ce type de filtre, nous pourrons réduire le nombre de comparaisons et donc l'effort computationnel.

Notons qu'en cas de score nul, c'est à dire qu'il n'a pas de recouvrement entre les mots, nous choisissons comme traduction celle la plus fréquente (MFT) donnée par le dictionnaire

sémantique selon le POS tag du mot cible. Ce sens prédominant ou, dans notre cas, traduction est largement utilisé dans la tâche WSD comme un back-off, lorsque les informations contextuelles ne sont pas en mesure d'apporter le sens approprié. Cette procédure peut donner de bons résultats, ce qui a été confirmé par : 'the first sense heuristic is so powerful' (Mccarthy et al., 2007). Les détails de l'algorithme sont donnés dans Annexe B.

3.3.6 Génération et réordonnancement

La phase de transfert est suivie de la phase de *génération*, qui vise à reconstruire la forme de surface du mot (par exemple la conjugaison des verbes, la forme plurielle...) après avoir été traduite sous la forme lemme. Le traitement suivant est le *réordonnancement* qui vise à mettre les mots traduits dans l'ordre de la langue cible.

Notre objectif dans cette étude est la normalisation de textes bruités, afin d'être utilisés par d'autres applications de Text Mining. Autrement dit, nous ne sommes pas intéressés par un système de traduction complet dédié à l'usage humain. C'est pourquoi, pour des applications basées sur les corpus, nous pouvons nous satisfaire de la forme lemme sans génération ni réordonnancement. En outre, notre choix de la langue matrice, qui régit la structure de la phrase comme langue cible, nous facilite la tâche de réordonnancement en le limitant au niveau des mots étrangers. En d'autres termes, nous aurons besoin seulement d'un réordonnancement local et non global au niveau phrase qui est une tâche plus complexe. Par conséquent, nous n'avons pas accordé beaucoup d'intérêt à cette étape en termes de précision et d'exactitude. Dans ce traitement, nous avons exploité et adapté certains outils existants provenant de traducteurs à base de règles et de statistiques disponibles en sources ouvertes.

3.3.6.1 Génération.

La génération constitue un défi pour de nombreux traitements du langage naturel tels que la traduction automatique. Particulièrement, lorsqu'il s'agit de traduire des langues moins infléchies vers des langues plus infléchies (par exemple de l'anglais vers le français) ou inversement.

Dans ce travail, nous avons effectué la génération de la morphologie des mots traduits par l'utilisation de leurs lemmes et de leurs caractéristiques morphologiques extraites lors de la phase d'analyse. Pour ce faire, nous utilisons le module de génération à base de règles

incorporées dans Apertium translator (Tyers et al., 2010) compilé avec un dictionnaire monolingue français issu d'une paire de langues existante. Avec ce générateur, il est possible de générer la forme infléchie correcte des mots si nous disposons des caractéristiques correctes. Ce qui explique sa dépendance de la phase d'analyse. Enfin, nous avons utilisé le module post-générateur qui gère les opérations orthographiques (par exemple les apostrophes).

Il faut mentionner que, dans le cas où la génération est nécessaire, les mots appartenant à la langue cible ne seront pas régénérés, nous les gardons inchangés. Nous utilisons leurs lemmes et leurs POS juste en phase de transfert. Cela signifie que seuls les mots cibles traduits ou les mots étrangers seront générés, ce qui permet de mieux préserver le sens de la phrase. Sinon (si la génération n'est pas nécessaire), nous conservons l'ensemble des phrases sous forme de lemme.

3.3.6.2 Réordonnancement

La complexité de la tâche de réordonnancement dépend du degré de divergence des structures de la langue source et celle de la langue cible. Par exemple, l'Anglais et le Français suivent l'ordre SVO (Subject Verb Object), l'Arabe VSO et le Japonais SOV.

Dans la MT basée sur des règles, le réordonnancement est basé sur l'alignement syntaxique entre les structures de la langue source et celle de la langue cible par l'utilisation de règles (par exemple Apertium). Pour la SMT, elle peut être effectuée avant la traduction (pré- réordonnancement) soit par l'inclusion d'une analyse syntaxique, soit pendant le processus de décodage, soit après la traduction (post- réordonnancement) (Bisazza and Federico, 2016). Du moment que nous sommes limités par la structure complexe du Code Switching (difficulté d'effectuer une analyse syntaxique) et l'absence de corpus parallèles, la seule solution possible, pour notre cas, est un post- réordonnancement statistique.

La technique employée pour le post- réordonnancement est basée sur un modèle de langage de 3 grammes entraîné sur un corpus de langue cible. Comme mentionné plus haut, nous faisons un ordonnancement superficiel, car nous devons remettre en ordre uniquement les mots traduits. En d'autres termes, nous effectuons un réordonnancement local des mots cibles. En effet, la structure globale de la phrase est maintenue par la langue matrice, de sorte qu'il n'est pas nécessaire de procéder à un réordonnancement global. Pour cette étape, nous

extrayons d'abord le mot cible avec les deux mots qui l'entourent pour former une séquence de trois mots. Ensuite, nous générons les différents ordres de mots de cette séquence. Puis, le modèle de langage nous fournit les probabilités des différentes combinaisons des trois mots. Finalement, nous sélectionnons la combinaison qui maximise cette probabilité, puisque la séquence la plus probable est susceptible d'être grammaticalement correcte.

3.3.6.3 Exemple d'étapes de traitement de Machine Normalization

Tableau 3.6. Étapes de traitement de phrases CS par le système MN

Etapes de traitement	Sortie
Entrée	Il veut pas un **healty diet** définitivement il dit que **li w9a3 w9a3**. **Daba** il est devenu gourmand, il doit suivre son **diet** alimentaire.
Analyses	^il/FR<prn><tn><p3><m><sg>$ vouloir/FR<vblex><pri><p3><sg>$ ^pas/FR<adv>$ ^un/FR<det><ind><m><sg>$ ^**healthy**/EN<adj><sint>$ ^**diet**/EN<n><sg>$ ^définitivement/FR<adv>$ ^il/FR<prn><tn><p3><m><sg>$ ^dire/FR<vblex><pri><p3><sg>$ ^que/FR<cnjsub>$ ^**li**/MA<prn> ^**w9a3**/MA<vb>$ ^**w9a3**/MA<vb>$ ^.<sent>$ ^**Daba**/MA<pr>$ ^il<prn><tn><p3><m><sg>$ ^être/FR<vbser><pri><p3><sg>$ ^devenir/FR<vblex><pp><m><sg>$ ^gourmand/FR<adj><m><sg>$ ^,<cm>$ ^il/FR<prn><tn><p3><m><sg>$ ^devoir/FR<vblex><pri><p3><sg>$ ^suivre/FR<vblex><inf>$ ^son/FR<det><pos><m><sg>$ ^**diet**/EN<n><sg>$ ^alimentaire/FR<adj><mf><sg>$ ^.<sent>$
Transfert	Il vouloir pas un **sain régime alimentaire** définitivement il dire que qui arriver arriver. Maintenant il est devenir gourmand, il devoir suivre son **régime alimentaire** alimentaire.
Génération	Il vouloir pas un **sain régime alimentaire** définitivement il dire que qui arriver arriver. Maintenant il est devenir gourmand, il devoir suivre son **régime alimentaire** alimentaire.

EN, English; FR, French; MA, Moroccan Arabic.
Les mots en gras représentent les mots à traduire et leurs traductions si disponibles.

Dans le Tableau 3.6, nous donnons un exemple d'étapes de traitement pour normaliser des phrases CS. Avec la présence de mots dialectaux (*daba*, *li* et *w9a3*) en Arabe Marocain (AM). En plus de mots appartenant aux langues standards, à savoir l'Anglais (EN) pour *diet*

et *healthy,* et le Français (FR) pour le reste des mots. La langue cible étant le français, la traduction est effectuée de l'AM et de l'EN vers le FR. Dans cet exemple, nous ne générons pas les mots cibles traduits, nous les conservons sous la forme de lemme, ce qui est très suffisant dans de nombreuses applications de traitement de texte.

3.4 Evaluation

Dans cette section, nous menons deux évaluations : la première concerne l'approche de la normalisation du dialecte, et la deuxième s'intéresse à l'approche de la normalisation du CS.

3.4.1 Évaluation de la normalisation du dialecte

L'objectif de cette section est d'évaluer le dictionnaire MANorm en testant la qualité du lexique de normalisation. A notre connaissance, il s'agit du premier travail sur la normalisation du dialecte AM en script Latin, par conséquent, nous ne disposons pas de référence pour évaluer la performance de notre système. Il est donc impossible de procéder à une évaluation automatique, c'est pourquoi nous avons créé notre propre référence. Nous avons d'abord combiné les dictionnaires de sortie des trois modèles, ce qui a produit un dictionnaire de normalisation de 3057 entrées (translittération, forme normalisée). Nous avons ensuite, validé chaque entrée, en vérifiant manuellement la correspondance entre chaque translittération et sa forme canonique. Cette opération a donné lieu à un dictionnaire de référence de 2225 entrées correctes. Des exemples d'entrées de ce dictionnaire sont donnés dans le Tableau 3.7.

Tableau 3.7. Exemples de résultats de normalisation

Forme canonique des mots (dictionnaire de dialectes AM)	**Translittérations (corpus)**
awel (le premier)	awl, awwal, awle, aaal, awale
choukran (merci)	chokran, chokrane, chkran, khokran, chokrn

3.4.1.1 Métriques d'évaluation

Nous avons utilisé deux métriques pour l'évaluation du dictionnaire : la précision et la

couverture, plutôt que les métriques courantes de la recherche d'informations (précision, rappel et FScore). En effet, nous sommes incapables de mesurer le rappel, qui est le rapport entre le nombre de résultats pertinents actuellement fournis et le total des résultats pertinents que nous devons fournir, puisque nous ne disposons pas du nombre exact des translittérations que nous pouvons normaliser par chaque forme canonique d'un mot donné. Donc nous ne pouvons pas définir la valeur totale des résultats pertinents. C'est pourquoi, au lieu du rappel, nous mesurons la couverture des mots du dictionnaire AM. La couverture représente le rapport entre les mots canoniques utiles ou les mots qui ont été capables d'attraper d'autres translittérations, et le total des mots du dictionnaire AM :

$$la\ couverture = \frac{le\ nombre\ des\ mots\ canoniques\ utiles}{la\ taille\ du\ dictionnaire\ AM} \qquad (3.3)$$

3.4.1.2 Tests et résultats

Afin de mesurer l'écart de performance entre les trois modèles word embedding séparément, nous avons calculé leurs couvertures et leurs précisions en comparant les résultats de chaque modèle avec le dictionnaire de référence. Les résultats de ces expériences sont résumés dans le Tableau 3.8.

Tableau 3.8. Performance des modèles word embedding

Modèles	**Précision**	**Couverture**
Wrod2vec CBOW	**0.815**	0.300
Word2vec Skip-Gram	0.775	0.280
FastText	0.414	0.163
Dictionnaire fusionné (CBOW + Skip-gram + FastText)	0.704	**0.463**

En examinant le Tableau 3.8, nous constatons en premier lieu que le modèle CBOW a atteint la meilleure précision, suivi de Skip-gram, alors que leur couverture est très faible. Deuxièmement, la performance de FastText est la plus faible. Finalement, nous pouvons remarquer que la combinaison des résultats des trois modèles, nous a permis d'atteindre une couverture nettement meilleure tout en conservant une bonne précision. Ces résultats montrent clairement que la combinaison des modèles permet de compenser le déséquilibre entre la haute précision et la faible couverture des modèles séparés.

D'autant plus, nous avons évalué les différentes fonctions de scoring de la similarité lexicale. Nous avons rapporté les résultats sur le Tableau 3.9. Nous notons, que ces valeurs présentées sont obtenues avec un seuil de 70%. D'autre part, ces résultats sont relatifs aux trois modèles fusionnés.

Tableau 3.9. Comparaison des fonctions de scoring de la similarité lexicale

Fonction de notation de la similarité lexicale	Précision	Couverture
Lexim (utilisé par Sridhar et Bertaglia)	**0.785**	0.300
Matching des séquences (utilisé par Htait)	0.671	0.427
Matching des séquences + Soundex	0.486	**0.578**
Matching des séquences + suppression des voyelles et des doubles consonnes	0.704	0.463

Le Tableau 3.9 montre que Lexim (utilisé par Sridhar et Bertaglia) surpasse toutes les autres fonctions de scoring en termes de précision, mais sa couverture est la plus faible. En termes de couverture, le matching de séquences avec Soundex a obtenu la meilleure couverture parmi toutes les autres fonctions. Cependant, le matching de séquences avec suppression des voyelles et des doubles consonnes a permis d'équilibrer les scores de précision et aussi de couverture. Pour cette raison, nous employons cette dernière fonction de scoring pendant la génération du dictionnaire MANorm.

Pour justifier notre choix de la valeur seuil utilisée dans le scoring de la similarité lexicale, nous avons mené une série d'expériences. Les résultats obtenus sont présentés dans le Tableau 3.10.

Tableau 3.10. Influence de la valeur seuil sur la précision et la couverture des modèles fusionnés

Valeur seuil	Précision	Couverture
60%	0.414	0.706
65%	0.428	0.697
70%	**0.704**	**0.463**
75%	0.703	0.462
80%	0.744	0.424

La principale conclusion que nous pouvons tirer de ce tableau est que les valeurs seuils

les plus élevées augmentent la précision mais réduisent la couverture. En faisant passer le seuil de 60% à 80%, la précision s'améliore considérablement, tandis que la couverture diminue drastiquement. Dans MANorm, nous avons donc utilisé le seuil moyen de 70 % qui équilibre entre la précision et la couverture du dictionnaire.

Lors de la validation de MANorm, nous avons constaté que l'agglutination, qui est principalement liée à l'inflexion des mots, est une véritable source d'ambiguïté puisqu'elle permet de générer un nombre important de translittérations. Notre principal souci étant de réduire la variation orthographique. Alors, dans le cas de formes infléchies (par exemple les verbes conjugués, les noms et adjectifs au pluriel et/ou au féminin), nous considérons le lemme des mots ou la forme d'inflexion la plus proche comme une normalisation correcte. En dialecte AM, le lemme des verbes est le passé de la troisième personne du singulier, et pour les noms et adjectifs, le lemme est la forme masculine du singulier. Le Tableau 3.11, illustre plusieurs exemples de normalisation des formes infléchies.

Tableau 3.11. Normalisation des formes infléchis

Etiquette POS	**Forme infléchie**	**Lemme/inflexion la plus proche**
Adjectif	saknin (ils habitent) sakna (elle habite)	Lem. masculin singulier : saken (il habite)
Nom	klamo (ses mots) klamek (tes mots)	Lem. masculin singulier : klam (mots)
Verb	3aqo (ils ont réalisé) 3aqna (nous avons réalisé)	Lem. passé, troisième personne du singulier : 3aq (il a réalisé)
	bakatni (elle m'a fait pleurer) bakitini (tu m'as fait pleurer) bakitona (vous m'avez fait pleurer)	Lem. passé, première personne du singulier : bkite (j'ai pleuré)
	3ajbatni (j'ai aimé) 3jbatni (j'ai aimé) 3jebni (j'ai aimé) kat3jabni (j'aime) kay3jabni (j'aime)	Inflexion la plus proche : 3jabni (j'ai aimé)

Nous avons également observé un autre problème lié à l'agglutination, par exemple, la

translittération "wqaaf" a été affectée à deux formes canoniques "wqef" (tenir) et "awqaf" (dotations islamiques). Cependant, nous avons remarqué que le mot "awqaf", utilisé dans le corpus, est une concaténation de "a" et de "wqaf" qui signifie "lève-toi" d'une manière forte. En vérifiant leurs contextes dans le corpus, nous avons constaté que "wqaaf" a la même signification que "wqef", que nous considérons alors comme correct.

3.4.1.3 Analyse des erreurs

En ce qui concerne les erreurs de normalisation, celles-ci sont liées à différentes raisons. Dans certains cas, nous avons observé qu'une translittération peut être attribuée à plusieurs formes canoniques. Dans un tel cas, pendant la validation, nous considérons comme correcte, la forme canonique qui est la plus proche sémantiquement à la translittération, en fonction de son contexte d'utilisation dans le corpus. Par exemple, la translittération "7amad" a été assignée à "7amd" (louange) et "7amed" (aigre) mais selon leur contexte, la forme correcte est la deuxième.

D'autres erreurs sont dues au fait que les conventions de translittération des phonèmes adoptées dans notre dictionnaire ne correspondent pas toujours à celles utilisées dans le corpus. Par exemple, le nom "7aj" (pèlerinage) a été attribué au verbe "haj" (rager) où "h" a été utilisé comme [ḥ] par certains utilisateurs. Cependant, dans le dictionnaire AM, nous utilisons "7" pour ce phonème et "h" pour représenter [ħ]. Ce problème est en partie dû au fait que, lors de la similarité lexicale, nous ne faisons pas la différence entre "7" et "h", vu que cela permet d'extraire plus de variantes de translittérations.

Le dernier type d'erreur observé survient lorsque la translitération et la forme canonique sont utilisées dans le même contexte, et sont lexicalement proches l'une de l'autre (similarité > 70%), bien qu'elles appartiennent à des mots différents. A titre d'exemple, "3id" (Eid/célébration religieuse) a été attribué à "sa3id" (heureux) comme forme canonique.

Finalement, même si les erreurs sont assez courantes dans MaNorm, il s'agit toujours d'une tentative intéressante de normaliser la variation orthographique des dialectes. Nous sommes convaincus qu'avec un corpus et un dictionnaire plus important, nous pouvons encore améliorer ses performances.

3.4.2 Evaluation de la phase transfert

Dans le cadre de ce travail, nous effectuons une évaluation *in vitro de* notre approche de traduction-désambiguïsation, puisqu'il s'agit du module le plus important de notre système. En d'autres termes, nous évaluons la performance de l'étape de transfert.

Afin de valider notre solution, nous avons choisi de l'appliquer aux phrases CS contenant des mots du Français et de l'Anglais qui sont des langues standards, en plus de l'Arabe Marocain (AM) dialectal. Dans ce sens, il faut préciser que ce choix dépend seulement des ressources disponibles, et il reste un cas applicatif que nous pouvons généraliser pour d'autres langues, vu l'aspect générique de notre approche.

3.4.2.1 Ressources

La première phase dans notre système MN, à savoir l'analyse, commence par une chaîne de prétraitements qui inclut la segmentation, la tokenisation, la normalisation et le nettoyage. Ces prétraitements sont effectués avec nos propres outils développés ainsi que ceux d'Apertium.

Quant à l'identification des langues (LID), pour les langues standards, nous avons utilisé comme analyseur morphologique, celui inclus dans la boîte à outils Apertium Translator (Corbí-bellot et al., 2005). Pour le dialecte, nous avons utilisé un dictionnaire bilingue Arabe Marocain-Français (Jaafar, 2012). Nous signalons que nous nous sommes intéressés à la translitération en script Latin, puisque le dictionnaire AM disponible suit ce script.

Afin de générer les modèles de langage requis pour l'analyse morphologique et la correction orthographique, nous avons combiné des corpus formels, et d'autres informels issus des médias sociaux pour élargir la couverture de toutes les langues concernées. Nous avons employé précisément, les textes Anglais et Français disponibles sur Europarl-v7 (Koehn, 2005) (texte formel) qui contient 60 millions de mots par langue, en plus de postes Twitter (Ling et al., 2013) (texte informel) composés de 559 phrases par langue.

En ce qui concerne la phase de transfert, nous avons choisi le dictionnaire sémantique BabelNet (Navigli and Ponzetto, 2012) pour les langues standards, et le dictionnaire bilingue précité pour la traduction du dialecte.

Quant à la réordonnancement, nous avons mis en œuvre notre propre module basé sur

la boîte à outils du modèle de langage KenLM (Heafield et al., 2013) avec un lissage Kneser-Ney, intégré au système de traduction automatique Moses[64]. Le corpus utilisé pour le Français, qui est la langue cible, contient un million de phrases extraites à partir de textes d'actualité du corpus Leipzig[65] (Goldhahn et al., 2012).

3.4.2.2 Apertium

Apertium[66] est une plateforme de traduction automatique libre à code source ouvert, permettant de construire des systèmes de traduction à transfert superficiel. La plateforme fournit un moteur de traduction automatique indépendant des langues. Elle fournit également, des outils linguistiques pour la gestion des données nécessaires à la conception d'un traducteur pour une paire de langue donnée. En plus de données linguistiques ouvertes pour plusieurs paires de langues.

À l'origine, Apertium a été conçu pour permettre la traduction de paires de langues très proches comme les langues parlées en Espagne par une équipe de chercheurs à l'université Alicante. Comme l'outil Apertium est open source, le projet s'est ouvert sur plusieurs langues aussi bien proches entre elles comme les langues des pays Scandinaves et les dialectes parlés en France. Puis à des paires de langues plus éloignées comme la paire Anglais-Catalan. Aujourd'hui, il couvre 51 pairs de langues en état stable.

Apertium suit l'approche typique de la traduction automatique basée sur le transfert qui divise la traduction en trois phases analyse, transfert et génération (Figure 3.7Figure 3.7. Architecture de la machine de traduction Apertium) :

- ***L'analyse*** *dans Apertium consiste en une analyse morphologique en plus d'un étiquetage POS du texte à traduire.*

- ***Le transfert*** *consiste en un transfert lexical et structurel. Il se base sur la détection de modèles de longueurs fixes de catégorie lexicales qui nécessitent un traitement pour l'accord, la réorganisation des mots, l'introduction de préposition si nécessaire …*

[64] http://www.statmt.org/moses/
[65] http://wortschatz.uni-leipzig.de/en/download/#corporaDownload.
[66] https://www.apertium.org/index.eng.html?dir=fra-epo#translation

- ***La génération*** *vise à reproduire correctement les formes lexicales du texte traduit obtenues, après la phase de transfert, en effectuant certaines opérations sur les formes infléchies telles que les contractions et les apostrophes.*

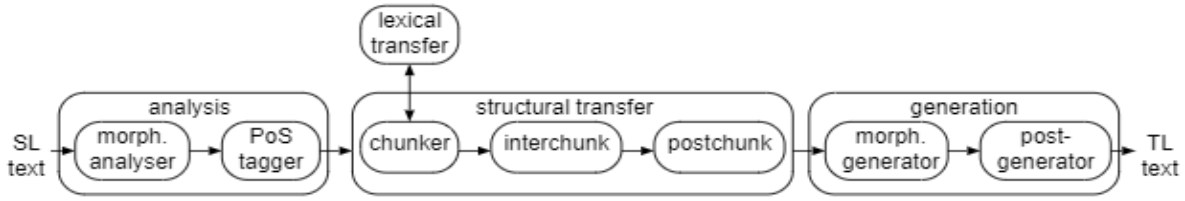

Figure 3.7. Architecture de la machine de traduction Apertium

Notre choix d'Apertium est justifié par trois raisons :

- Premièrement, par son système de traduction automatique à faible transfert. Nous n'avons donc pas à effectuer une analyse syntaxique approfondie de notre texte d'entrée, ce qui est impossible en raison de la complexité des phrases CS.
- Deuxièmement, Apertium couvre plusieurs langues, et encore plus c'est un traducteur libre qui permet d'adapter ses unités pour n'importe quelle paire de langues.
- Finalement, le concept adopté de séparation des algorithmes et des données facilite grandement la mise à niveau de n'importe quel module si nécessaire (il suffit d'ajouter les entrées manquantes dans le dictionnaire xml spécifique).

Le couple Anglais-Français existant sur Apertium, est encore à l'état d'incubateur (vient de démarrer). Pour remédier à ce problème, nous avons exploité les outils d'Apertium et les paires de langues existantes pour créer des analyseurs morphologiques pour l'Anglais et le Français. Ainsi que le générateur morphologique et le post-générateur pour le Français (langue cible).

Les analyseurs sont obtenus comme suit :

- D'abord, nous avons généré le Dictionnaire Morphologique (MD) en croisant les dictionnaires morphologiques de deux paires de langues existantes, à savoir, Espagnol-Français et Espagnol-Anglais, à l'aide de l'outil crossdics. Nous avons choisi l'espagnol parce que sa structure est la plus proche de l'anglais et du français parmi

les autres langues existantes.

- Ensuite, en utilisant la boîte à outils à états finis lttoolbox et les MDs, nous avons créé les analyseurs et le générateur. Le fichier de règles utilisé pour le poste réordonnancement a été également extrait du couple Espagnol-Français.
- Finalement, afin de résoudre l'ambiguïté de l'étiquetage POS pendant l'analyse, Apertium permet l'utilisation d'un étiqueteur statistique entraîné par un HMM[67] ou modèle de Markov caché. Nous avons compilé les étiqueteurs POS pour notre paire de langues avec les corpus proposés par Apertium.

3.4.2.3 BabelNet

BabelNet offre une traduction sémantique des mots de grande taille avec une large couverture linguistique. BabelNet est un réseau sémantique multilingue intégré dans lequel des millions de concepts sont lexicalisés dans plusieurs langues différentes. L'intégration a été faite fondamentalement par les connaissances lexicographiques et encyclopédiques de WordNet et Wikipedia, mais actuellement, il intègre de nombreuses autres sources.

BabelNet fournit des synonymes de mots et des traductions de et vers 500 langues. Ces traductions sont classées par catégories (les principales catégories sont Concepts et Entités nommées, les concepts sont également classés par domaine) et représentées par des synsets Babel, chaque synset Babel comprend les différents glosses, exemples, relations sémantiques (hyperonyme, hyponyme...), sens et autres informations. Plus de détails sur son concept sont disponibles dans (Navigli and Ponzetto, 2012). Un exemple des glosses des synsets fournis par BabelNet est donné dans Annexe C.

3.4.2.4 Dictionnaire Bilingue du dialecte Arabe Marocain (AM)

Ce dictionnaire comprend 48000 mots organisés suivant les catégories des parties de discours, comme par exemple : nom, verbe, adjectif, adverbe, préposition... Il fournit également la traduction des sens des mots avec quelques définitions et exemples. Pour la traduction, le dictionnaire fournit une liste des sens traduits. Nous avons choisi la traduction la plus fréquente car nous n'avons pas assez de ressources (définitions, exemples...) pour

[67] Le modèle de Markov, aussi appelé Chaîne de Markov, est un modèle statistique composé d'états et de transitions. Une transition matérialise la possibilité de passer d'un état à un autre. (http://igm.univ-mlv.fr/~dr/XPOSE2012/HiddenMarkovModel/description.html)

effectuer le scoring des sens. Nous utilisons également des lexiques spéciaux aux SM pour le dialecte AM.

3.4.2.5 Lexique Spécial au SM

Afin de traiter les abréviations, nous avons utilisé des dictionnaires de lexique de normalisation. Pour l'Anglais, nous nous sommes servis d'un dictionnaire[68] développé pour la normalisation des abréviations des micro-blogs (41181 tokens) (Han et al., 2012). Pour le Français, nous avons généré un dictionnaire (3226 tokens) en suivant la technique de normalisation décrite dans (Htait et al., 2018). Quant à l'AM, l'utilisation des abréviations n'est pas répandue chez les utilisateurs de ce dialecte, alors nous n'avons pas besoin d'un tel dictionnaire.

En ce qui concerne les expressions spéciales au SM, nous avons employé pour l'Anglais le dictionnaire[69] **NoSlang**. Pour le français, nous avons collecté ces expressions à partir de sources diverses, nous avons ainsi construit un lexique de 320 expressions. Et pour le AM, nous avons utilisé un lexique spécial SM (111 expressions) collecté à partir du web. Les idiomes Anglais (405 termes) traduits en Français ont été collectés à partir d'un glossaire en ligne[70]. Finalement, les traductions des expressions multi-mots (de l'Anglais au Français) ont été extraites de **BabelNet** et d'autres ressources web.

Des exemples du lexique des expressions spéciales, des expressions multi-mots et des idiomes du SM, traduits respectivement de l'Anglais et du MA vers le Français, sont donnés dans les tableaux ci-après :

Tableau 3.12. Exemples de lexique d'expressions spéciales aux SM en Anglais traduit en Français

Expression SM	Signification en Anglais	Traduction en Français
OMG (Anglais)	oh my god	oh mon dieu
amha (Français)	-	à mon humble avis
asap (Anglais)	as soon as possible	dès que possible

[68] http://people.eng.unimelb.edu.au/tbaldwin/etc/emnlp2012-lexnorm.tgz.
[69] https://www.noslang.com/dictionary/
[70] https://www.proz.com/glossary-translations/english-to-french-translations/idioms-maxims-sayings.

Tableau 3.13. Exemples de lexique des expressions spéciales aux SM en dialecte AM traduit en Français

Expression SM	Traduction en Français
tconnecta	Il s'est connecté
laykini	me faire un j'aime
statyat	les statues

Tableau 3.14. Exemples d'expressions multimots en EN traduites en FR

Multi-mots en anglais	Traduction en français
return back	retourner
in fact	en fait
instead of	au lieu de

Tableau 3.15. Exemples d'idiomes en EN traduits en FR

Idiom en Anglais	Traduction en Français
be on the clock	le temps presse
close the deal	conclure l'affaire
as and when	quand il le faut

3.4.2.6 Métriques d'évaluation

Nous avons utilisé comme métriques d'évaluation ceux de la recherche d'informations, qui sont la *précision*, le *rappel* et le *F1 score*. Ce choix est justifié par la nature de notre tâche, qui s'intéresse principalement à la désambiguïsation de la traduction pour la normalisation. Il ne s'agit pas donc d'un traitement de traduction automatique ordinaire où d'autres métriques spécifiques sont employées. Les formules de chacune des métriques sont données en dessous :

$$Précision = \frac{nombre\ des\ résultats\ pertinents}{nombre\ total\ des\ résultats\ fournis} \quad (3.4)$$

$$\text{Rappel} = \frac{nombre\ des\ résultats\ pertinents}{nombre\ total\ des\ résultats\ à\ fournir} \quad (3.5)$$

$$\text{F1 score} = 2 \times \frac{Précision \times Rappel}{Précision\ + Rappel} \quad (3.6)$$

3.4.2.7 Données

Les données expérimentales utilisées pour la validation sont de trois types. Le premier est une conversation Facebook contenant 231 phrases où 115 sont des phrases CS avec 171 mots cibles. Le second a été extrait des commentaires de YouTube contenant 363 phrases, dont 114 sont des phrases CS avec 197 mots cibles. Le dernier a été extrait des messages Twitter, il contient 418 phrases, dont 310 sont des phrases CS avec 675 mots cibles. Au total, nous avons testé notre système sur plus de 500 phrases CS contenant plus de mille mots cibles. Tous ces textes sont écrits en Français, Anglais et Arabe Marocain.

3.4.2.8 Tests et résultats.

Afin d'évaluer l'efficacité de l'approche proposée, nous avons réalisé plusieurs expériences. Comme expliqué dans la section LID, la langue cible est celle dont appartiennent la majorité des mots. Dans notre cas, nous avons identifié le Français en tant que langue cible, il s'agit donc de la langue matrice. Par conséquent, les langues sources sont l'Anglais et l'Arabe Marocain, qui sont donc, les langues intégrées. Par conséquent, nous devons effectuer la traduction de l'Anglais et de l'Arabe Marocain vers le Français.

Nous rappelons que notre principale évaluation porte sur l'étape de transfert, ainsi que l'approche du Matrix Langage Frame.

Pour tester le fonctionnement de la phase transfert, nous avons fourni à l'entrée de ce processus, le lemme d'un mot, sa langue et son POS tag. Ces caractéristiques ont été extraites au cours de l'étape d'analyse, comme mentionné précédemment. Tout d'abord, nous avons commencé par tourner notre système avec l'ensemble des synsets de BabelNet disponibles pour chaque mot. Toutefois, nous avons constaté que le traitement était très lent parce que le nombre de combinaisons pour certains cas était important. Pour réduire le temps

d'exécution, nous avons décidé de sélectionner les sept sens les plus familiers donnés par BabelNet. Nous avons également sélectionné les synsets ayant pour type *Concepts*, afin d'éliminer les Entités Nommées, et aussi ayant la même propriété POS que le mot cible. Ce qui nous a permis par la suite d'accélérer et d'affiner le résultat du scoring des sens.

Dans l'objectif d'évaluer la traduction des mots, nous avons comparé notre approche, le contexte vertical multilingue (MVC) avec deux références. La première référence c'est le contexte proche (NC : Nearest Context), le contexte comprend un mot à droite et un autre à gauche du mot cible. La seconde référence c'est la traduction la plus fréquente (MFT : Most Frequent Translation), que nous trouvons fournie en premier lieu sur les dictionnaires bilingues. Les résultats des tests de traduction-désambiguïsation sont présentés dans les tableaux ci-dessous.

Le Tableau 3.16 présente les résultats de l'évaluation de la traduction des mots cibles de l'Anglais vers le Français pour les conversations sur Facebook (FB), YouTube (YT) et Twitter (TW). Nous comparons notre approche MVC avec la MFT (Most Frequent Translation) et le NC (Nearest Context).

Tableau 3.16. Résultats de l'évaluation pour la traduction de l'EN vers le FR

Approche	Précision	Rappel	F1
MVC_{FB}	0.838	0.809	0.823
MFT_{FB}	0.778	0.751	0.764
NC_{FB}	0.817	0.774	0.794
MVC_{YT}	0.821	0.791	0.805
MFT_{YT}	0.744	0.725	0.734
NC_{YT}	0.736	0.710	0.722
MVC_{TW}	0.829	0.814	0.821
MFT_{TW}	0.767	0.771	0.768
NC_{TW}	0.774	0.759	0.766

Le Tableau 3.17 montre les résultats de la traduction des mots cibles en Arabe Marocain

vers le Français. Comme nous l'avons mentionné auparavant, la traduction donnée est celle du sens le plus fréquent fournie par le dictionnaire bilingue Arabe Marocain-Français.

Tableau 3.17. Résultats de l'évaluation pour la traduction de MA en FR

Approche	Précision	Rappel	F1
MA_{FB}	0.701	0.586	0.639
MA_{YT}	0.681	0.5	0.576
MA_{TW}	0.526	0.384	0.444

Le Tableau 3.18 présente les résultats de l'évaluation de la traduction en Français des mots cibles en Anglais et Arabe Marocain, dans le cas des conversations sur Facebook, YouTube et Twitter (TW).

Tableau 3.18. Performances globales des MVC, MFT et NC

Approche	Précision	Rappel	F1
MVC_{FB}	0.769	0.697	0.731
MFT_{FB}	0.729	0.625	0.670
NC_{FB}	0.671	0.579	0.619
MVC_{YT}	0.761	0.688	0.722
MFT_{YT}	0.712	0.612	0.655
NC_{YT}	0.631	0.547	0.583
MVC_{TW}	0.765	0.700	0.730
MFT_{TW}	0.724	0.635	0.672
NC_{TW}	0.650	0.571	0.605

Sur le dernier Tableau 3.19, nous donnons deux exemples de phrases CS et leurs sorties générées par notre approche, tout en les comparant avec des traducteurs existants.

Tableau 3.19. Exemples de traduction de mots cibles de l'EN et du MA vers le FR en utilisant notre approche et quelques traducteurs existants

Phrases	Mots à traduire	MVC	Google Translate	Microsoft Translate	Systran	PNMT & DeepL
il veut pas un **healthy diet** définitivement il dit que **li w9a3 w9a3**	healthy (EN) diet (EN) li (MA) w9a3 (MA)	sain régime alimentaire qui arriver	sain régime - -	en bonne santé régime alimentaire - -	En bonne santé Régime de l'ONU - -	sain régime alimentaire - -
daba il est devenu gourmand, il doit suivre son **diet** alimentaire	daba (MA) diet (EN)	maintenant régime alimentaire	- régime	- régime alimentaire	- Régime de l'ONU	- régime alimentaire

3.4.2.9 Test statistique

Afin de montrer davantage les performances de notre système par rapport aux autres références de base, nous avons effectué des tests de signification statistique pour comparer les résultats des différents systèmes considérés dans cette étude. Dans cette évaluation, nous avons établi une correspondance lexicale (lexical matching) entre les mots traduits et une référence annotée par un humain. Le choix du test dépend essentiellement de la nature des données de sorties. Sachant que ces données ne peuvent prendre que deux valeurs "match" ou "not-match". De ce fait, nous considérons nos sorties comme binomiales et par conséquent nous avons choisi le **Z-Test** qui est adapté à ce type de données. Le Z-Test a donné une *p-value* < *0,*05 (l'intervalle de confiance) pour les textes YouTube et Twitter, ce qui indique une amélioration significative du traitement de désambiguïsation de la traduction à l'aide de notre système (MVC) par rapport aux autres (MFT et NC). Cependant, pour Facebook, notre système ne montre pas de performance significative par rapport à l'approche du contexte proche (NC) de référence. La *p-value* était supérieure à l'intervalle de confiance. Cela est dû à la faible différence entre les sorties MVC et NC. Ce qui peut s'expliquer par le faible nombre d'occurrences d'une grande partie des mots cibles dans ce texte. Alors, dans cette situation, le MVC a été conduit à utiliser uniquement les mots environnants (contexte proche) des mots

cibles pour calculer les scores de recouvrement à la manière de l'approche NC.

3.4.2.10 Analyse des erreurs et Discussion

Notre objectif dans ce travail est la normalisation des textes SM. Particulièrement, nous nous intéressons à la normalisation des phrases CS à travers leur conversion en une version monolingue. Nous accordons plus d'intérêt à l'étape de transfert où nous essayons de traduire des phrases CS incluant des mots de langues standard et dialectale. Nous avons appliqué notre solution sur des phrases CS contenant des mots Français, Anglais et Arabes Marocain afin de les convertir aux phrases purement en Français. Les tests effectués ont donné des résultats significativement fiables, comme c'était montré lors de l'évaluation. Notre approche (MVC) a atteint une précision sensiblement plus élevée que les références de base NC et MFT. Ces résultats ont prouvé l'efficacité de l'identification automatique des langues source et cible en utilisant l'approche du Matrix Langage Frame, ainsi que la sélection lexicale en utilisant l'approche MVC.

Le Tableau 3.18 montre que notre système atteint une précision globale de 0,769 pour le premier texte (Facebook) et de 0,761 pour le second (YouTube), et pour le troisième (Twitter) la précision globale était de 0,765. Ces résultats sont un bon indicateur des performances de notre système. Le rappel pour les trois textes était respectivement de 0,697, 0,688 et 0,7. Ce qui indique également une assez bonne couverture de notre système.

Comme le montrent les tableaux ci-dessus, les performances de la traduction des mots Anglais est beaucoup plus élevée (précision respective : 0,838, 0,821 et 0,829) que celle des mots dialectes (précision respective : 0,701, 0,681 et 0,526). Ceci est lié à la richesse des ressources en Anglais. Alors que le dialecte souffre de la rareté de ses ressources, ce qui a impacté négativement la performance globale du système.

Il faut noter que notre évaluation concerne les mots de contenu. Quant aux mots de fonction qui ont été traduits avec Apertium, il n'est pas nécessaire de les évaluer puisqu'ils ne sont pas utiles pour les approches basées sur les corpus (cas de notre application cible). En effet, ces mots sont souvent supprimés du texte pendant la phase du prétraitement. Dans notre cas, nous avons gardé les mots de fonction, parce qu'ils peuvent être liés à des expressions multi-mots ou mots composés. Par leur suppression, nous risquons de perdre le sens de ces expressions.

Le Tableau 3.19 indique que la comparaison entre la traduction générée par notre étape de transfert et celles fournies par les autres traducteurs confirme notre constat. La majorité de nos lemmes de mots traduits sont alignés avec ceux des autres traducteurs. Cependant, les autres traducteurs ne peuvent pas traduire le dialecte car ils sont entraînés sur des corpus de langues standards. Contrairement à notre technique, qui est capable de traiter le dialecte puisqu'elle opère au niveau du mot.

Ces résultats ont permis de vérifier la faisabilité du contexte vertical multilingue (MVC) comme moyen d'élargir le choix lexical et ainsi d'aider à mieux choisir la signification appropriée des mots cibles. Si l'on considère un mot de contexte avec des synonymes limités (traductions ou synonymes), cela conduira à un faible scoring de sens pour le mot cible. Par contre, si nous détectons d'autres occurrences du mot cible, alors, les synsets des nouveaux mots de contexte enrichiront la comparaison avec les synsets du mot cible. Par conséquent, la sélection lexicale sera plus précise. Par exemple, sur le Tableau 3.19, nous pouvons remarquer que le mot *diet* a été traduit avec deux niveaux de contexte, qui ont fourni quatre mots de contexte (*définitivement*, *suivre* et *alimentaire en* Français et *healty en* Anglais).

Comme nous l'avons déjà mentionné dans ce document, si l'algorithme WSD ne peut pas désambiguïser en utilisant la technique MVC (Multilingual Vertical Context), il sélectionne alors la MFT (Most Frequent Translation). En effet, lors des tests, nous avons observé que certains mots ambigus sont traduits avec la technique MFT. En analysant ces résultats, nous avons constaté que s'il n'y a pas de relation (le score de recouvrement est nul) entre le mot cible et les mots qui l'entourent, cela signifie que leurs sens sont générales, et non spécifiques à un domaine donné. Dans ce cas, la MFT peut être le meilleur choix parce qu'elle représente la traduction du sens le plus général d'un mot. Toutefois, la fiabilité de cette explication dépend de la richesse des ressources lexicales pour une langue donnée. En revanche, s'il n'y a pas suffisamment des sens des mots, les combinaisons seront donc limitées pour le calcul des scores de recouvrement. Ainsi, il est très probable que l'algorithme renvoie un score de recouvrement nul et donc sélectionne la MFT. En conclusion, le MVC est utile pour désambiguïser un mot polysémique, lorsqu'il est utilisé dans un sens spécifique. Or, cette technique reste largement dépendante de la richesse du vocabulaire du dictionnaire. Cependant, dans le cas d'une signification générale, la MFT est plus appropriée.

Il est à noter que même si les références de base, la MFT et la NC, qui sont hors-contexte,

sont moins performantes par rapport à notre approche. Pourtant, elles peuvent être utilisées dans le cas où le mot cible est mentionné une seule fois dans le texte, en d'autres termes, lorsque l'application du contexte verticale n'est pas possible.

Comme mentionné auparavant, nous utilisons le POS tag des mots parmi d'autres critères durant le processus de la sélection lexicale. Par conséquent, le fonctionnement du MVC reste très sensible aux étiquettes POS attribuées pendant la phase d'analyse. Cependant, nous n'avons pas évalué l'étape d'analyse dans ce travail et donc, les erreurs de traduction liées à l'étiquetage POS incorrect ne sont pas prises en compte dans ces résultats. En outre, nous avons constaté que les idiomes et les mots composés ne sont pas largement présents dans notre texte d'entrée, ainsi nous ne pouvons pas évaluer leur couverture.

En ce qui concerne le dialecte AM, les formes infléchies des mots (pluriel, verbes conjugués...) ne sont pas détectées, en raison de l'absence d'un analyseur morphologique spécial pour ce dialecte. Pourtant, dans les trois conversations, les classes des mots AM les plus fréquents sont les noms et les adjectifs. Les formes infléchies sont rarement présentes. Donc, ce manque d'analyseur morphologique ne pénalise pas largement les résultats finaux. Cependant, la couverture reste moyenne d'une part, à cause des formes différentes des translitérations dialectales. D'autre part, les dictionnaires AM utilisées doivent être actualisés pour inclure aussi le lexique contemporain, surtout celui des SM. Ce qui nous oblige à travailler davantage sur ce point dans nos futurs travaux.

Finalement, avec notre solution, en combinant la détection automatique du mélange de langues contenues dans les phrases CS, avec la technique de Traduction-Désambiguïsation. Il est possible d'effectuer un transfert lexical tout en préservant le sens des mots avec une bonne précision. En outre, l'approche proposée peut être utilisée pour d'autres combinaisons de langues, si les ressources nécessaires sont disponibles.

3.5 Conclusion

Dans ce chapitre, nous avons présenté notre système Machine Normalization (MN), basé sur une approche de type MT. Il comprend un traducteur CS qui peut traiter les problèmes posés par le texte libre sur les SM. Le traducteur intégré dans notre système MN est indépendant de la langue. Contrairement aux autres traducteurs, il fonctionne sans indication des langues source et cible. Ce qui permet son intégration dans toute pipeline de traitement

automatique du langage.

Notre approche est linguistiquement fondée sur l'approche du Matrix Langage Frame, qui définit les langues mixées dans une phrase CS comme étant la langue dominante et les langues intégrées. Cette solution traduit l'ensemble des mots appartenant aux langues intégrées dans la langue dominante qui porte le sens de la phrase, ce qui permet de conserver la sémantique au niveau de la phrase. En outre, nous incluons un traitement WSD basé sur la connaissance dans l'étape de transfert. Afin de traduire les mots cibles avec préservation du sens, ce qui garantit la sémantique au niveau du mot.

Les résultats des tests semblent particulièrement prometteurs et encourageants pour poursuivre dans cette voie. Comme nous l'avons mentionné, notre système MN est dédié aux applications de Text Mining basées sur des corpus. Pourtant, il reste utile comme front-end ou prétraitement pour les systèmes de traduction automatique afin de normaliser les énoncés CS avant la traduction principale.

Ce travail est encore une première tentative de normalisation des textes SM ainsi que du CS. D'autres améliorations sont nécessaires, comme la résolution du problème des mots non reconnus (OOV). En particulier, les mots dialectes et l'étiquetage incorrect des POS.

Dans le cadre des travaux futurs, et afin de vérifier l'efficacité de notre approche de normalisation, nous comptons l'appliquer en tant que prétraitement dans une application d'analyse de texte basée sur l'apprentissage automatique.

Chapitre 4

Détection des Traits de Violence dans les Médias Sociaux

Notre objectif dans ce chapitre est de détecter les traits de violence des auteurs impliqués dans des actes de cyberviolence à partir de leurs écrits sur les médias sociaux. Cet objectif sera atteint au moyen des techniques puissantes de l'intelligence artificielle. En particulier les algorithmes Ensemble ML, avec des caractéristiques d'apprentissage liées aux émotions et aux Big Five traits de personnalité des utilisateurs en ligne. Ces techniques seront comparées avec ceux de DL (Deep Learning) afin de vérifier leurs performances.

Les travaux élaborés dans ce chapitre ont fait l'objet des publications suivantes :

- Zarnoufi, R., Boutbi, M. and Abik, M. (2020) 'AI to prevent cyber-violence: harmful behaviour detection in social media', Int. J. High Performance Systems Architecture, Vol. 9, No. 4, pp.182–191.

- Zarnoufi R., Abik M. (2020) Big Five Personality Traits and Ensemble Machine Learning to Detect Cyber-Violence in Social Media. In: Serrhini M., Silva C., Aljahdali S. (eds) Innovation in Information Systems and Technologies to Support Learning Research. EMENA-ISTL 2019. Learning and Analytics in Intelligent Systems, vol 7. Springer, Cham. https://doi.org/10.1007/978-3-030-36778-7_21

4.1 Introduction

Dans le chapitre précèdent, nous avons abordé le problème de normalisation des textes bruités générés par les utilisateurs des SM, et dont la nature constitue un obstacle à leur analyse. Dans le présent chapitre, nous attaquons le problème d'analyse des textes pour y détecter le contenu violent caractérisant les actes de cyberviolence.

La cyberviolence peut être définie par : un abus en ligne contre un individu ou un groupe, souvent avec des effets perturbants les victimes. La cyberviolence a été largement abordée dans la littérature sous différentes appellations comme le cyberharcèlement, le discours haineux, et le langage offensif, agressif ou toxique. Cependant, bien que, en réalité, ces études utilisent différentes expressions, leur intérêt reste la détection du contenu violent généré par l'auteur de l'acte de cyberviolence afin de protéger les autres utilisateurs. Par conséquent, dans la présente étude, nous considérons '*cyberviolence*' tout acte reflétant une violence virtuelle.

Quant aux techniques employées dans la détection de la cyberviolence, et après une analyse de la littérature, nous avons élaboré les conclusions suivantes :

- La majorité des études connexes dans le domaine computationnel ont été principalement axées sur les techniques d'apprentissage automatique supervisé basé sur des caractéristiques souvent de nature technique (voir section 2.2.4.2-a).
- D'autant plus, nous pouvons observer ces dernières années, la large introduction des techniques d'apprentissage profond dont le but est d'améliorer les performances des systèmes existants sans avoir recours à la préparation des caractéristiques d'apprentissage (features engineering). En revanche, ces techniques nécessitent de grandes quantités de données annotées qui sont difficiles à produire.
- Nous avons également noté que les études précédentes ont négligé des facteurs importants pour la détection du comportement violent, tels que, les caractéristiques de la personnalité et du comportement humain (Sanchez and Kumar, 2011).
- Finalement, les études psychologiques relatives à la cyberviolence recommandent d'étudier les caractéristiques psychologiques de la personnalité des auteurs en particulier les émotions et les traits de personnalité.

D'après les résultats des études en psychologie sur la cyberviolence, nous pensons que les facteurs psychologiques peuvent être très utiles dans le processus de détection. Alors, dans cette étude nous irons construire des modèles de prédiction du contenu violent basés sur le langage et les caractéristiques psychologiques.

Notre motivation principale derrière l'extraction des traits de violence à partir des écrits des individus vient de la relation étroite entre le langage, la personnalité et le comportement. En effet, un large éventail d'études ont été établies sur le lien entre l'emploi du langage et les caractères psychologiques. Ces travaux montrent que l'usage des mots est différent entre les individus, mais corrélé avec leurs personnalités et leurs comportements (Davahli et al., 2020; Mairesse et al., 2007; Moreno et al., 2021; Tausczik and Pennebaker, 2010; Yarkoni, 2010). Également, les études menées sur les utilisateurs des SM montrent une forte relation entre leurs écrits et leurs traits de personnalité (Azucar et al., 2018; Schwartz et al., 2013).

Ce chapitre est constitué de six sections : la première explique la relation entre le langage, la personnalité et les traits de violence. La deuxième présente l'approche adoptée dans sa globalité. Puis, dans la troisième nous considérons les émotions des individus comme caractéristiques d'apprentissage afin de détecter le contenu violent. Tandis que dans la quatrième nous considérons les Big Five traits de personnalité comme caractéristiques. La cinquième présente une comparaison avec des architectures DL. La dernière sera consacrée à la présentation de notre dataset de violence, que nous avons établie pour l'Arabe Marocain.

4.2 Approche

Dans notre approche, nous analysons les textes générés par les utilisateurs afin de révéler les caractéristiques comportementales des cyber-perpétrateurs. Plus précisément, nous supposons que le comportement violent du cyber-perpétrateur peut être identifié à partir de ses émotions et de ses traits de personnalité.

Pour vérifier cette hypothèse, nous avons adopté les techniques ML classiques qui passent par une étape d'extraction des caractéristiques suivie par l'étape d'apprentissage supervisé. En premier lieu, nous avons procédé à l'extraction des caractéristiques liées aux états émotionnels. Cette extraction repose sur la similarité sémantique en exploitant la technique du word embedding plutôt que la simple similarité lexicale (par appariement des mots). Sur ces caractéristiques, nous avons entraîné des algorithmes d'Ensemble ML pour

prédire la présence ou non d'un comportement violent d'un utilisateur à partir du texte écrit par ce dernier. En deuxième lieu, nous avons appliqué le même processus avec les caractéristiques basées sur les Big Five traits de personnalité. Finalement, nous avons comparé les performances des modèles générés avec ceux des techniques DL. La Figure 4.1 illustre l'ensemble de la méthodologie, y compris l'étape d'apprentissage supervisé.

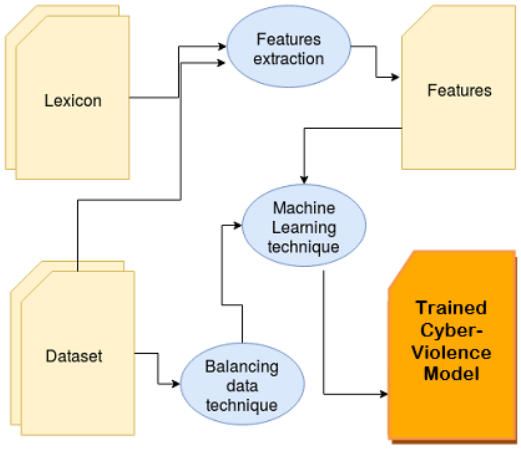

Figure 4.1 : Architecture du système de détection de la cyberviolence

Comme cas d'application, nous appliquons notre approche sur la détection du cyberharcèlement qui est une forme courante de cyberviolence, largement présente chez les jeunes et en particulier chez les adolescents. C'est le type de cyberviolence le plus étudié, car il impacte négativement la psychologie de ces personnes et donc leur réussite dans la vie. Ce qui en résulte que les ressources nécessaires, en termes de dataset et aussi de corpus lexicaux disponibles, sont principalement liées à l'étude de cette forme de cyberviolence.

Cette approche a plusieurs avantages, d'une part elle nous permet de mieux comprendre l'acte de cyberviolence. D'autre part, elle est basée sur le langage des émotions et des traits de personnalité, ceux-ci peuvent être présents dans presque tous les types de violence à travers les expressions employées par les cyber-perpétrateurs. Ce qui signifie que notre solution reste toujours applicable pour détecter d'autres formes de cyberviolence. En d'autres termes c'est une approche générique.

La principale différence entre notre approche et celles précédentes est notre intérêt particulier porté à l'auteur lui-même, en qui nous voulons découvrir et le comportement violent en fonction de ses états émotionnels et les traits de personnalité. Ce comportement violent est considéré comme la caractéristique commune de la majorité des formes de

cyberviolence. Regardant les autres approches, ces dernières se focalisent surtout sur la protection des victimes, en négligeant complétement l'acteur de la cyberviolence qui a aussi besoin de l'assistance pour qu'il s'arrête de nuire aux autres.

Bien que notre solution traite chaque tweet indépendamment, nous pouvons collecter un ensemble de tweets générés par le même utilisateur, et prédire si leur contenu est violent ou non violent. Ainsi, nous pouvons soulever une vue d'ensemble sur son comportement en ligne.

4.3 Détection des traits de violence à l'aide des émotions

Dans cette approche, nous supposons que les émotions du cyber agresseur peuvent être de bons indicateurs de son comportement violent. Par conséquent, nous allons étudier ce fait en moyennant un ensemble de lexiques d'émotions, un dataset de tweets pré-labélisés et des techniques Ensemble ML. Plus de détails sur notre approche seront donnés dans la suite de cette section.

4.3.1 Les émotions

Une émotion est une réaction affective temporaire d'intensité variable, qui se produit en réponse à un événement déclencheur. Il existe de nombreux types d'émotions qui exercent une influence sur notre façon de vivre et d'interagir avec les autres.

Les psychologues ont essayé d'identifier et de classifier les différents types d'émotions que les gens expérimentent. Différentes théories ont vu le jour pour catégoriser et expliquer ces émotions, nous présenterons trois parmi celles les plus influentes en psychologie.

En 1890, William James (William, 1890) a proposé quatre émotions fondamentales : la peur, le chagrin, l'amour et la rage, qui viennent comme conséquence d'une réaction physique suite à un évènement. En d'autres termes, un stimulus provoque une réponse physique et une émotion suivant cette réponse.

Le psychologue Paul Ekman a premièrement identifié six émotions fondamentales qui sont universelles pour toutes les cultures humaines (Ekman, 1972) : la peur, le dégoût, la colère, la surprise, le bonheur et la tristesse. Il a ensuite (Ekman, 1999) élargi sa liste à quinze émotions de base pour y inclure des éléments tels que la fierté, la honte, la gêne, la satisfaction

et l'excitation.

Dans les années 1980, Robert Plutchik (Plutchik, 1980) a introduit un autre système de classification des émotions connu sous le nom de "roue des émotions". Cette classification est le modèle le plus utilisé dans les études psychologiques sur les émotions. Plutchik a proposé huit dimensions émotionnelles primaires qui peuvent servir comme base pour d'autres émotions. Ces émotions de base sont composées de quatre formes positives contre quatre négatives : la joie contre la tristesse, la colère contre la peur, la confiance contre le dégoût, et la surprise contre l'anticipation. Ces émotions peuvent ensuite être combinées pour en créer d'autres. La Figure 4.2 montre comment cette combinaison peut être faite. Par exemple, joie + confiance = amour, et tristesse + dégout = remords.

Chacune des feuilles de la roue représente une émotion. L'émotion de base se trouve au milieu sur le deuxième cercle, entourée de deux autres formes. La couleur claire représente une faible intensité de cette émotion, alors que la couleur foncée signifie une forte intensité de la même émotion. Par exemple, l'émotion de base 'la confiance' possède une forme atténuée qui est l'acceptation et une autre intense qui est l'admiration.

Plutchik propose que les émotions aient une relation avec les traits de personnalité, il dit que la manière dont la personnalité est conceptualisée et décrite dans le langage découle de l'émotion (Plutchik and Conte, 1997). En outre, Revelle and Scherer, (2009) ont décrit cette relation par : ce que l'on prévoit est la personnalité, ce que l'on observe à un moment donné est l'émotion. Cependant, les expériences empiriques visant à établir cette relation ont été entravées par le manque de ressources complètes sur les émotions.

Toutefois, dans un travail plus récent, Mohammad and Kiritchenko, (2014) ont pu montrer cette relation : d'abord ils ont construit un lexique des émotions à partir de Twitter. Puis, ils se sont servis de ce lexique comme caractéristiques d'apprentissage d'un classifieur de phrases suivant le trait Big Five de personnalité appropriée. Ce classifieur a donné de bonnes performances en prouvant ainsi l'association entre les Big Five traits de personnalité et le langage des émotions.

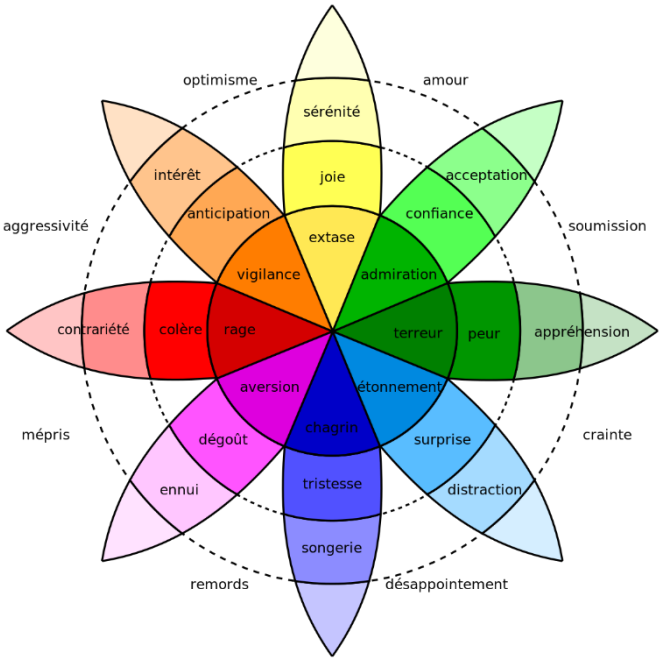

Figure 4.2. La roue des émotions de Plutchik

Le langage lié aux émotions peut se présenter sous forme gestuelle, verbale ou textuelle. Sa forme textuelle largement présente dans les SM est souvent utilisée pour exprimer les émotions qui se produisent suite à des expériences personnelles dans le monde réel ou virtuel. Par conséquent, l'analyse de ces textes, pour en extraire les émotions de son auteur, peut nous donner une idée sur sa personnalité et son comportement. Dans cette perspective, nous emploierons les émotions de base définies par Plutchik comme caractéristiques d'apprentissage des modèles ML, afin de détecter le contenu violent généré par les utilisateurs des SM. La section suivante donnera les détails de notre approche.

4.3.2 Méthode

La méthode suivie pour notre approche repose sur l'utilisation des algorithmes ML classiques dont le fonctionnement inclut deux phases : la première est l'extraction des caractéristiques d'apprentissage et la seconde est l'entraînement du modèle.

4.3.2.1 Extraction des caractéristiques

Afin d'extraire les caractéristiques linguistiques (mots, expressions, symboles), liées aux émotions nous avons opté pour l'approche du vocabulaire ouvert (Schwartz et al., 2013), plutôt que l'utilisation d'un lexique spécial figé comme celui du Linguistic Inquiry and Word

Count (LIWC) (Pennebaker et al., 2015). Le principale avantage du vocabulaire ouvert vient du fait que les caractéristiques linguistiques sont automatiquement identifiées et extraites à partir des textes écrits par les utilisateurs. Les lexiques spéciaux, quant à eux, se limitent à des listes de mots prédéfinis, ce qui les empêche de couvrir largement les mots utilisés dans les différents types d'émotions. Pour cela, nous avons utilisé EmoLex (Mohammad and Turney, 2013), un lexique extrait de tweets contenant des mots liés aux huit émotions de base proposées par Plutchik : anticipation, colère, peur, confiance, surprise, tristesse, joie et dégoût.

Dans cette étude, même si nous nous intéressons aux contenus violents, nous avons utilisé d'une manière indifférente les émotions positives et négatives. Parce qu'en réalité, certains comportements violents peuvent être associés à des expressions contenant des émotions positives. Comme dans le cas du sarcasme[71] ou du sadisme[72] où l'auteur peut exprimer de la joie tout en blessant quelqu'un d'autre.

Quant à l'extraction, nous avons d'abord appliqué la similarité lexicale pour faire correspondre chaque mot du lexique (pour chaque émotion) à chaque mot des messages (dataset des tweets). Cependant, nous avons constaté que la similarité lexicale n'est pas suffisante pour confirmer ou infirmer qu'un mot exprime l'émotion représentée par le lexique. Par conséquent, en plus de cette analyse lexicale, nous avons utilisé un modèle de word embedding pour effectuer la similarité sémantique et améliorer ainsi le processus d'extraction.

Le calcul de la similarité totale $Score_{sim}$ (formule 4.1) se déroule comme suit : pour chaque message et pour chaque type d'émotions, nous calculons d'abord la similarité lexicale $Score_{simlex}$, dans ce cas nous cherchons si ce message contient des mots du lexique de cette émotion. Pour chaque correspondance trouvée, nous incrémentons le score de similarité $Score_{sim}$ d'une valeur égale à 1. Puis, nous passons au calcul de la similarité sémantique $Score_{simsem}$, en calculant la distance cosinus entre les vecteurs représentatifs de chaque mot du message et ceux du lexique. Si la valeur de la distance est supérieure à un seuil donné (nous l'avons fixé à 60%), elle sera ajoutée au $Score_{simsem}$. Ce processus est répété pour chaque

[71]Sarcasme : L'utilisation de remarques qui signifient clairement le contraire de ce qu'elles disent, dans le but de blesser quelqu'un d'autre. (Dictionnaire Cambridge)
[72] Sadisme : Plaisir pris à faire souffrir, jouissance tirée du malheur des autres. (LAROUSSE)

type des huit émotions.

$$Score_{sim} = Score_{simlex} + Score_{simsem} \quad (4.1)$$

Il est à noter que les deux étapes de l'extraction des caractéristiques, qui sont les similarités lexicale et sémantique, seront évaluées dans la section de validation.

4.3.2.2 Algorithmes d'apprentissage et problème de dataset déséquilibré

Après avoir extrait les huit émotions de notre dataset, nous avons appliqué des techniques d'apprentissage supervisé afin de prédire les comportements violents. Cette tâche de prédiction peut être considérée comme une classification binaire, où l'étiquette *Oui* est donnée pour *violent* et *Non* pour *Non violent*. Les techniques ML utilisées pour la tâche en question varient entre les techniques classiques et celles de Deep Learning. Ces dernières années, les algorithmes DL ont montré de bonnes performances dans cette tâche, mais leurs performances dépendent de la disponibilité d'une très grande quantité de données d'apprentissage. Cependant, les algorithmes ML classiques sont toujours pratiques lorsqu'il s'agit de dataset de taille limitée.

Puisque, nous n'avons pas un large dataset, nous avons décidé d'explorer les ML classiques. En outre, comme nous l'avons mentionné, le dataset utilisé dans notre implémentation possède une distribution déséquilibrée avec 86% pour la classe négative (*Non violent*) et 14% pour la classe positive (*Violent*). Ce dataset déséquilibré influencera négativement la fonction de décision du modèle en favorisant la classe majoritaire pendant la phase d'apprentissage, et par conséquent il induira des erreurs pendant la phase de prédiction. Pour résoudre ce problème, nous avons choisi des classifieurs d'Ensemble basés sur les arbres de décision, à savoir Random Forest, Gradient Boosting, XGBoost et Adaboost, très connus par leur capacité à traiter des données déséquilibrées. De plus, avant la phase d'apprentissage, nous avons inclus un prétraitement de données afin de l'équilibrer et du coup améliorer le processus de détection.

Plusieurs techniques ont été développées pour remédier au problème de dataset déséquilibré. Dans la littérature, nous distinguons deux types d'approches fondamentaux. Le premier type, dit *orienté données,* qui fonctionne au niveau de données en essayant de les

pousser à joindre leur équilibre. Tandis que le deuxième type, *orienté algorithme,* fonctionne en modifiant les algorithmes de classification existants pour les rendre appropriés aux datasets déséquilibrés.

Dans la suite, nous expliquons notre choix de techniques d'équilibrage de dataset et aussi des classifieurs utilisés dans cette étape de détection.

a. Les techniques orientées données

Les techniques orientées données cherchent à réduire l'impact du déséquilibre indépendamment de l'algorithme employé, en les intégrant comme une phase de prétraitement. Ces techniques ont prouvé leur efficacité lorsqu'elles sont combinées à des techniques d'Ensemble (Feng et al., 2018). Leur fonctionnement repose sur le *rééchantillonnage (Resampling) du dataset qui peut être fait de trois façons différentes :*

- La réduction du nombre des échantillons de la classe majoritaire, nous parlons dans ce cas de sous-échantillonnage (undersampling), qui crée un sous-ensemble de l'ensemble de données original en éliminant des instances généralement de la classe majoritaire. Son inconvénient majeur est qu'elle peut provoquer la perte de données qui sont importantes pour le processus d'inférence.

- L'augmentation du nombre des échantillons de la classe minoritaire, c'est le suréchantillonnage (upsampling), qui créent un sur-ensemble de l'ensemble de données original en répliquant certaines instances ou en créant de nouvelles instances à partir d'instances existantes. Cette reproduction d'instances peut être faite aléatoirement, ce qui peut générer un surapprentissage, suite à la création de copies identiques aux instances existantes. Pour remédier à ce problème, d'autres techniques plus sophistiquées peuvent être utilisées. Comme SMOTE (Chawla et al., 2002) ou suréchantillonnage synthétique des minorités, et sa version modifiée MSMOTE. L'idée principale de SMOTE est de créer de nouveaux échantillons de la classe minoritaire en interpolant plusieurs instances de cette classe, qui sont du même voisinage en se servant du kNN (k nearest neighbours) algorithme.

- Une hybridation qui combine les deux méthodes d'échantillonnage.

Etant donné que notre dataset est de taille limitée, il est donc inapproprié d'utiliser un

équilibrage par sous-échantillonnage qui va nous faire perdre encore de données. Pour cela, nous avons opté pour SMOTE qui a prouvé sa capacité à améliorer les performances des classifieurs, comme signalé dans (Al-garadi et al., 2016).

b. Choix des techniques orientées algorithme

A la tête des techniques orientées algorithme, nous trouvons Ensemble ML basées sur des arbres de décision. Nous avons adopté cette approche, parce qu'elles possèdent d'une part, une structure hiérarchique qui lui permet d'apprendre les signaux des deux classes, majoritaire et aussi minoritaire, ce qui la rend très adéquate pour un dataset déséquilibré. D'autre part, ces techniques ont montré un grand pouvoir prédictif dans plusieurs tâches NLP, comme l'analyse de sentiments (Fersini et al., 2016) et la détection du langage offensif (Rajendran et al., 2019).

L'idée derrière Ensemble ML est de combiner des faibles modèles d'apprentissage, afin de produire un modèle de prédiction fort et ainsi améliorer le résultat global. Ceci est très utile pour traiter des données déséquilibrées au niveau algorithmique. Ces algorithmes peuvent être utilisés également dans des problèmes de régression ou de classification. Nous pouvons classer les techniques d'Ensemble en différentes catégories, dont les plus employées sont, le *Bagging* (Breiman, 1996) et le *Boosting* (Schapire, 1990).

Le Bagging ou boostrap aggregating, le bootstrap est une technique d'échantillonnage dans laquelle des sous-ensembles (bag) d'observations sont créés à partir de l'ensemble de données original, avec remplacement. Un modèle de base (modèle faible) est créé sur chacun de ces sous-ensembles en *parallèle* et *indépendamment* les uns des autres. Les prédictions finales sont déterminées en combinant (agrégation) les prédictions de tous les modèles par un vote majoritaire (cas de classification). L'avantage principale de cette agrégation est la réduction de l'erreur de prédiction et donc produire un fort modèle. Le modèle le plus connu de ce type est le Random Forest.

Le Boosting est un processus *séquentiel*, où chaque modèle postérieur tente de corriger les erreurs du modèle antérieur (apprenant faible). Ces modèles de base sont entraînés sur des données pondérées. Après chaque cycle de classification, les échantillons mal classés seront accordés un poids plus important. Le modèle final (apprenant fort) est obtenu à travers la combinaison de tous les modèles par un vote majoritaire pondéré. Les modèles les plus

répandus de ce type sont Gradient Boosting, XGBoost, CatBoot et AdaBoost.

Dans notre cas, nous avons utilisé quatre types de classifieurs d'Ensemble qui sont :

- *Random Forest* : c'est un algorithme qui construit une forêt d'arbres de décision à partir de sous-ensembles aléatoires, extraites du dataset et aussi des caractéristiques d'apprentissage. Il construit un grand nombre d'arbres de décision, qui seront combinées selon des règles de majorité à la fin du processus de classification.
- *Gradient Boosting* : cette technique aussi combine des arbres de décision, avec la différence que la combinaison des modèles se passe séquentiellement, depuis le début du processus de classification, et non à posteriori comme pour Random Forest. Chaque arbre postérieur dans la série est construit sur les erreurs calculées par l'arbre précédent en minimisant les fonctions de perte. Ce qui en résulte, un fort modèle de prédiction.
- *XGBoost ou Extreme Gradient Boosting* : c'est une version avancée du Gradient Boosting. En général plus rapide et plus performant. Sa rapidité vient de son implémentation du traitement parallèle. Il comprend également une variété de régularisations[73] qui réduisent l'overfitting[74] (surapprentissage) et améliorent les performances globales. Il intègre aussi une fonction pour le traitement des valeurs manquantes.
- *AdaBoost ou Adaptive Boosting* : pareil, cet algorithme construit un classifieur puissant en combinant plusieurs classifieurs, peu performants, par une surpondération des échantillons mal classés et une sous-pondération des échantillons correctement classés. Les échantillons mal classés auront une probabilité plus élevée d'être sélectionnés pendant le cycle suivant. Le modèle postérieur se charge de prédire leurs classes correctement. L'échantillonnage employé est adaptatif puisqu'il dépend des erreurs commises d'un faible apprenant au suivant.

Dans la section suivante, nous présenterons les différentes expériences menées ainsi que

[73] La régularisation est une technique utilisée pour réduire l'erreur de prédiction et éviter l'overfitting et donc favoriser la généralisation du modèle.
[74] L'overfitting arrive dans les cas où l'apprentissage s'effectue trop longtemps ou lorsque les exemples d'entraînement sont rares. Ce qui amène l'algorithme d'apprentissage à s'adapter à des caractéristiques aléatoires très spécifiques des données d'entraînement qui n'ont pas de relation causale avec la fonction cible. (Wikipedia)

les résultats obtenus.

4.3.3 Évaluation

L'évaluation de notre approche concerne trois aspects principaux : Premièrement, la validation de notre choix des différents classifieurs de la catégorie Ensemble ML (Random Forest, Gradient Boosting, XGBoost et AdaBoost). Deuxièmement, le test de l'impact de l'intégration de l'analyse sémantique pendant la phase d'extraction des caractéristiques. De plus amples détails seront présentés dans la suite, mais tout d'abord nous présenterons les ressources sur lesquels nous avons appliqué nos classifieurs.

4.3.3.1 Ressources

Dans cette section, nous présentons les ressources utilisées dans nos différentes expériences en termes de lexiques et de dataset.

a. Lexiques

Comme nous l'avons dit précédemment, notre méthode repose sur les émotions comme caractéristiques d'apprentissage. Nous avons compilé ces caractéristiques à partir du corpus lexical EmoLex (Mohammad and Turney, 2013) ou NRC Emotion Lexicon[75] (Tableau 4.1). Emolex contient neuf types de lexiques qui représentent la relation entre des mots/expressions et les huit émotions de Plutchik (anticipation, colère, peur, confiance, surprise, tristesse, joie et dégoût), en plus des sentiments (négatifs et positifs).

Ce lexique a été généré automatiquement à partir de tweets comportant des hashtags de mots d'émotion tels que #happy, #anger, etc. Il contient près de 17 000 uni-grammes pondérés avec des scores d'association à valeur réelle. La distribution du lexique par émotion est comme suit : l'anticipation 3309 entrées, la colère 5614 entrées, la peur 3814 entrées, la confiance 1668 entrées, la surprise 6032 entrées, la tristesse 2558 entrées, la joie 3435 entrées et le dégoût 5363 entrées. Son annotation a été faite manuellement sur le Mechanical Turk[76] d'Amazon. Emolex a été établi premièrement pour l'Anglais, puis il a été traduit dans plus de cent langues.

[75] NRC Word-Emotion Association Lexicon (NRC : Conseil National de Recherches du Canada) : https://www.saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

[76] C'est une plateforme web de crowdsourcing qui vise à faire effectuer par des humains, contre rémunération, des tâches plus ou moins complexes pour une machine.

Tableau 4.1. Exemples de lexique Emolex avec les poids correspondants

Emotion	Mot	Poids
l'anticipation	#expecting	2.237478095
la colère	jerk	0.593667390
la peur	security	0.518031195
la confiance	admitting	1.485154665
la surprise	tricks	0.936144418
la tristesse	hibernate	1.067902590
la joie	yey	1.747070367
le dégoût	#vomit	1.518608679

Nous notons qu'avant de réaliser nos tests, le lexique relatif à chacun des huit états émotionnels a été regroupé dans une classe indépendante pour faciliter son utilisation.

b. Dataset

L'un des anciens obstacles que les chercheurs rencontrent lorsqu'ils tentent de détecter les formes de cyberviolence est le manque de datasets annotés et de corpus lexicaux. Comme mentionné précédemment, la cyberintimidation ou le harcèlement en ligne (cyberbullying), est la forme de cyberviolence la plus étudiée dans les travaux antérieurs. C'est pourquoi les datasets correspondants sont largement disponibles. Alors, nous avons utilisé un dataset annoté sur le harcèlement[77] (Rezvan et al., 2018), il est mis gratuitement à la disposition des chercheurs sous demande auprès de ses auteurs. Ce dataset comprend cinq types différents de contenu de harcèlement : racial, sexuel, lié à l'apparence, intellectuel et politique. Il a été conçu en explorant d'abord les données de Twitter à l'aide d'un lexique de mots profanes ou offensants. Ensuite, des juges humains ont annoté les tweets par le label *oui,* en cas de présence d'harcèlement, et *non* dans le cas contraire. Le corpus total se compose de 25 000 tweets annotés pour les cinq types de contenu harcelant.

Nous avons décidé d'entrainer nos modèles sur le harcèlement racial qui inclut 5000 tweets. Ce type d'harcèlement est parmi les formes de violence les plus populaires. Le Tableau

[77]https://github.com/Mrezvan94/Harassment-Corpus

4.2 présente un exemple de deux entrées de ce dataset.

Puisque ce dataset est disponible sous forme brute, donc, avant de l'exploiter dans nos expériences, nous l'avons d'abord nettoyé. Ce nettoyage vise à en exclure des éléments indésirables, non porteurs de sens, tels que les URLs, les mots vides, les mentions @ et les chiffres.

Tableau 4.2 Exemples d'entrées du dataset d'harcèlement racial

Décision	***Tweet***
Non (non violent)	@brandonlee161 paki haha i'm joking how are you mate? (paki haha je plaisante comment vas-tu mon pote ?)
Oui (violent)	@asadowaisi his father forgot to board train to lahore in 1947 and left this paki pig in india. (@asadowaisi son père a oublié de prendre le train pour lahore en 1947 et a laissé ce cochon de paki en inde.)

4.3.3.2 Expériences

Nous avons conduit plusieurs expériences pour examiner à la fois les caractéristiques d'apprentissage et l'impact de l'analyse sémantique pendant l'étape d'extraction, ainsi que le choix du classifieur dans l'étape de prédiction.

a. Tests de validation des caractéristiques d'apprentissage

En ce qui concerne l'extraction des caractéristiques, nous avons effectué deux expériences pour chaque classifieur. La première examine uniquement l'analyse lexicale et la seconde l'analyse sémantique. Nous notons que pour le calcul de la similarité sémantique, nous avons utilisé le modèle de word embedding Word2Vec entraîné sur le dataset Google News plutôt que celui entraîné sur Wikipedia. Vu que le premier contient moins de mots formels que le second, ce qui convient plus à notre dataset composé de tweets.

b. Tests de validation du choix des algorithmes d'Ensemble

Quant à l'étape de prédiction, pour évaluer les performances des algorithmes d'Ensemble choisis, nous avons conduit cinq expériences. Nous avons considéré comme référence, la technique SVM pénalisé qui combine à la fois l'apprentissage et l'équilibrage de dataset. Ce classifieur est considéré comme une très bonne variante du SVM qui supporte la pondération

des classes (majoritaire et minoritaire) afin de traiter les données déséquilibrées avec plus de précision. De plus, c'est l'algorithme le plus utilisé dans la détection du cyberharcèlement. Ces expériences sont comme suit :

- Dans la première expérience, nous avons appliqué le SVM pénalisé à l'ensemble de données (avec class_weight='balanced').
- Dans la seconde expérience, nous avons utilisé le classifieur Random Forest et 100 comme nombre maximum d'estimateurs (le nombre d'arbres dans la forêt).
- Dans la troisième expérience, nous avons appliqué le classifieur Gradient Boosting avec 100 estimateurs.
- Dans la quatrième expérience, nous avons utilisé XGBoost avec ses paramètres de base sans aucun réglage, sauf le nombre d'estimateurs qu'on a fixé à 200.
- Pour la dernière expérience, nous avons utilisé AdaBoost avec Random Forest comme estimateur de base, et 100 estimateurs.

Avant de commencer les expériences, nous avons divisé le dataset en deux parties, une pour l'entraînement (80%), et l'autre pour le test (20%). Ces implémentations sont toutes effectuées avec Scikit-Learn[78].

4.3.3.3 Résultats et discussion

a. Métriques d'évaluation

Le choix de la métrique d'évaluation a de grande importance pour mesurer les performances d'un modèle surtout dans le cas de dataset déséquilibré. Comme nous l'avons déjà mentionné, notre ensemble de données souffre du problème de classes déséquilibrées : 86 % pour la classe négative et 14 % pour la classe positive. Si nous utilisons une métrique classique comme l'exactitude, nous aurons une valeur de 86% en ne prédisant que la classe majoritaire. Ce qui n'est pas vraiment précis, puisqu'elle ne représente pas les performances globales du système. Dans une telle situation, l'exactitude n'est plus une métrique appropriée.

C'est pourquoi nous avons choisi comme principale métrique la AUC (Area Under the Curve ROC) associée à la courbe ROC (Receiver Operating Characteristic). Cette métrique est

[78] Une bibliothèque de ML en python : https://scikit-learn.org

largement utilisée comme métrique d'évaluation en cas de distribution déséquilibrée de classes. La courbe ROC trace le taux de vrais positifs (TP_{Rate}) en fonction du taux de faux positifs (FP_{Rate}), en permettant de séparer le signal (TP) du bruit (FP). La AUC est la surface sous la courbe ROC, elle est considérée comme une synthèse de la courbe ROC. La AUC mesure la capacité d'un modèle à différencier entre les classes. Autrement dit, elle mesure la qualité du classement des prédictions. Plus la valeur du AUC est grande plus le modèle est capable de différencier entre les classes positives et négatives.

La AUC égale à l'intégralité de l'aire se trouvant sous l'ensemble de la courbe ROC. La formule utilisée pour son calcul est la suivante :

$$AUC = \frac{1 + TP_{Rate} - FP_{Rate}}{2} \tag{4.2}$$

TP_{Rate} ou encore la sensibilité ou le rappel du modèle, représente la proportion des échantillons positifs qui ont été correctement classés. Tandis que FP_{Rate} représente la proportion des échantillons négatifs incorrectement classés comme positifs. Ces deux métriques sont définies comme suit :

$$\mathrm{T}P_{Rate} = \frac{TP}{TP + FN} \qquad FP_{Rate} = \frac{FP}{FP + TN} \tag{4.3}$$

Nous définissons encore le TN_{Rate} aussi connu par la spécificité du modèle et qui représente la proportion des échantillons négatifs qui ont été correctement classés. Finalement le FN_{Rate} qui représente la proportion des échantillons positifs incorrectement classés comme négatifs:

$$TN_{Rate} = \frac{TN}{TN + FP} \qquad FN_{Rate} = \frac{FN}{FN + TP} \tag{4.4}$$

avec :

- TP : true positive ou vrai positif est le nombre des résultats où le modèle prédit correctement la classe positive.

- TN : true negative ou vrai négatif est le nombre des résultats où le modèle prédit correctement la classe négative.

- FP : false positive ou faux positif est le nombre des résultats où le modèle prédit incorrectement la classe positive.

- FN : false negative ou faux négatif est le nombre des résultats où le modèle prédit incorrectement la classe négative.

Par ailleurs, nous rapportons également les résultats obtenus suivant les autres métriques d'évaluation, à savoir, la précision, le rappel et le F1 score. Ceci est dans le but de gagner une vue d'ensemble sur le comportement des modèles à travers les différents critères d'évaluation. Par contre, nous n'avons pas utilisé l'exactitude (accuracy) qui ne convient pas à l'évaluation des modèles basés sur des datasets déséquilibrés, puisqu'elle favorisera toujours la classe majoritaire. Les formules des différentes métriques utilisées sont :

- La précision : mesure le taux de prédictions positives qui sont correctes (vrais positifs). C'est une mesure qualitative :

$$Précision = \frac{TP}{TP + FP} \tag{4.5}$$

- Le rappel : mesure le taux de cas positifs que le classifieur a correctement prédit, par rapport à tous les cas positifs existants dans le dataset. C'est une mesure quantitative :

$$\text{Rappel} = \frac{TP}{TP + FN} \tag{4.6}$$

- Le F1 score : c'est la moyenne harmonique de la précision et le rappel, il combine ces deux métriques d'une façon équilibrée en permettant ainsi de capturer plusieurs aspects sur la performance d'un modèle :

$$\text{F1} = 2 \times \frac{Précision \times Rappel}{Précision\ + Rappel} \tag{4.7}$$

Nous précisons que, puisqu'on a un dataset déséquilibré, alors tous les résultats reportés dans ces expériences seront donnés en '*macro average*'. En fait, le macro average d'une mesure calcule cette dernière, indépendamment, pour chaque classe et établit ensuite la moyenne. Ce qui permet d'accorder la même importance à toutes les classes. Toutes les métriques sont calculées à l'aide du module *sklearn.metrics*[79].

b. Résultats obtenus et discussion

Les résultats présentés dans cette sous-section sont relatifs à l'étude de l'efficacité des émotions en tant que caractéristiques d'apprentissage avec les classifieurs d'Ensemble, en plus de l'impact de l'incorporation de l'analyse sémantique lors de l'extraction des caractéristiques.

Nous notons qu'afin d'évaluer l'impact de SMOTE comme un prétraitement avant l'entraînement des modèles, nous l'avons appliqué à notre ensemble de données pour équilibrer la distribution de ses classes. Cependant, nous n'avons observé aucun changement significatif dans la performance des modèles, et les résultats rapportés dans les tableaux ci-dessus sont sans SMOTE. Une explication de ce comportement est, peut-être que tous ces modèles sont dédiés aux datasets déséquilibrés et ne tolèrent aucune technique supplémentaire d'équilibrage des données.

Tableau 4.3 Matrice de confusion du XGBOOST

		Prédiction	
		Violent	**Non Violent**
Actuel	**Violent**	1.30% (**TP_{rate}**)	11.45% (**FN_{rate}**)
	Non Violent	0.80% (**FP_{rate}**)	86.43% (**TN_{rate}**)

Le premier résultat présenté est la matrice de confusion du classifieur le plus performant XGBoost. Comme indiqué sur le Tableau 4.3, le taux de TN égal à 86,43%, ce qui est un bon indicateur de la performance du modèle. Tandis que, le taux de TP a eu une faible valeur. Ce résultat est dû au dataset déséquilibré, où la majorité des tweets sont étiquetés par le label *non violents* contre une minorité étiquetés par *violents*. Pour cela, et comme nous l'avons expliqué, nous ne pouvons pas baser notre évaluation sur des simples métriques, d'autre

[79] https://scikit-learn.org/stable/modules/model_evaluation.html#classification-metrics

seront également utilisées, et en particulier la AUC.

Pour mieux comprendre le comportement du classifieur XGBoost, il est intéressant de visualiser sa courbe ROC.

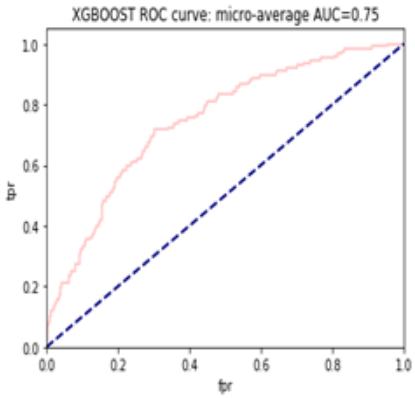

Figure 4.3 Courbe ROC XGBOOST

Sur la Figure 4.3, nous représentons la courbe ROC de ce classifieur. La ligne pointillée représente un classifieur aléatoire où le TP_{rate} est égale au FP_{rate} (AUC = 0.5). Tous les points en dessus de cette ligne correspondent à la situation où la proportion d'échantillons correctement classés, appartenant à la classe positive, est supérieure à la proportion d'échantillons incorrectement classés appartenant à la classe négative. Ce qui indique une bonne sensibilité du modèle.

Concernant l'évaluation des classifieurs d'Ensemble choisis, le Tableau 4.4 montre les résultats obtenus à partir des expériences où l'on compare leurs performances, y compris l'analyse sémantique.

Tableau 4.4: Résultats des classifieurs avec la similarité sémantique

Classifieur	***AUC***	***Précision***	***Rappel***	***F1***
SVM Pénalisé	0.53	0.42	0.50	0.45
Random forest	0.71	**0.78**	0.58	**0.66**
Gradient Boosting	0.73	0.73	0.54	0.62
XGBoost	**0.75**	0.74	0.55	0.63
Adaboost	0.72	0.71	**0.59**	0.64

Comme le montre ce tableau, XGBoost présente les meilleurs résultats en termes d'AUC avec une valeur de 0.75. Le deuxième meilleur classifieur est Gradient Boosting avec un score AUC de 0.73, puis Adaboost avec 0.72. Parmi les cinq classifieurs, le SVM pénalisé a obtenu les plus faibles résultats dans toutes les métriques.

Pour prouver l'efficacité de l'analyse sémantiques, nous présentons sur le Tableau 4.5 les résultats des mêmes classifieurs, mais cette fois, en adoptant uniquement la similarité lexicale pendant l'extraction des caractéristiques.

Tableau 4.5: Résultats des classifieurs avec la similarité lexicale

Classifieur	***AUC***	***Précision***	***Rappel***	***F1***
SVM Pénalisé	0.56	0.42	0.50	0.45
Random forest	0.67	0.69	0.55	0.61
Gradient Boosting	**0.73**	0.64	0.52	0.57
XGBoost	0.72	**0.72**	0.55	**0.62**
Adaboost	0.69	0.67	**0.57**	0.61

En comparant les résultats des deux tableaux Tableau 4.4 et Tableau 4.5, nous pouvons confirmer que l'incorporation de l'analyse sémantique a amélioré les performances de tous les classifieurs en termes de précision et de rappel. Les classifieurs Random Forest, Gradient Boosting et Adaboost avec analyse sémantique donnent de meilleurs résultats dans toutes les métriques. Le même fait s'est produit avec la AUC de XGBoost qui est devenue le meilleur score. Cependant, pour le SVM pénalisé, l'analyse sémantique a détérioré les performances de la AUC et la précision, et n'a presque pas eu d'influence sur les autres métriques.

Finalement, afin de définir les caractéristiques les plus pertinentes qui peuvent être considérées comme de puissants prédicteurs du comportement violent d'un utilisateur en ligne, nous avons visualisé les caractéristiques importantes de XGBoost (Figure 4.4). Cette visualisation souligne que l'anticipation et la colère sont les facteurs les plus influents.

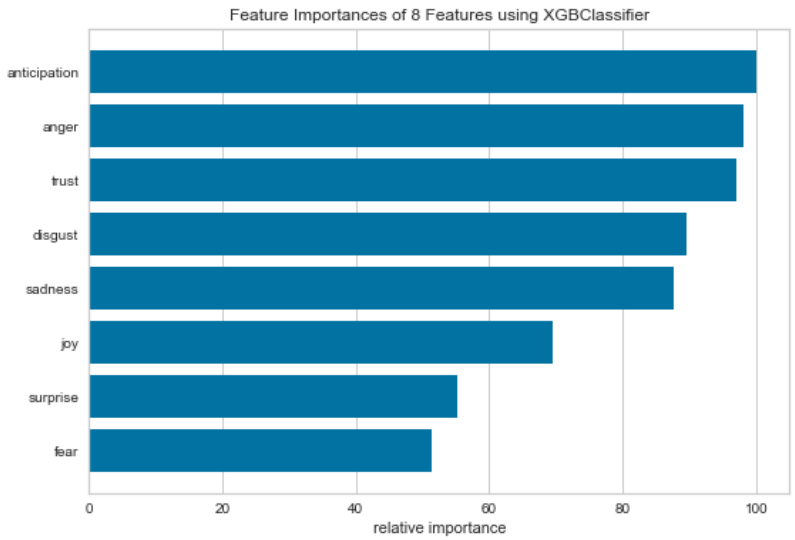

Figure 4.4 Importance des caractéristiques relatives aux émotions dans la prédiction du contenu violent avec le classifieur XGBoost

En résumé, nous avons entraîné des modèles d'Ensemble ML sur des caractéristiques liées aux émotions des auteurs des messages. Vu la distribution déséquilibrée des classes dans notre dataset, nous avons évalué les performances de notre système avec la métrique AUC. Pendant l'évaluation, nous avons comparé les deux types d'extraction de caractéristiques, lexicale et sémantique. Également, nous avons comparé les quatre modèles d'Ensemble ML, les meilleures performances ont été obtenues par XGBoost. L'analyse sémantique a aussi prouvé sa capacité à augmenter les performances globales de la plupart des classifieurs. Ces résultats affirment clairement l'association entre l'état émotionnel de l'auteur et son comportement violent. Cela étant, les caractéristiques psychologiques de l'utilisateur extraites du textes générés par ce dernier peuvent être de bons indicateurs de la nature de son comportement en ligne.

Dans la suite, nous allons explorer d'autres caractéristiques liées à la personnalité des perpétrateurs qui peuvent être un facteur prédictif puissant de son comportement violent.

4.4 Détection des traits de violence à l'aide des Big Five traits de personnalité

Dans cette section, nous étudions une autre caractéristique psychologique importante du perpétrateur de l'acte de violence, à savoir les traits de personnalité. Ces derniers n'ont pas été suffisamment abordés dans la littérature, même si elles constituent un critère essentiel dans les études de psychologie sur la cyberviolence (Van Geel et al., 2016). Par conséquent,

ce travail explorera la corrélation des traits de personnalité du cyber-auteur avec son comportement violent, qui peut être un indicateur puissant dans la détection de l'acte de cyberviolence.

Dans cette section, nous présentons notre approche pour la détection des comportements violents basée sur les cinq grandes facettes de la personnalité connues par le modèle Big Five, suivie des différents tests conduits et leurs résultats.

4.4.1 Les Big Five traits de personnalité

Les Big Five traits de personnalité est un modèle descriptif de personnalité proposé par de nombreux chercheurs en psychologie. En particulier Lewis Goldberg qui l'a premièrement introduit dans (Goldberg, 1980) et qui a été validé par McCrae & Costa dans (Mccrae and Costa, 1987). Par la suite, le modèle devient l'objet de plusieurs travaux (Johnson, 2017). Ces efforts étaient élaborés dans le but de réduire les modèles de personnalité existants en se concentrant sur les traits essentiels qui peuvent décrire la personnalité.

Ce modèle a été nommé ainsi, parce qu'il propose de mesurer la personnalité des individus selon cinq dimensions majeures, indépendantes les unes des autres. Les cinq dimensions de ce modèle conceptuel sont : l'ouverture, la conscience, l'extraversion, l'agréabilité et le neuroticisme. La description de ces traits est présentée sur la Figure 4.5.

Chacun des cinq grands traits de personnalité représente des catégories extrêmement larges qui couvrent de nombreux termes liés à la personnalité. Chaque trait englobe une multitude d'autres facettes. Par exemple, le trait conscience inclut les facettes : compétence, organisation, conscience professionnelle, autodiscipline, délibération, recherche de la réussite (Lim, 2020).

En psychologie, Big Five est la théorie la plus acceptée en littérature en pratique (tests psychométriques). Son objectif est d'évaluer la personne par rapport à chacun des cinq traits et construire ainsi une idée approximative sur sa personnalité.

Un autre aspect important du modèle de Big Five est son approche d'évaluation de la personnalité. Il considère la conceptualisation des traits comme un spectre, plutôt que comme des catégories noires et blanches. Chaque trait de personnalité est un spectre et les individus sont classés sur une échelle entre ces deux extrémités (Lim, 2020). Par exemple, lorsqu'on

Figure 4.5. Big Five traits de Personnalité

mesure l'extraversion, une personne n'est pas qualifiée de purement extravertie ou introvertie, mais placée sur une échelle déterminant son niveau d'extraversion.

Pouvoir décrire la personnalité s'avère très utile surtout lorsqu'il s'agit d'expliquer ou de décrire des phénomènes psychologiques. Nous citons à titre d'exemples les troubles mentaux chez certains individus, ou encore le comportement violent en ligne qui fait l'objet de ce travail. Comme nous l'avons évoqué dans l'état d'art, plusieurs études en psychologie ont montré l'association entre les actes de violence avec les Big Five. Alors, nous allons exploiter cette évidence, en utilisant ces facteurs comme caractéristiques d'apprentissage pour détecter le contenu violent dans les messages générés par les utilisateurs des SM.

4.4.2 Méthode

Puisque nous avons opté pour les techniques ML classiques, le processus de détection du contenu violent se déroulera sur deux phases : l'extraction des caractéristiques liées aux traits Big Five, suivies par l'apprentissage du modèle de prédiction.

4.4.2.1 Extraction des caractéristiques

Nous avons choisi les Big Five traits de personnalité (l'ouverture, la conscience, l'extraversion, l'agréabilité et le neuroticisme) comme caractéristiques d'apprentissage. Afin

d'extraire les caractéristiques linguistiques (mots, expressions, symboles) associées à ces cinq traits, nous avons adopté l'approche du vocabulaire ouvert. Comme nous l'avons déjà expliqué, dans cette approche, les caractéristiques linguistiques sont automatiquement identifiées et extraites à partir des textes écrits par les utilisateurs. En d'autres termes, le vocabulaire ouvert est construit à partir des médias sociaux. Alors, il est de nature plus proche du langage utilisé dans ces médias, surtout par les adolescents, qui représentent la catégorie la plus vulnérable à la cyberviolence. Finalement, ce langage est continuellement mis à jour, ce qui aura une répercussion sur la phase d'extraction et donc sur l'ensemble du processus de la détection de la cyberviolence. Ce qui fait que l'approche du vocabulaire ouvert reste la plus adaptée à ce cas d'utilisation. En revanche, le lexique spécial, qui se limite à des listes de mots prédéfinis, ne peut donc pas assurer une large couverture du lexique utilisé dans chacun des traits de personnalité.

Afin d'élargir la couverture de ce lexique, nous avons utilisé une technique de renforcement de lexique basée sur la similarité sémantique à l'aide du word embedding. Pour chaque mot, dans chacun des traits Big Five, nous avons extrait les dix mots les plus similaires. Ces mots ont été ensuite ajoutés aux lexiques appropriés en préservant la même pondération d'origine. Cette opération nous a permis de passer d'un lexique de 1000 à 9350 mots. Le Tableau 4.6 montre des exemples d'enrichissement de lexique du trait d'agréabilité.

Tableau 4.6 Exemples d'enrichissement du lexique du trait d'agréabilité produits à l'aide du word embedding

Lexique d'origine	Lexique enrichi
Wonderful	Marvelous, fantastic, great, fabulous, terrific, lovely, amazing, beautiful, magnificent
So excited	Thrilled, pleased, enthused, delighted, ecstatic, proud, psyched, exciting, elated

Après l'augmentation du lexique, nous avons entamé l'extraction des caractéristiques relatives aux traits Big Five à partir des messages composant le dataset. Pour ce faire, et en plus de la correspondance lexicale, nous avons utilisé la similarité sémantique avec le word embedding. L'objectif est de mieux contextualiser le processus de correspondance entre les mots composants le lexique et ceux composants les messages (tweets). La technique que nous avons employée est identique à celle décrite dans la section précédente.

4.4.2.2 Algorithmes d'apprentissage

Parallèlement avec la section précédente, et après l'extraction des caractéristiques, nous avons appliqué les mêmes techniques d'apprentissage supervisé pour la prédiction du comportement violent : Random Forest, Gradient Boosting, XGBoost et Adaboost.

4.4.3 Évaluation

L'évaluation de notre système a été faite à travers une série d'expériences dont l'objectif est de valider notre approche en testant le fonctionnement de ses différents constituants. Cette évaluation touche, en premier lieu, l'impact des cinq dimensions de la personnalité, combinées aux classifieurs d'Ensemble, sur la détection du contenu violent. En deuxième lieu, elle touche l'effet de l'extension du lexique de Big Five sur la même tâche. Les détails sur les ressources, les outils et les métriques utilisées ainsi que les résultats obtenus seront donnés ci-après.

4.4.3.1 Ressources

Dans cette section, nous présentons les ressources utilisées dans nos expériences en termes de lexiques et de dataset.

c. Lexiques.

Concernant le lexique lié aux Big Five traits de personnalité (l'ouverture, la conscience, l'extraversion, l'agréabilité et le neuroticisme) nécessaire à la phase d'apprentissage des modèles, nous avons utilisé le lexique[80] élaboré dans les travaux de Schwartz et al. (2013). Ce lexique a été construit à partir d'un corpus Facebook de 309 millions mots, collecté via une application de test de personnalité basée sur un questionnaire. Puis, les mots et les expressions ont été extraits de ce corpus. Ensuite, à travers la modélisation thématique (Topic Modeling) avec la technique LDA (Latent Dirichlet Allocation), un ensemble de thèmes a été généré. Finalement, une analyse corrélationnelle a été appliquée afin de joindre des ensembles de vocabulaires (mots, expressions et thèmes) au trait Big Five approprié. Ces vocabulaires sont pondérés en fonction de leur fréquence d'utilisation dans le trait correspondant. Le lexique original final comprend 200 entrées pour chacun des cinq traits.

[80] https://wwbp.org/data.html

Tableau 4.7: Exemple du lexique d'agréabilité avec les poids correspondants

Mot	***Poids***	***Type de corrélation avec le trait d'agréabilité***
amazing a great blessed	0.056682 0.056981 0.057403	Corrélation positive
fuck fucking	-0.120624 -0.113133	Corrélation négative

Le Tableau 4.7 montre un exemple du lexique de mots associé au trait *agréabilité*. Dans cet exemple, les mots positivement liés à l'*agréabilité* ont été attribués un poids positif, tandis que les mots négativement corrélés avec ce trait ont eu un poids négatif.

Comme nous l'avons mentionné précédemment, et afin d'étendre sa couverture, chaque mot de ce lexique a été renforcé par les mots les plus similaires extraits du vocabulaire généré par le modèle Word2Vec. Le lexique renforcé final est composé de 10000 entrées.

d. Dataset

De nouveau, nous appliquons notre approche sur le même dataset « racial » que nous divisons à 80% pour l'entraînement et 20% pour le test.

4.4.3.2 Expériences

Les expériences menées visent à examiner les performances des différentes techniques ML avec les caractéristiques relatives à la personnalité.

a. Tests de validation de l'extension du lexique Big Five

Comme nous l'avons déjà expliqué, nous avons utilisé un lexique lié aux Big Five traits de personnalité. Nous avons enrichi ce lexique en utilisant le word embedding, afin d'étendre sa couverture et améliorer par conséquent la correspondance sémantique. Pendant ce processus de correspondance, nous employons un modèle Word2vec pour faire correspondre les mots des messages aux mots du lexique Big Five. Le modèle Word2vec utilisé est celui entraîné sur un dataset de Google News.

b. Tests de validation du choix de modèle

Coté algorithmes, nous avons conduit cinq expériences, chacune avec un algorithme ML : Random Forest, Gradient Boosting, XGBoost et AdaBoost. De plus, nous nous sommes servis du SVM pénalisé comme référence. Nous avons conservé les mêmes paramètres des classifieurs que nous avons employés pendant les expériences de la section précédente.

Nous notons que toutes nos méthodes ML ont été implémentées à l'aide de la bibliothèque Scikit-Learn.

4.4.3.3 Résultats et discussion

c. Métriques d'évaluation.

Pareil, dans cette évaluation nous utilisons les mêmes métriques précédentes, soit la AUC appropriée aux distributions de classe déséquilibrée et également les métriques précision, rappel et F1 score en macro-average.

d. Résultats obtenus et discussion

Le Tableau 4.8 montre les résultats obtenus à partir des cinq expériences, où nous comparons les performances des cinq classifieurs avec le renforcement du lexique.

Tableau 4.8: Résultats des performances des classifieurs avec renforcement de lexique

Algorithme ML	***AUC***	***Précision***	***Rappel***	***F1***
SVM pénalisé	0.5	0.79	0.54	0.64
Random Forest	**0.73**	0.77	0.64	0.69
Gradient Boosting	0.71	0.75	0.52	0.61
XGBoost	0.72	0.65	**0.65**	0.65
AdaBoost	0.72	**0.82**	0.61	**0.69**

Comme le montre ce tableau, le meilleur score AUC a été atteint par Random Forest (0.73). AdaBoost atteint les meilleurs résultats en termes de précision et F1 (0.82, 0.69 respectivement). Random Forest a surpassé XGBoost dans toutes les métriques, sauf pour le rappel (0.65 pour XGBoost contre 0.64 pour Random Forest), qui a été le plus élevé parmi tous les autres classifieurs. Enfin, entre les cinq classifieurs, le SVM pénalisé a donné les plus

faibles résultats dans toutes les métriques, sauf la précision (0.79) où SVM a été classé deuxième.

Quant à la validation de l'efficacité du renforcement du lexique Big Five, nous reportons les résultats du test des classifieurs sans renforcement dans le Tableau 4.9.

Dans l'ensemble, les résultats obtenus confirment l'impact positif du renforcement du lexique. Tous les classifieurs ont montré de faibles performances dans toutes les métriques, sauf le gradient Boosting qui a marqué 0.92 de précision, qui peut être dû à un overfitting.

Tableau 4.9: Résultats des performances des classifieurs sans renforcement de lexique

Algorithme ML	***AUC***	***Précision***	***Rappel***	***F1***
SVM pénalisé	0.5	0.31	0.52	0.38
Random Forest	0.62	0.75	0.58	0.65
Gradient Boosting	0.60	**0.92**	0.50	0.64
XGBoost	0.61	0.58	**0.58**	0.58
AdaBoost	**0.63**	0.81	0.56	**0.66**

Également, nous avons analysé les caractéristiques d'apprentissage les plus importantes relativement aux Big Five traits de personnalité utilisées avec le modèle XGBoost.

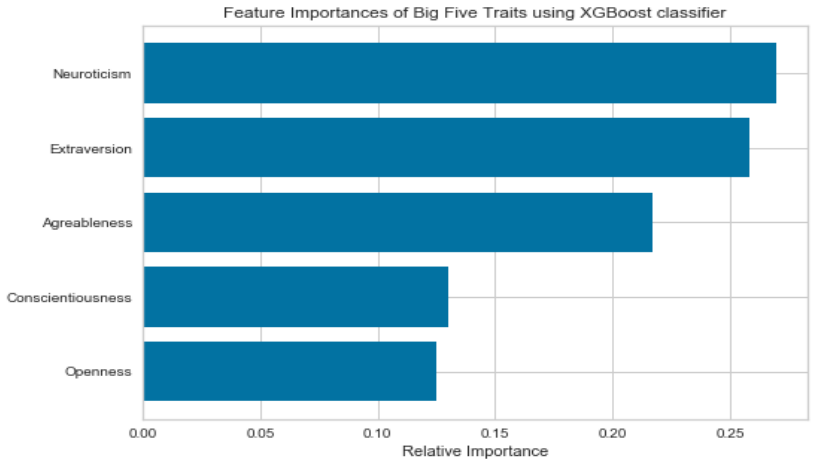

Figure 4.6. Importance des caractéristiques Big Five traits de personnalité dans la prédiction du contenu violent avec le classifieur XGBoost

La Figure 4.6 montrent clairement que le trait neuroticisme, suivi par l'extraversion, sont les facteurs les plus influents dans la détection du contenu violent.

En résumé, Ensemble ML a prouvé son bon fonctionnement dans le cas de dataset déséquilibré. La meilleure performance est venue en faveur d'AdaBoost et Random Forest, suivi de XGBoost. Le renforcement du lexique a également prouvé sa capacité à augmenter les performances globales de la plupart des classifieurs. Evidemment, ces résultats affirment la forte association entre les traits de personnalité de l'auteur et son comportement violent. Ce qui implique que les caractéristiques psychologiques du cyber-auteur, extraites des textes rédigés par ce dernier, peuvent être un bon signe de son comportement en ligne.

4.5 Combinaison des émotions et des Traits Big Five

Pendant les expériences précédentes nous avons testé la détection du contenu violent fondée sur des algorithmes Ensemble ML avec des caractéristiques basées sur les émotions et les Big Five traits de personnalité séparément. Dans cette expérience nous combinerons ces deux dites caractéristiques pour évaluer leur influence sur cette tâche. Les résultats sont présentés sur le Tableau 4.10 montrant une hausse en performance surtout en AUC qui a atteint 0.80, ce qui signifie que la combinaison des caractères de personnalité a été plus efficace dans le processus de la détection des contenus violents.

Tableau 4.10 : Résultats des performances des classifieurs avec la combinaison des émotions et les traits Big Five

Algorithme ML	***AUC***	***Précision***	***Rappel***	***F1***
SVM pénalisé	0.38	0.13	0.5	0.20
Random Forest	0.73	0.70	0.54	0.60
Gradient Boosting	0.79	0.76	0.57	0.65
XGBoost	**0.80**	0.77	**0.59**	**0.66**
AdaBoost	0.74	**0.79**	0.56	0.65

En vue d'évaluer d'avantage nos caractéristiques d'apprentissage basées sur les caractères de la personnalité, nous avons entraîné les mêmes classifieurs d'ensemble sur des bi-grammes et des gros mots[81]. Ces caractéristiques font partie des caractéristiques les plus employées dans la littérature de cette tâche.

[81] http://www.cs.cmu.edu/~biglou/resources/bad-words.txt

Comme c'est montré sur le Tableau 4.11, le classifieur Adaboost a obtenu les meilleurs résultats en termes de AUC et de rappel. En général ces résultats ont été nettement bons. Ce qui signifie que les bi-grammes et les mots grossiers peuvent être de bonnes caractéristiques pour la détection des contenus violents. Toutefois, les résultats obtenus avec la combinaison des émotions et des traits Big Five restent les meilleurs, ce qui montre que ces derniers sont plus efficaces dans la détection des contenus violents.

Tableau 4.11 Résultats des classifieurs avec des bi-grammes et des mots grossiers comme caractéristiques

Classifieur	***AUC***	***Précision***	***Rappel***	***F1***
SVM Pénalisé	0.34	0.06	0.50	0.10
Random forest	0.66	0.56	0.61	0.58
Gradient Boosting	0.62	**0.93**	0.54	**0.68**
XGBoost	0.65	0.87	0.52	0.65
Adaboost	**0.74**	0.60	**0.68**	0.63

4.6 Détection des traits de violence à l'aide des Techniques Deep Learning

Comme nous l'avons évoqué dans l'état d'art, les techniques deep learning qui reposent sur l'utilisation de larges datasets annotés, sont récemment devenues très répandues vu leur bonne performance assurée dans plusieurs tâches NLP (Yoav Goldberg, 2017). Ce qui encourage les chercheurs à les employer de plus en plus dans de nouvelles tâches. Elles ont été aussi employées pour la détection du contenu violent, où ils ont donné de bons résultats. Ce qui nous a incité à les explorer dans notre tâche afin de comparer nos résultats obtenus avec les algorithmes ML classiques basés sur l'apprentissage à travers des caractéristiques bien définies. Dans la suite, nous présenterons les expériences menées avec les architectures DL pour lesquelles nous avons opté, à savoir CNN, RNN et BERT, ainsi que leurs configurations et les résultats obtenus.

4.6.1 Evaluation

4.6.1.1 Dataset

Nous avons appliqué notre approche sur le même dataset « racial » que nous avons divisé

à 70% pour l'entraînement, 20% pour la validation et 10% pour le test. Les données d'entraînement sont utilisées pour ajuster les poids du réseau neuronal. Celles de validation sont utilisées pour détecter le sur-apprentissage pendant l'entraînement. En effet, si l'exactitude (accuracy) par rapport aux données d'entraînement augmente, et que l'exactitude par rapport aux données de validation reste la même ou diminue, alors, le réseau est sur-ajusté et l'entraînement doit être interrompue. Les données de test sont utilisées pour tester le modèle final afin de confirmer le pouvoir prédictif réel du réseau.

4.6.1.2 Expériences

Le processus de classification basée sur le DL repose sur trois étapes principales, qui sont la préparation des données, la construction et l'entraînement du modèle et le test de prédiction.

a. Classification avec CNN et RNN

Pendant la phase de préparation de données, le texte d'entrée composé de tweets a subi les traitements suivants :

- Premièrement le nettoyage pour éliminer les marqueurs de ponctuation, les URLs, les mentions @, et les mots vides (stop words).
- La tokenisation pour transformer chaque tweet en séquence de mots séparés. Puis, l'encodage des tokens générés afin d'associer chaque token à une valeur numérique qui lui servira comme index. Cette opération fournit une séquence de nombres entiers.
- Le remplissage (padding) qui sert à standardiser la longueur des tweets en ajoutant des zéros à la fin de chaque séquence. Dans notre cas, nous avons fixé à 100 la longueur de la séquence.
- La dernière étape consiste à convertir les tokens en vecteurs denses de taille fixe à l'aide du word embedding. Le word embedding définit la forme initiale de représentation des données nécessaire pour la couche d'entrée du réseau. La matrice d'embedding (l'ensemble des vecteurs des mots) sera passée comme paramètres dans la couche d'entrée.

Puisque notre texte d'entrée est composé de tweets, nous avons utilisé un modèle Glove

entraîné sur un corpus de Twitter, pour que l'embedding des mots soit plus représentatif. La taille de l'embedding choisi est de 200 dimensions, ce qui veut dire que chaque mot sera codé sous la forme d'un vecteur de 200 valeurs.

Quant à la construction des modèles, ceux-ci sont composés de la même couche d'entrée et de la même couche de sortie. Pour les couches cachées intermédiaires, nous avons adopté deux architectures, une basée sur le CNN et l'autre sur le RNN.

Le réseau CNN a été construit comme suit :

- La couche d'entrée, de type embedding, reçoit la matrice du word embedding préalablement générée.
- L'embedding est introduit dans un CNN (1D) avec 128 noyaux de convolution, suivi par une deuxième couche Conv1D de 64 noyaux, puis une couche pooling, toutes séparées par des taux de dropout de 0.3.
- La couche de sortie, composée d'une seule unité, fait usage d'une fonction d'activation '*sigmoïde*' pour fournir des valeurs de probabilités comprise entre 0 et 1. Plus ces valeurs sont proches de 0, plus le contenu du tweet est non violent, et inversement plus ces valeurs sont proches de 1, plus le contenu est violent.

Dans le cas de RNN, nous avons utilisé un BLSTM en deux couches, séparées par des dropout[82] d'un taux de 0.3. Ce modèle est composé de trois couches de base, qui sont :

- Une couche d'embedding à 300 dimensions.
- Deux couches BLSTM de 128 et 64 unités respectivement.
- Une couche dense pour récupérer les résultats avec une seule unité et une fonction d'activation sigmoïde.

Nous avons également testé une hybridation de réseaux CNN et BLSTM. Nous avons connecté une couche Conv1D de 128 unités avec une couche BLSTM de type GRU composée de 64 unités dont la sortie est injectée dans une couche de pooling.

[82] Dropout ratio : le taux d'abandon de neurones qu'est le ratio d'unités cachées à désactiver dans chaque entrainement des lots.

Concernant l'implémentation, nous avons utilisé la bibliothèque TensorFlow[83] et particulièrement l'API Keras[84].

b. Classification avec BERT

Certes BERT est basé sur l'architecture Transformers, cependant il est composé uniquement d'encodeurs dont la fonction est le traitement de texte et la génération des embeddings, tandis que le décodage est fait par le module de sortie chargé de prédiction. Ce modèle a été employé dans la classification du texte avec un fine tuning pour l'adapter à des tâches spécifiques puisqu'il a été pré-entraîné sur un domaine général. Dans ce cas les sorties de BERT sont injectées à l'entrée d'un autre réseau neuronale, puis l'entraînement est lancé sur un dataset approprié. Les étapes suivies pour classifier le contenu des tweets sont :

- Le pré-traitement du texte d'entrée : l'objectif est de générer les trois types d'entrées requises par BERT. La première entrée est constituée des identificateurs des tokens. Pour les générer, le texte du dataset passe d'abord à travers la tokenisation et le padding. BERT utilise son propre tokenizer pré-entraîné basé sur WordPiece (Wu et al., 2016). Le principe de ce tokenizer consiste à décomposer certains mots en sous-mots, ce qui permet de décomposer des mots inconnus en d'autres connus. Ces tokens seront ensuite convertis en identificateurs formant une liste d'entiers liés de manière unique à un mot spécifique. La deuxième donnée d'entrée est le masque d'attention permettant de différencier entre le contenu et le padding. La dernière entrée est l'identificateur de type d'entrée indiquant à quelle phrase chaque token appartient. Finalement, ces trois types d'entrées seront transformées en *tenseurs*[85]
- La combinaison de BERT avec un classifieur : l'embedding généré par BERT sera injecté dans un module de classification. Le classifieur utilisé dans cette expérience est constitué d'une couche dropout suivie d'une couche dense de sortie.
- L'entraînement du modèle sur le dataset de violence.

Dans cette expérience, nous avons employé le modèle 'BERT-base-uncased[86]' (12

[83] TensorFlow est une implémentation de librairies deep learning open source développée chez Google.
[84] https://keras.io/api/
[85] Les tenseurs sont des tableaux multidimensionnels avec un type uniforme. C'est la structure de données adoptée par les systèmes DL. (https://www.tensorflow.org/guide/tensor)
[86] Uncased signifie que le modèle ne fait pas la distinction entre les lettres en majuscules et celles en minuscules.

couches d'encodeurs, 12 attention heads, et 110 million de paramètres). En plus de RoBERTa-base, étant une version optimisée de BERT et Twitter RoBERTa, étant un modèle affiné pour la détection du langage offensif. L'implémentation a été faite avec la librairie *transformers* développé par Hugging Face[87].

Dans tous les modèles, nous avons utilisé la fonction d'optimisation *'adam'*, et la fonction de perte *'binary_crossentropy'*, puisque nous visons une classification binaire. Le réseau a été entraîné sur 10 époques[88] avec un batch_size[89] égale à 100.

4.6.1.3 Résultats et discussion

Nous avons conduit deux expériences, la première est effectuée avec les données initiales, tandis que la seconde est effectuée en appliquant la technique d'équilibrage SMOTE. La SMOTE fonctionne en augmentant le nombre d'échantillons en faveur de la classe minoritaire, cette opération a permis de passer de 4998 à 6378 échantillons, donc une augmentation de 27%. Les résultats des expériences menues sont présentés sur le Tableau 4.12. Nous rappelons que les mesures sont en macro-average.

Tableau 4.12 résultats des tests réalisés avec les techniques DL : RNN, CNN et BERT

Algorithme DL	***AUC***	***Précision***	***Rappel***	***F1***
CNN	**0.53**	0.43	0.50	0.46
CNN + SMOTE	0.48	0.48	0.47	0.47
BLTSM	0.49	0.43	0.50	0.46
BLSTM + SMOTE	0.51	0.49	0.49	0.49
CNN + BLSTM (GRU)	0.42	0.43	0.50	0.46
CNN + BLSTM (GRU) + SMOTE	0.52	0.52	0.52	0.52
BERT	0.57	0.58	0.62	0.59
RoBERTa	-	0.43	0.50	0.46
Twitter RoBERTa	**0.67**	**0.67**	**0.77**	**0.71**

[87] https://huggingface.co
[88] Le nombre d'époques : le nombre d'itérations sur l'ensemble d'entraînement.
[89] Batch_size : la taille du lot qu'est le nombre d'instances d'entraînement à considérer en même temps.

D'après ces résultats, nous pouvons aboutir à deux conclusions. La première est que la technique SMOTE a légèrement amélioré les performances des deux modèles, ceux-ci restent très comparables. Pour la deuxième conclusion, bien que les techniques DL sont connues par leur haute performance dans plusieurs tâches NLP, elles nécessitent en revanche, de grande quantité de données pour atteindre de telle performance. Ce constat a été confirmé par les faibles valeurs des différentes métriques surtout avec le dataset initial, contrairement aux techniques ML qui peuvent aboutir à de bons résultats même avec un dataset de taille réduite. Alors, pour mieux explorer les capacités des techniques DL dans la détection des contenus violents, nous aborderons un dataset plus large dans un travail futur. En ce qui concerne les modèles Transformer, comme ils ont été entraînés sur un corpus volumineux de domaine général, ils doivent être affinés sur des domaines spécifiques pour fournir de meilleurs résultats.

4.7 Dataset de violence pour l'Arabe Marocain

Pendant notre étude sur la détection de violence, nous avons validé nos solutions sur des datasets en Anglais. Mais, nous n'avons pas pu appliquer ces solutions sur un texte écrit en Arabe Marocain. Puisqu'un dataset en Arabe Marocain annoté pour la violence en ligne n'existait pas encore. Afin de franchir cet obstacle, nous avons décidé de construire notre propre dataset dédié à la détection des traits de violence chez les Marocains. Pour ce faire, nous avons collecté des commentaires YouToube, puis à travers une application web et à l'aide d'une vingtaine de volontaires, nous sommes arrivés à collecter et annoter un corpus de violence de 23k commentaires. Les détails de cette opération seront présentés dans ce qui suit.

Il est à noter que dans notre dataset, nous considérons tout texte contenant un langage nocif en tant que violent, indépendamment des terminologies adoptées dans la littérature (offensif, toxique, abusif, agressif, harcelant, dangereux...).

4.7.1 Collecte du texte

Vu la popularité de YouTube au Maroc, nous l'avons choisi comme source de données pour notre dataset de violence. Cette plateforme procure 46,39% de l'activité des marocains sur le web. En outre, l'extraction de textes à partir de YouTube est ouverte et gratuite pour tout le monde et sans restriction.

Nous avons remarqué que la majorité des datasets consultés pendant ce travail souffre d'un déséquilibre entre les deux classes '*violent*' et '*non violent*', par exemple dans (Rezvan et al., 2020) 14.8% des tweets sont harassants et dans (Davidson et al., 2017), seulement 3% des tweets annotés contenaient un discours de haine. Ce qui peut engendrer un problème de surapprentissage lors de l'entraînement et donc il faut intervenir pour équilibrer le dataset en moyennant les techniques présentées précédemment. Afin d'éviter ce problème et produire un dataset contenant suffisamment d'instances pour chaque classe, il faut au préalable, c'est-à-dire lors de la collecte, considérer l'équilibre des distributions entre les deux classes. Pour ce faire, nous avons cherché des thèmes liés à la violence pour garantir qu'une partie importante des commentaires contiendra des contenus violents.

L'extraction des commentaires a été faite à l'aide de l'API YouTube et un script qui prend en entrée des mots clés pour accomplir sa recherche. Nous avons utilisé des mots clés relatifs

à des actes de violence et d'autres ordinaire que nous avons sélectionnés depuis les titres des vidéos dans des chaînes populaires. Chaque mot clé utilisé retourne les commentaires des cinq premières vidéos trouvées en résultat. Pour le contenu violent, nous avons choisi des chaînes comme MC Talib, Kifache tv et Aldar.ma, tandis que pour le contenu non violent, nous avons opté pour des chaînes comme Mustapha Swinga officiel et Najib El Mokhtari. Nous avons également veillé à diversifier les thèmes des vidéos afin de capturer les différents contenus violents qui peuvent leur être associés : sport, cuisine, vie sociale, comédie, politique et autres.

Il est intéressant de noter que le corpus collecté comprend des commentaires écrits en script Arabe et aussi en Latin. Ce qui nécessitera une phase de normalisation avant son utilisation par les technique ML.

Après l'extraction du corpus, il a été nettoyé pour en exclure les commentaires vides, composés seulement de URLs, hashtags, chiffres, ainsi que les répétitions. L'étape suivante était la conception de l'application web qui servira comme interface d'annotation.

4.7.2 L'annotation

L'annotation a été faite à travers une application web développée avec le Framework python Django. Quant au déploiement, nous avons bénéficié d'un hébergement gratuit pendant une année sur *pythonAnywhere*[90] qui offre un environnement intégré de développement web, du code jusqu'au hébergement.

Dans cette phase, nous avons annoté les 23 k commentaires avec l'aide de 17 volontaires, du genre féminin et masculin, dont l'âge varie entre 20 et 40 ans. L'annotation s'est déroulée sur une période d'une année. Ces annotateurs viennent de différentes villes marocaines (Casablanca, Fès, Salé). Chaque annotateur avait la possibilité d'annoter autant de commentaires suivant un ordre aléatoire. Nous avons demandé aux annotateurs de sélectionner pour chaque commentaire proposé le label '***Oui'*** s'il contient du contenu violent,

[90] https://www.pythonanywhere.com/

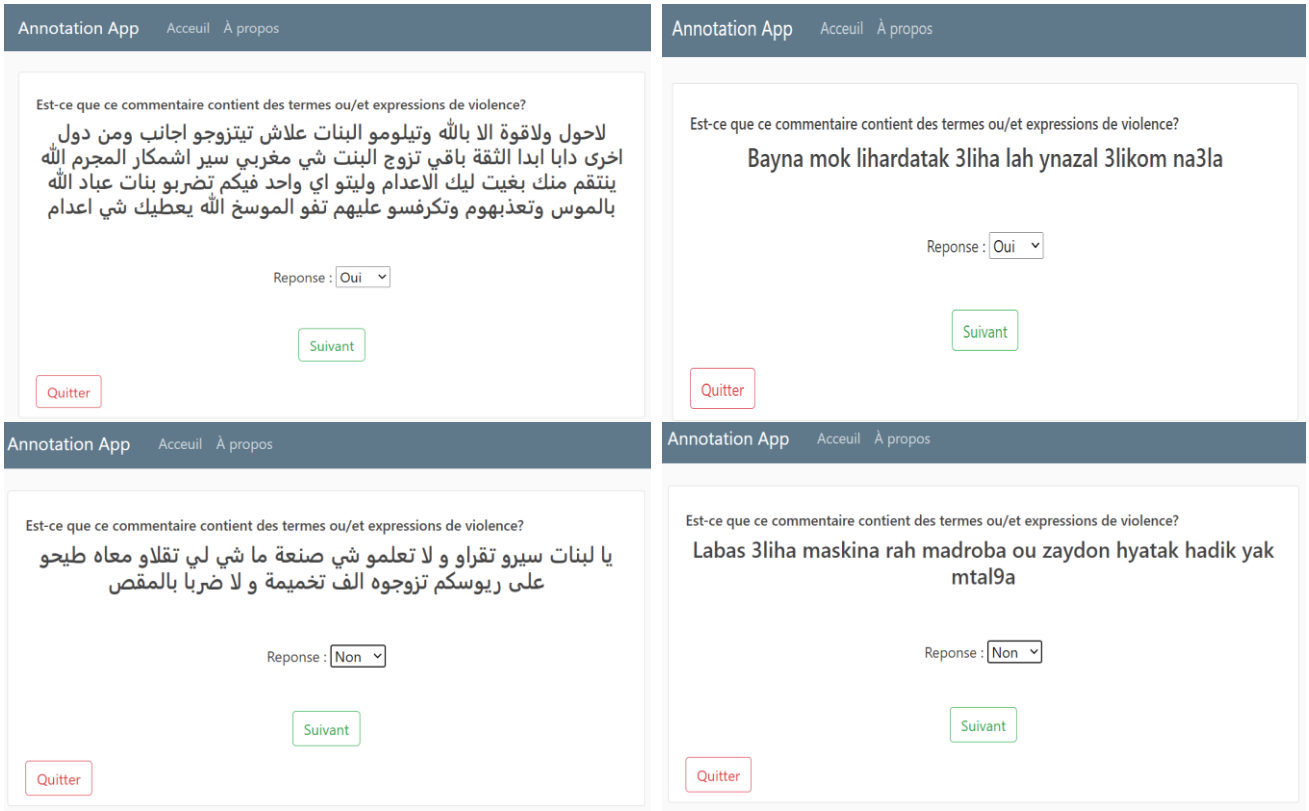

Figure 4.7. Exemples d'annotation des commentaires YouTube. Le label 'Oui' est donné aux commentaires contenant des expressions violentes. Le label 'Non' est donné aux commentaires ordinaires.

et le label '***Non***' dans le cas contraire. Des exemples d'annotation sont illustrés sur la Figure 4.7. L'annotation des 23k commentaires a été faite sur trois reprises, en d'autres termes, chaque commentaire a été annoté trois fois par trois annotateurs différents. Finalement, nous avons sélectionné le label approprié pour chaque commentaire suivant un vote majoritaire.

La prochaine étape sera le nettoyage et la normalisation de ce dataset écrit en Arabe Marocain afin de le préparer au traitement par des algorithmes machine-learning.

4.7.3 Résultats

Une fois les trois annotations ont été achevées, nous avons effectué un vote majoritaire pour avoir un seul label (violent ou non violent) pour chaque commentaire. En résultat, nous avons eu un dataset pour l'identification de la cyberviolence dans les textes en Arabe Marocain composé de 23k commentaires. Dans ce dataset, les deux classe *violent* et *non violent* sont parfaitement équilibré avec 11500 commentaires chacune. En d'autres termes chaque classe représente 50% du dataset.

Quant à l'évaluation, pour mesurer l'accord inter-annotateurs nous avons adopté comme métrique : le kappa de Cohen (Cohen, 1960) qui a donné 0.56 et Krippendorff-alpha (Krippendorff, 2011) qui a donné 0.55 (Tableau 4.13). Ces valeurs indiquent une assez bonne cohérence entre annotateurs qui peut être expliquée par la diversité des âges, des genres et des régions. Nous avons également calculé la distribution de l'accord présentée dans le Tableau 4.14. Comme c'est montré sur ce tableau, l'accord total (où les trois annotateurs ont donné le même label) est d'environ 66%, ce qui reflète un accord significatif entre les annotateurs.

Tableau 4.13 Métriques d'évaluation du corpus

Métriques	Scores
Cohen kappa (moyenne)	0.561
Krippendorff alpha	0.557

Tableau 4.14 Distribution des accords entre annotateurs

Type d'accord	Distribution
Accord total	15372 = 66.83%
Accord partial avec vote majoritaire	7628 = 33.16%

4.8 Conclusion

Dans le but de participer au bien-être des individus, nous avons mené cette étude afin de trouver un moyen de détecter automatiquement les comportements violents à partir du texte généré par les utilisateurs en ligne. Ce qui peut conduire à la détection des cyber-auteurs.

Les psychologues affirment que l'acte de cyberviolence est lié aux caractéristiques de la personnalité de son auteur. Au cours de cette étude, nous avons essayé de démontrer la véracité et la validité de cette affirmation d'une manière computationnelle. Pour ce faire, nous avons extrait des caractéristiques liées à l'état émotionnel et les traits Big Five de la personnalité des utilisateurs, et nous avons ainsi construit des modèles de prédiction d'une façon supervisée sur un dataset de cyberharcèlement. Nous avons utilisé les algorithmes

Ensemble Machine Learning qui ont prouvé leur bonne performance dans le traitement des datasets déséquilibrés. Les résultats obtenus montrent que la personnalité d'un individu est un indicateur important de sa capacité à adopter un comportement violent pendant ces communications virtuelles. Ce qui en découle qu'il faut encore forger dans les caractéristiques psychologiques de ces utilisateurs et de les intégrer dans l'apprentissage des modèles pour améliorer leur pouvoir prédictif.

Nous avons appliqué notre solution au cas du cyberharcèlement, pourtant, ce n'est qu'un cas d'utilisation parmi d'autres. L'application de cette technique peut être généralisée pour la détection du contenu violent dans d'autres types de cyberviolence, comme par exemple le troll ou bien le discours de haine.

Finalement, pour juger qu'un auteur présente un comportement violent, nous ne pouvons pas baser notre jugement sur un seul message violent. Cependant, il faut procurer une idée globale sur son attitude dans ses différentes communications à travers un suivi de son activité en ligne. Si ses messages publiés sont majoritairement violents, alors, à partir de ce moment, nous pouvons confirmer sa tendance à la violence. Dans ce cas, il faut assister et accompagner ce type de personnes afin de les aider à changer leur comportement et ainsi atténuer ou mettre fin à leur mal envers les autres. Cette opération peut être utilisée comme un acte préventif pour protéger les adolescents, par exemple (la catégorie la plus touchée par la violence), à travers des plateformes ou des applications conçues par les établissements responsables de l'éducation, et spécialement dédiées à cet objectif.

Chapitre 5

Conclusion

Dans ce dernier chapitre, nous récapitulerons d'abord les méthodes proposées, puis, nous résumerons nos résultats et finalement nous fournirons des perspectives pour le futur.

Actuellement, la communication virtuelle fait naturellement partie de la vie de nombreuses personnes. Cela est grâce aux applications d'internet et de téléphones intelligents qui ont facilité à tout le monde l'accès aux réseaux sociaux, ceux-ci offrent à leurs utilisateurs un contenu attractif avec beaucoup de liberté. Cette situation a incité à la prolifération de la cyberviolence malgré les efforts déployés pour l'atténuer.

Ce constat a amené de nombreux chercheurs à s'intéresser à ce phénomène afin de réduire sa propagation surtout chez les adolescents. Plusieurs études psychologiques et comportementales ont été menées pour détecter le cercle de la cyberviolence. En particulier les victimes, afin de les aider à dépasser cette mauvaise expérience.

Ayant un très grand intérêt à ce phénomène, nous avons mené cette recherche pour contribuer à la résolution de ce problème et joindre nos efforts aux autres chercheurs pour le bien des êtres-humains.

5.1 Synopsis

L'objectif principal de cette thèse est la détection du contenu violent dans les messages publiés en SM. Pour atteindre cet objectif, nous avons passé par plusieurs sous-objectifs que nous avons décrits en détails pendant les chapitres de cette recherche.

Dans le chapitre 2, nous avons passé en revue les approches relatives à nos principales contributions, qui sont la normalisation des textes SM, considérée comme notre premier objectif avant de faire son analyse. Quant à l'analyse, celle-ci s'intéresse aux techniques de détection des contenus violents basés sur l'apprentissage supervisé.

Dans le chapitre 3, nous avons présenté le processus de la normalisation des textes SM qui sont de nature bruitée et non formelle, et dont les outils existants sont incapables de traiter. Ce processus passe à travers un ensemble de prétraitements. Ces prétraitements incluent la normalisation du code switching, l'identification de langues standards et aussi du dialecte, la correction orthographique, le remplacement du lexique spécial au SM et la normalisation du dialecte. Nous avons évalué nos solutions sur des textes SM contenants du code switching entre deux langues standards qui sont le Français et l'Anglais, en plus de l'Arabe Marocain. Ces textes ont été extraits de trois plateformes SM : Facebook, YouTube et Twitter. Nous avons eu de bons résultats pour les langues standards par opposition au

dialecte qui nécessite encore le développement de ressources et outils linguistiques plus avancés et plus spécifiques.

Le chapitre 4 a abordé le problème de la détection du contenu violent dans les textes SM. Les solutions adoptées reposent sur les techniques Ensemble ML classiques avec des caractéristiques d'apprentissage liées à la personnalité des individus, en l'occurrence, les émotions et les Big Five traits de personnalité. Nous avons entraîné les classifieurs sur un dataset d'harcèlement en ligne, puis, nous les avons comparés aux techniques deep learning. Les modèles générés ont montré leur capacité à identifier les messages violents parmi d'autres.

Dans ce même chapitre, nous avons présenté notre dataset dédié à la détection du contenu violent dans les textes écrits en Arabe Marocain que nous avons élaborés durant cette thèse.

5.2 Résumé des contributions et des résultats

Afin d'analyser les textes générés par les utilisateurs des SM, tels que l'analyse de comportement ou les traits de personnalité, il faut faire face à la nature bruitée de ce type de textes. Ce bruit est produit par l'utilisation de lexique spécial, de mots mal orthographiés et de mélange de langues différentes, ce qui est connu par l'alternance codique ou le code switching.

Utilisation des outils et ressources existants : Les techniques de NLP existantes ne sont pas en mesure de traiter ce type de texte en raison de sa complexité, et la plupart d'entre elles sont conçues pour des langues et des usages standards. Par conséquent, il faut bâtir des solutions spécifiques à ce type de contenus. Afin d'éviter de réinventer la roue, nous avons pensé à la possibilité d'exploiter les outils et les ressources existants dans nos approches.

Approches indépendantes de la langue : Pour pallier les limites des solutions basées sur les règles et celles basées sur les corpus, nous avons conçu une solution qui s'appuie sur une approche basée sur les connaissances tout en profitant des outils existants. Cette stratégie nous a permis de garder l'aspect indépendant de la solution sans recourir aux règles linguistiques et sans développer des corpus annotés spécifiques. Ce qui a été l'un de nos objectifs de départ dans cette thèse et auquel nous avons essayé de répondre.

Normalisation du code switching : La contribution principale dans cette première phase est la normalisation du code switching, l'un des défis majeurs du NLP surtout lorsqu'il s'agit de mélange entre langues standards et dialectes. Ce traitement se caractérise par son aspect sémantique qui tient à préserver la sémantique sur deux niveaux. D'abord, au niveau phrase, à travers le choix de la langue dominante en tant que langue cible de la traduction des différents mots composant la phrase CS. Ensuite au niveau mot, où les mots cibles sont convertis en respectant leur sémantique, par l'emploi de la technique WSD qui vise à identifier le sens d'un mot dans son contexte d'utilisation.

La WSD avec un contexte vertical multilingue (MVC) : Concernant la WSD, nous avons employé BabelNet, un dictionnaire sémantique multilingue. Dans notre approche, l'identification du sens, et donc de la traduction appropriée d'un mot cible, repose sur le calcul du score de chevauchement entre les définitions, les exemples et les relations sémantiques des synonymes (fournis par BabelNet) d'un mot cible, celui à traduire, et ceux des mots de son contexte. Au lieu du contexte local, nous avons construit le MVC qui inclut les mots entourant le mot cible dans toutes les occurrences de ce mot, tout au long du texte. En plus, ce contexte peut inclure des mots de langues différentes. La traduction ayant le score maximal sera sélectionnée. Nous avons évalué notre approche sur trois corpus, les résultats réalisés pour les langues standards ont dépassé les références de bases considérées. Quant au dialecte, où la traduction a été basée sur un dictionnaire bilingue des sens les plus fréquents, les résultats ont été bien modestes ce qui implique qu'il faut encore travailler sur le traitement du dialecte.

Normalisation du dialecte : Une autre contribution importante dans cette phase est la normalisation de l'Arabe Marocain. Nous avons employé des modèles de word embedding, qui est une technique non supervisée, pour associer plusieurs translittérations d'un mot dialectal à une seule forme canonique, en moyennant la propriété des synonymes les plus similaires d'un mot, offerte par ces modèles. Nous avons montré que cette technique est efficace pour atténuer le degré de diversité de forme d'écriture. En outre cette normalisation peut être étendue avec un corpus plus volumineux, ce qui permettra de couvrir encore plus de translittérations.

Normalisation du lexique spécial au SM : Nous avons collecté et construit des corpus pour couvrir les différentes langues et types de lexiques utilisés en SM, en particulier, celui

des abréviations et des symboles.

Après la phase du prétraitement du texte bruité, vient la phase de détection du contenu violent à l'aide des méthodes ML. Dans cette phase, nous avons appliqué différentes techniques ML classique avec des caractéristiques bien définies fondées sur des recherches faites en psychologie, que nous avons comparées à des architectures deep learning différentes.

Détection du contenu violent dans les textes en explorant la relation entre le langage et la personnalité : Nous avons remarqué que les caractéristiques psychologiques des auteurs, y compris les émotions et les traits de personnalité, n'ont pas été largement abordées par ces travaux, alors qu'elles constituent un critère essentiel dans les études psychologiques sur la cyberviolence. Ce qui nous a motivé à explorer cet aspect manquant dans les études précédentes en moyennent les écrits des utilisateurs en ligne, surtout que plusieurs études, en behaviorisme et la linguistique cognitive, ont fourni des preuves liant le langage à la personnalité. Alors, nous avons conduit cette étude pour répondre à la question : "Y a-t-il une association entre les écrits et la personnalité d'une personne d'une part et sa tendance à montrer un comportement violent d'autre part ? ".

Caractéristiques psychologiques pour la détection du contenu violent : afin de répondre à notre question de recherche, nous avons extrait d'abord des caractéristiques liées à la personnalité des utilisateurs en ligne, notamment, les émotions et les Big Five traits de personnalité. Puis, nous avons construit, sur la base de ces caractéristiques, des modèles dédiés à la prédiction du harcèlement en ligne. Les résultats obtenus démontrent l'association entre les écrits et la personnalité de l'auteur et son comportement nuisible. Ce résultat confirme ou répond à notre hypothèse, et affirme en effet que les émotions et les traits Big Five d'un individu sont un indicateur sérieux de sa disposition à présenter un comportement violent. Par conséquent, la construction d'un modèle ML basé sur les caractéristiques de la personnalité permettra d'identifier les auteurs de la cyberviolence.

Ressources linguistiques : dans le cadre de cette thèse, nous avons pu élaborer des ressources linguistiques diverses, notamment pour l'Arabe Marocain, qui serviront des recherches ultérieures. Nous citons : un lexique spécial au SM, des modèles de langage de type word embedding, un dictionnaire de normalisation et un dataset dédié à la détection de la cyberviolence. Concernant l'Anglais et le Français, nous avons collecté des lexiques dont

nous nous sommes servis durant la phase de la normalisation, en l'occurrence : un dictionnaire de normalisation pour le Français, un lexique de symboles SM, un lexique des expressions idiomatiques et des modèles de langages générés pour chacune de ces deux langues.

5.3 Exploitation potentielle de nos résultats

D'abord, la solution présentée dans cette recherche visant la normalisation des textes bruités peut être exploitée comme un prétraitement des textes des SM dans plusieurs applications NLP.

Encore, notre solution de détection du contenu violent peut être aussi employée comme un prétraitement dont on se sert pour éliminer ce type de contenu présent dans les corpus et les datasets. Ce qui permettra d'éviter le biais dont souffre les datasets actuels et qui génère des messages violents envers des minorités à cause de la couleur, la race, la religion et le genre, dans les assistants vocaux par exemple.

En outre, ces résultats peuvent être exploitées dans le cadre des interventions de e-santé par les associations intéressées à la lutte contre ce phénomène, les établissements d'enseignement, et les plateformes de médias sociaux, comme aide à la décision de l'implication ou non d'un utilisateur dans un acte de cyberviolence. Ce qui permettra de soutenir l'auteur de cet acte à l'aide d'une assistance psychologique en ligne, à travers un chat réel ou un chatbot, afin de l'aider à changer son comportement et ainsi à assurer une expérience d'utilisateur en ligne en toute sécurité.

De plus, cette initiative et d'autres similaires peuvent bien participer à la lutte contre le discours de haine, l'un des préoccupations majeures des états et aussi à l'échelle international, surtout celui contre certaines races et ethnicités (United Nations, 2019).

Finalement, les résultats obtenus montrent qu'aujourd'hui, l'intelligence artificielle peut jouer un rôle majeur dans l'amélioration de la santé mentale des individus et du bien-être humain en général. Ce n'est que le début et l'avenir reste très prometteur.

5.4 Perspectives

Cette thèse n'est qu'un premier pas de recherche dans le domaine NLP, soit au niveau

du prétraitement du texte SM bruité, soit au niveau de la détection du contenu violent en SM. Plusieurs améliorations sont demandées et beaucoup de problèmes sont encore non résolus. Voici les principales perspectives de ce travail :

- Améliorer l'identification du dialecte et les langues standards en cas de code switching. Cela peut être réalisé en générant des modèles de langage plus robustes, surtout pour l'Arabe Marocain, à partir de corpus extraits du web.

- La technique de normalisation de l'Arabe Marocain a réalisé de bonnes performances, cependant elle doit être améliorée avec des modèles d'embedding plus large, afin d'étendre la couverture des différentes translitérations. Cette tâche devient maintenant une nécessité, vu la forme non standard de ces textes d'une part, et d'autre part, le manque de ressources pour l'uniformiser. Cette opération permettra de rendre ce dialecte exploitable par les techniques d'apprentissage supervisées.

- Dans cette étude, nous avons couvert la normalisation de l'Arabe Marocain écrit en script Latin. Dans les travaux à venir, nous allons étendre ce traitement pour inclure aussi le dialecte écrit en script Arabe dont l'utilisation est devenue très répandue dans les SM. Cela sera fait à travers des corpus composés de commentaires collectés du YouTube et d'un dictionnaire de forme canonique en script Arabe.

- En raison des limites matérielles, nous avons appliqué nos techniques de détection du contenu violent sur un dataset de petite taille. Dans un travail à venir nous allons tester la validité de notre approche sur un grand ensemble de données qui sera aussi comparé aux techniques DL.

- Parmi nos objectifs futurs, il y a la détection du contenu violent dans les textes écrits en Arabe Marocain. Dans cette recherche, nous avons couvert la langue anglaise du fait du manque de dataset annotés pour l'Arabe Marocain. Cependant, grâce au dataset que nous avons élaboré, nous pouvons appliquer nos approches

- Finalement, le traitement de l'Arabe Marocain sera notre première préoccupation, puisque sa forme constitue un obstacle devant son analyse que nous attaquerons ultérieurement.

Publications

- Zarnoufi R., Jaafar H., Abik M. (2019) Language Identification for User Generated Content in Social Media. In : Rocha Á., Serrhini M. (eds) Information Systems and Technologies to Support Learning. **EMENA-ISTL 2018**. Smart Innovation, Systems and Technologies, vol 111. Springer, Cham. https://doi.org/10.1007/978-3-030-03577-8_73

- Zarnoufi, R., Boutbi, M. and Abik, M. (2018) 'Cyber-violence: Harmful Behavior Detection in Social Media'. **MISC'2018**. International Conference on Modern Intelligent Systems Concepts. Mohammed V University - Faculty of Sciences, Rabat, Morocco, December 12 - 13, 2018. https://www.researchgate.net/publication/329774997_Cyberviolence_Harmful_Behavior_Detection_in_Social_Media

- Zarnoufi R., Abik M. (2020) Big Five Personality Traits and Ensemble Machine Learning to Detect Cyber-Violence in Social Media. In: Serrhini M., Silva C., Aljahdali S. (eds) Innovation in Information Systems and Technologies to Support Learning Research. **EMENA-ISTL 2019**. Learning and Analytics in Intelligent Systems, vol 7. Springer, Cham. https://doi.org/10.1007/978-3-030-36778-7_21

- Randa Zarnoufi, Hamid Jaafar, and Mounia Abik. 2020. Machine Normalization: Bringing Social Media Text from Non-Standard to Standard Form. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 19, 4, Article 49 (April 2020), 30 pages. *https://doi.org/10.1145/3378414* (**IF 1.42**)

- Randa Zarnoufi, Walid Bachri, Hamid Jaafar, Mounia Abik, "MANorm: A Normalization Dictionary for Moroccan Arabic Dialect Written in Latin Script", **COLING (Class A)/WANLP**, Barcelona, Spain (Online December 2020). https://www.aclweb.org/anthology/2020.wanlp-1.14/

- Zarnoufi, R., Boutbi, M. and Abik, M. (2020) 'AI to prevent cyber-violence: harmful behaviour detection in social media', **Int. J. High Performance Systems**

Architecture, Vol. 9, No. 4, pp.182–191.

- Hamid Jaafar, Randa Zarnoufi (2021). Pour une Numérisation du Patrimoine Linguistique et Sociolinguistique de l'Arabe Marocain. **INALCO**. ***Accepté***.
- Classical Machine Learning vs Deep Learning to Detect Cyberviolence in Social Media. **SIMBIG'21**. ***En cours de publication*** .
- Violent Content Dataset for Moroccan Arabic Dialect. **MoroccanAI Conference 2021**. ***Accepté***.

Références

Abidi, K., Smaili, K., 2018. An Automatic Learning of an Algerian Dialect Lexicon by using Multilingual Word Embeddings, in: 11th Edition of the Language Resources and Evaluation Conference, LREC. pp. 832–838.

Agirre, E., Lacalle, D., Soroa, A., 2014. Random Walks for Knowledge-Based Word Sense Disambiguation. Comput. Linguist. 40, 57–84. doi:10.1162/COLI

Agirre, E., Martinez, D., 2004. Unsupervised WSD based on automatically retrieved examples : The importance of bias, in: Proceeding of EMNLP 2004.

Agirre, E., Soroa, A., 2009. Personalizing PageRank for Word Sense Disambiguation, in: Proceedings of the 12th Conference of the European Chapter of the ACL. pp. 33–41. doi:10.3115/1609067.1609070

Agrawal, S., Awekar, A., 2018. Deep Learning for Detecting Cyberbullying Across Multiple Social Media Platforms, in: ECIR 2018. Advances in Information Retrieval. Lecture Notes in Computer Science. pp. 141–153.

Al-badrashiny, M., Eskander, R., Habash, N., Rambow, O., 2014. Automatic Transliteration of Romanized Dialectal Arabic 30–38.

Al-garadi, M.A., Varathan, K.D., Ravana, S.D., 2016. Computers in Human Behavior Cybercrime detection in online communications : The experimental case of cyberbullying detection in the Twitter network. Comput. Human Behav. 63, 433–443. doi:10.1016/j.chb.2016.05.051

Almeida, T.A., Silva, T.P., Santos, I., Gómez Hidalgo, J.M., 2016. Text normalization and semantic indexing to enhance Instant Messaging and SMS spam filtering. Knowledge-Based Syst. 108, 25–32. doi:10.1016/j.knosys.2016.05.001

Antonova, A., Misyurev, A., 2014. Improving the precision of automatically constructed human-oriented translation dictionaries, in: Proceedings Ofthe 3rd Workshop on Hybrid Approaches to Translation (HyTra) @ EACL. pp. 58–66.

Apidianaki, M., Wisniewski, G., Sokolov, A., Max, A., Yvon, F., 2012. WSD for N-best Reranking and Local Language Modeling in SMT. Proc. Sixth Work. Syntax. Semant. Struct. Stat. Transl. 1–9.

Azucar, D., Marengo, D., Settanni, M., 2018. Predicting the Big 5 personality traits from digital footprints on social media: A meta-analysis. Pers. Individ. Dif. 124, 150–159. doi:10.1016/j.paid.2017.12.018

Back, M.D., Stopfer, J.M., Vazire, S., Gaddis, S., Schmukle, S.C., Egloff, B., Gosling, S.D., 2010. Facebook profiles reflect actual personality, not self-idealization. Psychol. Sci. 21, 372–374.

Badjatiya, P., Gupta, S., Gupta, M., Varma, V., 2017. Deep Learning for Hate Speech Detection in Tweets, in: Proceedings of the 26th International Conference on World Wide Web Companion. pp. 759–760.

Balakrishnan, V., Khan, S., Fernandez, T., Arabnia, H.R., 2019. Cyberbullying detection on twitter using Big Five and Dark Triad features. Pers. Individ. Dif. 141, 252–257. doi:10.1016/j.paid.2019.01.024

Baldwin, T., 2017. Language identification in the Wild, The First Workshop on Multi-Language Processing in a Globalising World, Dublin, Ireland.

Banerjee, S., Pedersen, T., 2003. Extended Gloss Overlaps as a Measure of Semantic Relatedness, in: 18th International Joint Conference on Artificial Intelligence (IJCAI). Acapulco, Mexico, pp. 805–810.

Banerjee, S., Pedersen, T., 2002. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet, in: Fourth International Conference on Computational Linguistics and Intelligent Text Processing (CICLING-02). Mexico City.

Barba, E., Pasini, T., Navigli, R., 2021. ESC: Redesigning WSD with Extractive Sense Comprehension, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 4661–4672. doi:10.18653/v1/2021.naacl-main.371

Basile, P., Caputo, A., Semeraro, G., 2014. An Enhanced Lesk Word Sense Disambiguation Algorithm through a Distributional Semantic Model, in: Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers (COLING 14). pp. 1591–1600.

Berguer, A., 2015. LES COLLEGIENS ET LES LYCEENS SONTILS EGAUX FACE AU RISQUE D'ETRE VICTIMES ET/OU AUTEURS DE CYBERVIOLENCE ET DE CYBERHARCELEMENT ? Int. J. Violence Sch. 15, 88–118.

Bertaglia, T.F.C., Nunes, M. das G.V., 2016. Exploring Word Embeddings for Unsupervised Textual User-Generated Content Normalization, in: Proceedings of the 2nd Workshop on Noisy User-Generated Text. pp. 112–120.

Bevilacqua, M., Pasini, T., Raganato, A., Navigli, R., 2021. Recent Trends in Word Sense Disambiguation : A Survey, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21) Survey Track. pp. 4330–4338.

Bhat, I.A., Bhat, R.A., Shrivastava, M., Sharma, M.D., 2018. Universal Dependency Parsing for Hindi-English Code-switching, in: Proceedings OfNAACL-HLT 2018. pp. 987–998.

Bisazza, A., Federico, M., 2016. A Survey of Word Reordering in Statistical Machine Translation : Computational Models and Language Phenomena. Comput. Linguist. 42, 163–205. doi:10.1162/COLI

Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent Dirichlet Allocation. J. Mach. Learn. Reseach 3, 993–1022.

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., 2017. Enriching Word Vectors with Subword Information. Trans. Assoc. Comput. Linguist. 5, 135–146.

Boukous, A., 1995. Société, langues et cultures au Maroc. Enjeux symboliques. Rabat, Publications de la Faculté des Lettres et des Sciences Humaines.

Boumans, L.P., 1998. The Syntax of Codeswitching Analysing Moroccan Arabic / Dutch Conversations. The Netherlands: Tilburg University Press.

Breiman, L., 1996. Bagging predictors. Mach. Learn. 24, 123–140. doi:10.3390/risks8030083

Bretschneider, U., Wöhner, T., Peters, R., 2014. Detecting Online Harassment in Social Networks, in: Thirty Fifth International Conference on Information Systems. pp. 1–14.

Brown, P.F., Pietra, S.A. Della, Pietra, V.J. Della, Mercer, R.L., 1993. The mathematics of statistical machine translation: Parameter estimation. Comput. Linguist. 19, 263–311.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D., 2020. Language models are few-shot learners, in: Advances in Neural Information Processing Systems. pp. 1877--1901.

Cacheda, F., Fernandez, D., Novoa, F., Carneiro, V., 2019. Early Detection of Depression: Social Network Analysis and Random Forest Techniques. J Med Internet Res 21.

Carpuat, M., Wu, D., 2007. Improving statistical machine translation using word sense disambiguation. Proc. 2007 Jt. Conf. Empir. Methods Nat. Lang. Process. Comput. Nat. Lang. Learn. 61–72. doi:10.3115/1219840.1219888

Caubet, D., 2017. Vers une littératie numérique pour la darija au Maroc , une démarche collective, in: Studies on Arabic Dialectology and Sociolinguistics. Proceedings of the 12th International Conference of AIDA.

Çetinoğlu, Ö., Schulz, S., Vu, N.T., 2016. Challenges of Computational Processing of Code-Switching. doi:10.18653/v1/W16-5801

Chan, Y., Ng, H., Chiang, D., 2007. Word sense disambiguation improves statistical machine translation. Annu. Meet. … 45, 33.

Chatzakou, D., Kourtellis, N., Blackburn, J., Cristofaro, E. De, Stringhini, G., Vakali, A., 2017. Mean Birds : Detecting Aggression and Bullying on Twi er, in: Proceedings of the 2017 ACM on Web Science Conference New York: USA. pp. 13–22.

Chawla, N. V, Bowyer, K.W., Hall, L.O., 2002. SMOTE : Synthetic Minority Over-sampling Technique 16, 321–357.

Chen, X., Liu, Z., Sun, M., 2014. A unified model for word sense representation and disambiguation, in: EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. pp. 1025–1035. doi:10.3115/v1/d14-1110

Cheung, P., Fung, P., 2005. Translation disambiguation in mixed language queries. Mach. Transl. 18, 251–273. doi:10.1007/s10590-004-7692-5

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation, in: EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. pp. 1724–1734. doi:10.3115/v1/d14-1179

Chomsky, N., 1970. Remarks on nominalization. Read. English Transform. Gramm. 184–221.

Cohen, J., 1960. A coefficient of agreement for nominal scales. Educ. Psychol. Meas. 20, 37-46

ST-A coefficient of agreement for nominal.

Conicet, U., Rodriguez, J.M., Conicet, U., Godoy, D., Conicet, U., 2018. Textual Aggression Detection through Deep Learning 177–187.

Cook, P., Stevenson, S., 2009. An Unsupervised Model for Text Message Normalization, in: Proceedings Ofthe NAACL HLTWorkshop on Computational Approaches to Linguistic Creativity. pp. 71–78.

Corbí-bellot, A.M., Forcada, M.L., Ortiz-rojas, S., Pérez-, J.A., Ramírez-sánchez, G., Sánchez-martínez, F., Alegria, I., Sarasola, K., Taldea, I.X.A., Fakultatea, I., Unibertsitatea, E.H., Donostia, E.-, 2005. An Open-Source Shallow-Transfer Machine Translation Engine for the Romance Languages of Spain, in: EAMT Conference Proceedings. pp. 79–86.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20, 273–297. doi:10.1007/BF00994018

Costa-Jussà, M.R., Centelles, J., 2015. Description of the Chinese-to-Spanish Rule-Based Machine Translation System Developed Using a Hybrid Combination of Human Annotation and Statistical Techniques. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 15, 1–13.

Costa-Jussà, M.R., Fonollosa, J.A.R., 2015. Latest trends in hybrid machine translation and its applications. Comput. Speech Lang. 32, 3–10. doi:10.1016/j.csl.2014.11.001

Cotelo, J.M., Cruz, F.L., Troyano, J.A., Ortega, F.J., 2015. Expert Systems with Applications A modular approach for lexical normalization applied to Spanish tweets. Expert Syst. Appl. 42, 4743–4754. doi:10.1016/j.eswa.2015.02.003

Crego, J.M., Johanson, J., Senellart, J., 2014. SYSTRAN RBMT Engine : hybridization experiments !

Dadvar, M., de Jong, F., Ordelman, R., Trieschnigg, D., 2012a. Improved Cyberbullying Detection Using Gender Information, in: 12th -Dutch-Belgian Information Retrieval Workshop. DIR'2012. pp. 22–25.

Dadvar, M., Eckert, K., 2018. Cyberbullying Detection in Social Networks Using Deep Learning Based Models ; A Reproducibility Study, in: DaWaK. pp. 1–13.

Dadvar, M., Ordelman, R., Jong, F. De, Trieschnigg, D., 2012b. Towards User Modelling in the Combat against Cyberbullying, in: Natural Language Processing and Information Systems. pp. 277–283.

Dadvar, M., Trieschnigg, D., de Jong, F., 2014. Experts and Machines against Bullies: A Hybrid Approach to Detect Cyberbullies, in: Advances in Artificial Intelligence. AI 2014. Lecture Notes in Computer Science. pp. 275–281.

Dadvar, M., Trieschnigg, D., Jong, F. De, 2011. Experts and Machines Against Bullies : A Hybrid Approach to Detect Cyberbullies.

Das, A., Gambäck, B., 2014. Identifying Languages at the Word Level in Code-Mixed Indian Social Media Text, in: Proc. of the 11th Intl. Conference on Natural Language Processing. pp. 378–387.

Davahli, M.R., Karwowski, W., GutierrezE., Fiok, K., Wrobel, G., Taiar, R., Ahram, T., 2020. Personality and Text: Quantitative Psycholinguistic Analysis of a Stylistically

Differentiated Czech Text. Psychol. Stud. (Mysore). 12, 1–23.

David-Ferdon, C., Hertz, M.F., 2007. Electronic media, violence, and adolescents: An emerging public health problem. J. Adolesc. Heal. 41, S1–S5.

Davidson, T., Warmsley, D., Macy, M., Weber, I., 2017. Automated Hate Speech Detection and the Problem of Offensive Language ∗, in: ICWSM 2017.

Devlin, J., Chang, M., Lee, K., Toutanova, K., 2019. BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of NAACL-HLT 2019. pp. 4171–4186.

Dhar, M., 2018. Enabling Code-Mixed Translation : Parallel Corpus Creation and MT Augmentation Approach, in: Proceedings Ofthe First Workshop on Linguistic Resources for Natural Language Processing. pp. 131–140.

Dinakar, K., Reichart, R., Lieberman, H., 2011. Modeling the Detection of Textual Cyberbullying. Int. AAAI Conf. Web Soc. Media, North Am. 11–17.

Dongsuk, O., Kwon, S., Kim, K., Ko, Y., 2018. Word Sense Disambiguation Based on Word Similarity Calculation Using Word Vector Representation from a Knowledge-based Graph, in: Proceedings Ofthe 27th International Conference on Computational Linguistics. pp. 2704–2714.

Dostert, L.E., 1959. Approaches to the Reduction of Ambiguity in Machine Translation. J. SMPTE 68, 234–235.

Ekman, P., 1999. Basic Emotions, in: Dalgleish, T., Power, M.J. (Eds.), Handbook of Cognition and Emotion. pp. 45–60.

Ekman, P., 1972. Universals and Cultural Differences in Facial Expressions of Emotion. Nebraska Symp. Motiv. 19, 207–282.

Elfardy, H., Diab, M., 2012. Token Level Identification of Linguistic Code Switching, in: Proceedings of COLING 2012: Posters. pp. 287–296.

Eryigit, G., Torunoglu-Selamet, D., 2017. Social media text normalization for Turkish. Nat. Lang. Eng. 1–41. doi:10.1017/S1351324917000134

Farzindar, A., Inkpen, D., 2018. Natural Language Processing for Social Media Second Edition, 2nd ed.

Feng, W., Huang, W., Ren, J., 2018. Class imbalance ensemble learning based on the margin theory. Appl. Sci. 8. doi:10.3390/app8050815

Fersini, E., Messina, E., Pozzi, F.A., 2016. Expressive signals in social media languages to improve polarity detection. Inf. Process. Manag. 52, 20–35. doi:10.1016/j.ipm.2015.04.004

Florian, R., Cucerzan, S., Schafer, C., Yarowsky, D., 2002. Combining Classifiers for word sense disambiguation. Nat. Lang. Eng. 8, 327–341. doi:10.1017/S1351324902002978

Forcada, M.L., Sánchez-martínez, F., Ramirez-Sánchez, G., Tyers, F.M., 2011. Apertium : A free / open-source platform for rule-based machine translation. Mach. Transl. 25, 127–144. doi:10.1007/s10590-011-9090-0

Founta, A., Chatzakou, D., Kourtellis, N., Blackburn, J., Vakali, A., Leontiadis, I., 2017. A

Unified Deep Learning Architecture for Abuse Detection.

Fung, P., Xiaohu, L., Shun, C.C., 1999. Mixed Language Query Disambiguation 333–340. doi:10.3115/1034678.1034732

Gale, W.A., Church, K.W., Yarowsky, D., Laboratories, T.B., Nj, M.H., 1992. One Sense Per Discourse, in: Proceedings of the DARPA. Speech and Natural Language Workshop (Harriman, NY). pp. 233–237.

Gambäck, B., Sikdar, U.K., 2017. Using Convolutional Neural Networks to Classify Hate-Speech 85–90.

Gao, J., Nie, J.Y., He, H., Chen, W., Zhou, M., 2002. Resolving query translation ambiguity using a decaying co-occurrence model and syntactic dependence relations, in: SIGIR Forum (ACM Special Interest Group on Information Retrieval). pp. 183–190. doi:10.1145/564405.564409

Goldberg, L.R., 1980. Language and individual differences : The search for universals in personality lexicons. Rev. Personal. Soc. Psychol. 2, 141–165.

Goldhahn, D., Eckart, T., Quasthoff, U., 2012. Building Large Monolingual Dictionaries at the Leipzig Corpora Collection : From 100 to 200 Languages, in: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012). pp. 759–765.

Gumpel, T., Sutherland, K.S., 2010. The relation between emotional and behavioral disorders and school-based violence. Aggress. Violent. Behav. 15, 349–356.

Guthrie, D., Allison, B., Liu, W., Guthrie, L., Wilks, Y., 2006. A Closer Look at Skip-gram Modelling, in: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06). pp. 1222–1225.

Hall, J.L., 2015. DEBATING DARIJA : LANGUAGE IDEOLOGY AND THE WRITTEN REPRESENTATION OF MOROCCAN ARABIC IN MOROCCO.

Hall, M., Caton, S., 2017. Am I who I say I am? Unobtrusive self-representation and personality recognition on Facebook. PLoS One 12. doi:10.1371/journal.pone.0184417

Hamers, J.F., Blanc, M., 1983. Bilingualité et bilinguisme, in: Bruxelles : P. Mardaga, D. 1983 (Ed.), Collection Psychologie et Sciences Humaines.

Han, B., Baldwin, T., 2011. Lexical Normalisation of Short Text Messages : Makn Sens a # twitter, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics. pp. 368–378.

Han, B., Cook, P., Baldwin, T., 2012. Automatically Constructing a Normalisation Dictionary for Microblogs, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Pages 421–432, Jeju Island, Korea, 12–14 July 2012. pp. 421–432.

Haugen, E., 1950. The Analysis of Linguistic Borrowing. Language (Baltim). 26, 210–231.

Heafield, K., Pouzyrevsky, I., Clark, J.H., 2013. Scalable Modified Kneser-Ney Language Model Estimation, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. pp. 690–696.

Hládek, D., Staš, J., Pleva, M., 2020. Survey of automatic spelling correction. Electron. 9, 1–29. doi:10.3390/electronics9101670

Hochreiter, S., Schmidhuber, J., 1997. Long Short-Term Memory. Neural Comput. 9, 1735–1780.

Htait, A., Fournier, S., Bellot, P., 2018. Unsupervised Creation of Normalisation Dictionaries for Micro-Blogs in Arabic , French and English, in: International Conference on Computational Linguistics and Intelligent Text Processing.

Huang, Q., Singh, V.K., Atrey, P.K., 2014. Cyber Bullying Detection Using Social and Textual Analysis, in: International Workshop on Socially-Aware Multimedia. pp. 3–6.

Hutchins, W.J., 1986. Machine Translation: past, present, future. Ellis Horwood, Chichester, UK. (Halstead Press, New York) Hutchins.

Ide, N., Véronis, J., 1998. Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art. Comput. Linguist. 24, 1–40. doi:10.1016/j.csl.2004.05.005

Jaafar, H., 2012. PhD Thesis: Le Nom et l'Adjectif dans l'Arabe Marocain: Etude Lexicologique. University Sidi Mohammed Ben Abbdellah.

Jauhiainen, T., Lui, M., Zampieri, M., Baldwin, T., Lindén, K., 2019. Automatic Language Identification in Texts: A Survey. J. Artif. Intell. Res. 65, 675–782.

Jiang, J.J., Conrath, D.W., 1997. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. Proc. Int. Conf. Res. Comput. Linguist. 19–33. doi:10.1.1.269.3598

Johnson, J.A., 2017. Big-Five Model, in: V. Zeigler-Hill, T.K.S. (Eds. . (Ed.), Encyclopedia of Personality and Individual Differences. New York: Springer, p. (1-16). doi:DOI: 10.1007/978-3-319-28099-8_1212-1

Johnson, M., Schuster, M., Le, Q. V, Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., Hughes, M., Dean, J., 2017. Google ' s Multilingual Neural Machine Translation System : Enabling Zero-Shot Translation. Trans. Assoc. Comput. Linguist. 5, 339–351.

Joshi, A.K., 1985. Processing of sentences with intrasentential code switching, in: Dowty, D.R. & Karttunen, L. & Zwicky, A. (eds. . (Ed.), Natural Language Parsing. Cambridge University Press, pp. 190–205.

Jurgens, D., Tsvetkov, Y., Jurafsky, D., 2017. Incorporating Dialectal Variability for Socially Equitable Language Identification. Acl 51–57. doi:10.18653/v1/P17-2009

Kaufmann, M., Kalita, J., 2010. Syntactic Normalization of Twitter Messages, in: International Confer- Ence on Natural Language Processing, Kharagpur, India. pp. 1–7.

Kilgarriff, A., Rosenzweig, J., 2000. English Senseval: Report and Results, in: Second Conf on Language Resources and Evaluation. pp. 1239–1244. doi:10.1023/A:1002693207386

Koehn, P., 2020. Neural Machine Translation, Cambridge: Cambridge University Press. Cambridge University Press. doi:https://doi.org/10.1017/9781108608480

Koehn, P., 2005. Europarl : A Parallel Corpus for Statistical Machine Translation, in: MT Summit. pp. 79–86.

Koehn, P., Birch, A., Callison-burch, C., Federico, M., Bertoldi, N., Cowan, B., Moran, C., Dyer, C., Constantin, A., Herbst, E., 2007. Moses : Open Source Toolkit for Statistical Machine Translation, in: Proceedings Ofthe ACL 2007 Demo and Poster Sessions. pp. 177–180.

Kontostathis, A., Reynolds, K., Garron, A., Edwards, L., 2013. Detecting Cyberbullying : Query Terms and Techniques, in: WebSci '13 Proceedings of the 5th Annual ACM Web Science Conference. pp. 195–204.

Kowalski, R.M., Giumetti, G.W., Schroeder, A.N., Lattanner, M.R., 2014. Bullying in the Digital Age : A Critical Review and Meta-Analysis of Cyberbullying Research Among Youth. Psychol. Bull. © 2014 Am. Psychol. Assoc. 140, 1073–1137. doi:10.1037/a0035618

Krippendorff, K., 2011. Computing Krippendorff ' s Alpha-Reliability. Dep. Pap. 12.

Kumar, R., Ojha, A.K., Malmasi, S., Zampieri, M., 2018. Benchmarking Aggression Identification in Social Media, in: Proceedings Ofthe First Workshop on Trolling, Aggression and Cyberbullying. pp. 1–11.

Le, A.C., Shimazu, A., 2004. High WSD Accuracy Using Naive Bayesian Classifier with Rich Features, in: Proceedings of the 18th Pacific Asia Conference on Language, Information and Computation. pp. 105–114.

Leacock, C., Chodorow, M., 1998. Combining Local Context and WordNet Similarity for Word Sense Identification. WordNet An Electron. Lex. database. 265–283. doi:citeulike-article-id:1259480

Lecun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D., Laboratories, T.B., 1990. Handwritten Digit Recognition with a Back-Propagation Network, in: NIPS. pp. 396–404.

Lesk, M., 1986. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone, in: SIGDOC-86: 5th International Conference on Systems Documentation. pp. 24–26.

Li, S., 2015. Lifetime Achievement Award Translating Today into Tomorrow. Comput. Linguist. 41, 4943. doi:10.1162/COLI

Lim, A., 2020. The big five personality traits. [WWW Document]. Simply Psychol. https//www.simplypsychology.org/big-five-personality.html.

Lin, D., 1998. An Information-Theoretic Definition of Similarity. Proc. ICML 296–304. doi:10.1.1.55.1832

Ling, W., Xiang, G., Dyer, C., Black, A., Trancoso, I., 2013. Microblogs as Parallel Corpora, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria. pp. 176–186.

Lopez Ludeña, V., San Segundo, R., Montero, J.M., Barra Chicote, R., Lorenzo, J., 2012. Architecture for Text Normalization using Statistical Machine Translation techniques, in: IberSPEECH 2012. Madrid, Spain: Springer. pp. 112–122. doi:10.1016/j.jacc.2018.03.023

Loureiro, D., Jorge, A., 2019. LIAAD at SemDeep-5 Challenge: Word-in-Context (WiC), in: The 5th Workshop on Semantic Deep Learning (SemDeep-5). pp. 1–5.

Luo, F., Liu, T., Xia, Q., Chang, B., Sui, Z., 2018. Incorporating glosses into neural word sense

disambiguation, in: ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers). pp. 2473–2482. doi:10.18653/v1/p18-1230

Lusetti, M., Ruzsics, T., Göhring, A., Samardži´c, T.S., Stark, E., 2018. Encoder-Decoder Methods for Text Normalization, in: Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects. pp. 18–28.

Mager, M., Çetinoglu, Ö., Kann, K., 2019. Subword-level language identification for intra-word code-switching, in: NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference. pp. 2005–2011. doi:10.18653/v1/n19-1201

Mahmud, A., Ahmed, K.Z., Khan, M., 2008. Detecting flames and insults in text.

Mairesse, F., Walker, M.A., Mehl, M.R., Moore, R.K., 2007. Using linguistic cues for the automatic recognition of personality in conversation and text. J. Artif. Intell. Res. 30, 457–500. doi:10.1613/jair.2349

Manandise, E., Gdaniec, C., 2011. Morphology to the rescue redux: Resolving borrowings and code-mixing in machine translation. Commun. Comput. Inf. Sci. 100 CCIS, 86–97. doi:10.1007/978-3-642-23138-4_6

Mccarthy, D., Koeling, R., Carroll, J., 2007. Unsupervised Acquisition of Predominant Word Senses. Comput. Linguist. 33, 553–590.

Mccrae, R.R., Costa, P.T., 1987. Validation of the five factor model of personality across instruments and observers Validation of the Five-Factor Model of Personality Across Instruments and Observers. J. Pers. Soc. Psychol. 52, 81–90. doi:10.1037/0022-3514.52.1.81

McNamee, P., 2005. Language identification: a solved problem suitable for undergraduate instruction. J. Comput. Sci. Coll. 20, 94–101.

Mihalcea, R., Tarau, P., Figa, E., 2004. PageRank on Semantic Networks , with Application to Word Sense Disambiguation, in: In Proceedings of the 20th International Conference on Computational Linguistics(COLING, Geneva, Switzerland). pp. 1126–1132.

Mikolov, T., Corrado, G., Chen, K., Dean, J., 2013. Efficient Estimation of Word Representations in Vector Space, in: In Proceedings of Workshop at ICLR.

Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.J., 1990. Introduction to wordnet: An on-line lexical database. Int. J. Lexicogr. 3, 235–244. doi:10.1093/ijl/3.4.235

Miller, G.A., Chodorow, M., Landes, S., Leacock, C., Thomas, R.G., 1993. Using a semantic concordance for sense identification, in: In Proceedings of the Workshop on Human Language Technology, Association for Computational Linguistics. pp. 303–308. doi:10.3115/1075812.1075866

Min, W., Mott, B.W., 2015. NCSU _ SAS _ WOOKHEE : A Deep Contextual Long-Short Term Memory Model for Text Normalization, in: Proceedings of the ACL 2015 Workshop on Noisy User-Generated Text. pp. 111–119.

Mohammad, S.M., Kiritchenko, S., 2014. Using Hashtags to Capture Fine Emotion Categories from Tweets Using Hashtags to Capture Fine Emotion Categories from Tweets. Comput.

Intell. 31, 301–326. doi:10.1111/coin.12024

Mohammad, S.M., Turney, P.D., 2013. Crowdsourcing a Word–Emotion Association Lexicon. Comput. Intell. 29, 436–465.

Moreno, J.D., Martínez-Huertas, J., Olmos, R., Jorge-Botana, G., Botella, J., 2021. Can personality traits be measured analyzing written language? A meta-analytic study on computational methods. Pers. Individ. Dif. 177. doi:10.1016/j.paid.2021.110818

Muller, B., Sagot, B., Seddah, D., 2019. Enhancing BERT for Lexical Normalization, in: Proceedings Ofthe 2019 EMNLP Workshop W-NUT: The 5th Workshop on Noisy User-Generated Text. pp. 297–306.

Munezero, M., Montero, C.S., Kakkonen, T., Sutinen, E., 2014. Automatic Detection of Antisocial Behaviour in Texts. Informatica 38, 3–10.

Muysken, P., 1995. Cross-Disciplinary Perspectives on Code-Switching, in: Lesley Milroy, U. of N. upon T., Pieter Muysken, R.U.N. (Eds.), One Speaker, Two Languages. Cambrige University Press, pp. 177–198.

Myers-Scotton, C., 1995. A lexically based model of code-switching, in: Lesley Milroy, U. of N. upon T., Pieter Muysken, R.U.N. (Eds.), One Speaker, Two Languages: Cross-Disciplinary Perspectives on Code-Switching. Cambridge: Cambridge University Press., pp. 233–256. doi:10.1017/CBO9780511620867.011

Myers-Scotton, C., Jake, J., 2001. Explaining Aspects of Codeswitching and their implications, in: Nicol, J. (Ed.), One Mind, Two Languages: Bilingual Language Processing. Blackwell, Oxford., pp. 84–116.

Naaman, M., Boase, J., Lai, C.H., 2010. Is it really about me?: Message content in social awareness streams., in: Ings of the 2010 ACM Conference on Computer Supported Cooperative Work. pp. 189 –192.

Nahar, V., Al-Maskari, S., Li, X., Pang, C., 2014. Semi-supervised Learning for Cyberbullying Detection in Social Networks, in: Databases Theory and Applications. ADC 2014. Lecture Notes in Computer Science. pp. 160–171.

Nahar, V., Li, X., Pang, C., 2013. An Effective Approach for Cyberbullying Detection. Commun. Inf. Sci-ence Manag. Eng. 3, 238.

Nahar, V., Unankard, S., Li, X., Pang, C., 2012. Sentiment analysis for effective detection of cyber bullying. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 7235, 767–774. doi:10.1007/978-3-642-29253-8_75

Nalinipriya, G., Asswini, M., 2015. A DYNAMIC COGNITIVE SYSTEM FOR AUTOMATIC DETECTION AND PREVENTION OF CYBER-BULLYING ATTACKS. ARPN J. Eng. Appl. Sci. 10, 4618–4626.

Nansel, T., Overpeck, M., Pilla, R., Ruan, W., Simons-Morton, B., Scheidt, P., 2001. Bullying Behaviors Among US Youth: Prevalence and Association With Psychosocial Adjustment. JAMA. 285, 2094–2100.

Navigli, R., 2009. Word Sense Disambiguation: A Survey. ACM Comput. Surv. 41, 69. doi:10.1145/1459352.1459355

Navigli, R., Lapata, M., 2010. An Experimental Study of Graph Connectivity for Unsupervised

Word Sense Disambiguation. IEEE Trans. Pattern Anal. Mach. Intell. 32, 678–692. doi:10.1109/TPAMI.2009.36

Navigli, R., Ponzetto, S.P., 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artif. Intell. 193, 217–250. doi:10.1016/j.artint.2012.07.001

Nguyen, D., Do, A.S., 2013. Word Level Language Identification in Online Multilingual Communication, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Pages 857–862, Seattle, Washington, USA, 18-21 October 2013. pp. 857–862.

Nobata, C., Tetreault, J., 2016. Abusive Language Detection in Online User Content, in: Proceddin of International World Wide Web Conference. pp. 145–153.

Olweus, D., 1978. Aggression in the Schools: Bullies and Whipping Boys, Hemisphere Publishing Corporation. Washington, DC: Hemisphere Publ. Corp.

Owen, T., 2016. Cyber Violence: Towards a Predictive Model Drawing Upon Genetics, Psychology and Neuroscience. Int. J. Criminol. Sociol. Theory 9, 1–11.

Page, L., Sergey, B., Rajeev, M., Terry, W., 1999. The PageRank Citation Ranking: Bringing Order to the Web, in: Technical Report. doi:10.1109/IISWC.2012.6402911

Panchenko, A., Ruppert, E., Faralli, S., Ponzetto, S.P., Biemann, C., 2017. Unsupervised does not mean uninterpretable: The case for word sense induction & disambiguation, in: 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 - Proceedings of Conference. pp. 86–98. doi:10.18653/v1/e17-1009

Park, G., Schwartz, H.A., Eichstaedt, J.C., Kern, M.L., Kosinski, M., Stillwell, D.J., Ungar, L.H., Seligman, M.E.P., 2015. Automatic Personality Assessment Through Social Media Language. J. Pers. Soc. Psychol. 108, 934–952.

Partanen, N., Hamalainen, M., Alnajjar, K., 2019. Dialect Text Normalization to Normative Standard Finnish, in: Proceedings Ofthe 2019 EMNLP Workshop W-NUT: The 5th Workshop on Noisy User-Generated Text. pp. 141–146.

Pasini, T., Navigli, R., 2020. Train-O-Matic : Supervised Word Sense Disambiguation with no (manual) effort. Artif. Intell. 279, 103215. doi:10.1016/j.artint.2019.103215

Paul, S., Smith, P.K., Blumberg, H.H., 2012. Investigating legal aspects of cyberbullying. Psicothema 24, 640–645.

Pedersen, T., 2007. Unsupervised Corpus-Based Methods for WSD, in: Word Sense Disambiguation: Algorithms and Applications,. pp. 133–166. doi:10.1007/978-1-4020-4809-8_6

Pennebaker, J.W., Boyd, R.L., Jordan, K., Blackburn, K., 2015. The Development and Psychometric Properties of LIWC2015 The Development and Psychometric Properties of. doi:10.15781/T29G6Z

Pennington, J., Socher, R., Manning, C.D., 2014. GloVe : Global Vectors for Word Representation, in: EMNLP. pp. 1532–1543.

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations, in: NAACL HLT 2018 - 2018 Conference of

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference. pp. 2227–2237. doi:10.18653/v1/n18-1202

Peterson, J., Densley, J., 2017. Cyber violence: What do we know and where do we go from here? Aggress. Violent Behav. 34, 193–200. doi:10.1016/j.avb.2017.01.012

Plutchik, R., 1980. Emotion: A psychoevolutionary synthesis., New York: ed.

Plutchik, R., Conte, H.R., 1997. Circumplex models of personality and emotions. American Psychological Association. doi:https://doi.org/10.1037/10261-000

Poplack, S., 1980. Sometimes I ' ll start a sentence in Spanish y termino en ESPAÑOL: toward a typology of code-switching. Linguistics 18, 581–618.

Przybyła, P., 2017. How big is big enough? Unsupervised word sense disambiguation using a very large corpus. Arxiv.

Rada, R.O.Y., Mili, H., Bicknell, E., Blettner, M., 1989. Development and Application of a Metric on Semantic Nets. doi:10.1109/21.24528

Rajendran, A., Zhang, C., Abdul-Mageed, M., 2019. UBC-NLP at SemEval-2019 Task 6: Ensemble Learning of Offensive Content With Enhanced Training Data, in: Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval). pp. 775–781. doi:10.18653/v1/s19-2136

Ranasinghe, T., Zampieri, M., Hettiarachchi, H., 2019. BRUMS at HASOC 2019: Deep Learning Models for Multilingual Hate Speech and Offensive Language Identification, in: FIRE 2019.

Resnik, P., 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy, in: IJCAI'95 Proceedings of the 14th International Joint Conference on Artificial Intelligence. Montreal, Quebec, Canada, pp. 448–453.

Revelle, W., Scherer, K.R., 2009. Personality and emotion. David Sander Klaus R. Scherer (eds.). Oxford Companion to Emot. Affect. Sci. 304–306. doi:10.1007/0-387-29907-6_6

Rezvan, M., Shalin, V.L., Sheth, A., 2018. A Quality Type-aware Annotated Corpus and Lexicon for Harassment Research, in: WebSci '18. Web Science. ACM. doi:10.1145/3201064.3201103

Rezvan, M., Shekarpour, S., Alshargi, F., Thirunarayan, K., Shalin, V.L., Sheth, A., 2020. Analyzing and learning the language for different types of harassment. PLoS One 15. doi:10.1371/JOURNAL.PONE.0227330

Robinson, D., Zhang, Z., Tepper, J., 2018. Hate Speech Detection on Twitter: Feature Engineering v . s . Feature Selection, in: ESWC. pp. 1–4.

Rosa, H., Pereira, N., Ribeiro, R., Ferreira, P.C., Carvalho, J.P., Oliveira, S., Coheur, L., Paulino, P., Veiga Simão, A.M., Trancoso, I., 2019. Automatic cyberbullying detection: A systematic review. Comput. Human Behav. 93, 333–345. doi:10.1016/j.chb.2018.12.021

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323, 533–536. doi:10.1038/323533a0

Salawu, S., He, Y., Lumsden, J., 2017. Approaches to Automated Detection of Cyberbullying:

A Survey. IEEE Trans. Affect. Comput. 3045, 1–20. doi:10.1109/TAFFC.2017.2761757

Saloot, M.A., Idris, N., Shuib, L., Raj, R.G., 2015. Toward Tweets Normalization Using Maximum Entropy, in: Proceedings of the ACL 2015 Workshop on Noisy User-Generated Text. pp. 19–27. doi:10.18653/v1/W15-4303

Samghabadi, N.S., Patwa, P., Pykl, S., Mukherjee, P., Das, A., Solorio, T., 2020. Aggression and Misogyny Detection using BERT : A Multi-Task Approach, in: Proceedings Ofthe Second Workshop on Trolling, Aggression and Cyberbullying LREC 2020. pp. 126–131.

Samih, Y., Maharjan, S., Attia, M., Kallmeyer, L., Solorio, T., 2016. Multilingual Code-switching Identification via LSTM Recurrent Neural Networks. Proc. Second Work. Comput. Approaches to Code Switch. 50–59. doi:10.18653/v1/W16-5806

Sampasa-Kanyinga, H., Roumeliotis, P., Xu, H., 2014. Associations between cyberbullying and school bullying victimization and suicidal ideation, plans and attempts among Canadian schoolchildren. PLoS One 9, e102145.

Sanchez, H., Kumar, S., 2011. Twitter Bullying Detection, in: NSDI. pp. 15–22.

Schapire, R.E., 1990. The Strength of Weak Learnability. Mach. Learn. 5, 197–227. doi:10.1023/A:1022648800760

Scherrer, Y., Ljubeši´c, N.L., 2016. Automatic normalisation of the Swiss German ArchiMob corpus using character-level machine translation, in: Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016). pp. 248–255.

Schuster, M., Paliwal, K.K., 1997. Bidirectional recurrent neural networks. IEEE Trans. Signal Process. 45, 2673–2681. doi:10.1109/78.650093

Schwartz, H.A., Eichstaedt, J.C., Kern, M.L., Dziurzynski, L., Ramones, S.M., Agrawal, M., Shah, A., Kosinski, M., Stillwell, D., Seligman, Martin E. P. Ungar, L.H., 2013. Personality, gender, and age in the language of social media: the open-vocabulary approach. PLoS One 8, e73791.

Scozzafava, F., Maru, M., Brignone, F., Torrisi, G., Navigli, R., 2020. Personalized PageRank with Syntagmatic Information for Multilingual Word Sense Disambiguation, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 37–46. doi:10.18653/v1/2020.acl-demos.6

Shannon, C.E., 1948. A Mathematical Theory of Communication. Bell Syst. Tech. J. 27, 379–423, 623–656.

Sheeba, J.I., Vivekanandan, K., 2013. Low Frequency Keyword Extraction with Sentiment Classification and Cyberbully Detection Using Fuzzy Logic Technique, in: IEEE International Conference on Computational Intelligence and Computing Research. pp. 1–5.

Shervin, M., Zampieri, M., 2017. Detecting Hate Speech in Social Media, in: Proceedings OfRecent Advances in Natural Language Processing. pp. 467–472.

Sidarenka, U., Scheffler, T., Stede, M., 2013. Rule-Based Normalization of German Twitter Messages, in: In Proc. of the GSCL Workshop Verarbeitung Und Annotation von Sprachdaten Aus Genres Internetbasierter Kommunikation.

Simov, K., Osenova, P., Popov, A., 2016. Towards Semantic-based Hybrid Machine

Translation between Bulgarian and English, in: The 2nd Workshop on Semantics-Driven Machine Translation. Association for Computational Linguistics, pp. 22–26. doi:10.18653/v1/W16-0604

Sinha, R.M.K., Thakur, A., 2005. Machine Translation of Bi-lingual Hindi-English (Hinglish) Text. 10th Mach. Transl. summit (MT Summit X) 149–156.

Slonje, R., Smith, P.K., 2008. Cyberbullying: Another main type of bullying? Scand. J. Psychol. 49, 147–154.

Solorio, T., Liu, Y., 2008. Part-of-speech tagging for English-Spanish code-switched text, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP '08. p. 1051. doi:10.3115/1613715.1613852

Squicciarini, A., Rajtmajer, S., Liu, Y., Griffin, C., 2015. Identification and characterization of cyberbullying dynamics in an online social network, in: IEEE/ACM International Confer-Ence on Advances in Social Networks Analysis and Mining. International Confer-Ence on Advances in Social Networks Analysis and Mining. pp. 280–285.

Sridhar, V.K.R., 2015. Unsupervised Text Normalization Using Distributed Representations of Words and Phrases, in: Proceedings of NAACL-HLT. pp. 8–16.

Stahlberg, F., 2020. Neural machine translation: A review. J. Artif. Intell. Res. 69, 343–418. doi:10.1613/JAIR.1.12007

Sutskever, I., Vinyals, O., Le, Q. V., 2014. Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems. pp. 3104–3112.

Tachicart, R., Bouzoubaa, K., 2021. Moroccan Data-Driven Spelling Normalization Using Character Neural Embedding. Vietnam J. Comput. Sci. 08, 113–131. doi:10.1142/s2196888821500044

Tachicart, R., Bouzoubaa, K., 2019. Towards Automatic Normalization of the Moroccan Dialectal Arabic User Generated Text, in: Smaïli K. (Eds) Arabic Language Processing: From Theory to Practice. ICALP 2019. Communications in Computer and Information Science, Vol 1108. Springer, Cham. pp. 264–275.

Tachicart, R., Bouzoubaa, K., Jaafar, H., 2016. Lexical Differences and Similarities between Moroccan Dialect and Arabic, in: 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt). pp. 331–337.

Tausczik, Y.R., Pennebaker, J.W., 2010. The psychological meaning of words: LIWC and computerized text analysis methods. J. Lang. Soc. Psychol. 29, 24–54. doi:10.1177/0261927X09351676

Tommasel, A., Rodriguez, J.M., Godoy, D., 2018. Textual Aggression Detection through Deep Learning, in: Proceedings Ofthe First Workshop on Trolling, Aggression and Cyberbullying. pp. 177–187.

Tyers, F.M., Sánchez-Martinez, F., Forcada, M.L., 2012. Flexible finite-state lexical selection for rule-based machine translation. Proc. th 16th Int. Conf. Eur. Assoc. Mach. Transl. 213–220.

Tyers, F.M., Sánchez-martínez, F., Ortiz-rojas, S., Forcada, M.L., 2010. Free / Open-Source Resources in the Apertium Platform for Machine Translation Research and

Development. Prague Bull. Math. Linguist. 67–76. doi:10.2478/v10108-010-0015-5.PBML

UNESCO, 2017. Violence et harcèlement à l'école: Rapport sur la situation dans le monde., in: International Symposium on School Violence and Bullying: From Evidence to Action, Seoul. p. 56 p.

United Nations, 2019. United Nations Strategy and Plan of Action on Hate Speech.

Van Der Goot, R., Cetinoglu, O., 2021. Lexical Normalization for Code-switched Data and its Effect on POS-tagging, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics. pp. 2352–2365.

Van Der Goot, R., Ramponi, A., Caselli, T., Cafagna, M., Mattei, L. De, 2020. Norm It ! Lexical Normalization for Italian and Its Downstream Effects for Dependency Parsing, in: Proceedings Ofthe 12th Conference on Language Resources and Evaluation (LREC 2020). pp. 6272–6278.

Van Der Goot, R., Van Noord, G., 2017. MoNoise: Modeling Noise Using a Modular Normalization System. Comput. Linguist. Netherlands J. 7, 129–144.

Van Geel, M., Goemans, A., Toprak, F., Vedder, P., 2016. Which personality traits are related to traditional bullying and cyberbullying ? A study with the Big Five , Dark Triad and sadism. PAID. doi:10.1016/j.paid.2016.10.063

Van Gompel, M., Van Den Bosch, A., 2014. Translation Assistance by Translation of L1 Fragments in an L2 Context. Assoc. Comput. Linguist. Conf. 871–880.

Vasilescu, F., Langlais, P., Lapalme, G., 2004. Evaluating Variants of the Lesk Approach for Disambiguating Words, in: LREC. Portugal.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Llion, J., N. Gomez, A., Kaiser, Ł., Polosukhin, I., 2017. Attention Is All You Need, in: 31st Conference on Neural Information Processing Systems. pp. 5998–6008.

Veltman, C.J., 1988. The future of the Spanish language in the United States.

Vickrey, D., Biewald, L., Teyssier, M., Koller, D., 2005. Word-Sense Disambiguation for Machine Translation, in: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT '05). pp. 771–778. doi:10.3115/1220575.1220672

Voss, C., Tratz, S., Laoudi, J., Briesch, D., 2014. Finding Romanized Arabic Dialect in Code-Mixed Tweets, in: Proceedings of the 9th International Conference on Language Resources and Evaluation. pp. 188–199.

Wang, L., Fuketa, M., Morita, K., Aoe, J., 2011. Context constraint disambiguation of word semantics by field association schemes. Inf. Process. Manag. 47, 560–574. doi:10.1016/j.ipm.2011.01.001

Wang, Y., Wang, M., Fujita, H., 2019. Knowledge-Based Systems Word Sense Disambiguation : A comprehensive knowledge exploitation framework ☆. Knowledge-Based Syst. 190, 1–13. doi:10.1016/j.knosys.2019.105030

Wiegand, M., Siegel, M., 2018. Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language, in: Proceedings of GermEval 2018, 14th

Conference on Natural Language Processing (KONVENS 2018). pp. 1–10.

Wilks, Y., Stevenson, M., 1997. The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation. Nat. Lang. Eng. 4, 135–143. doi:10.1017/S1351324998001946

William, J., 1890. THE EMOTIONS., in: The Principales of Psychology.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V, Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., Dean, J., 2016. Google ' s Neural Machine Translation System : Bridging the Gap between Human and Machine Translation. arXiv Prepr. arXiv1609.08144 1–23.

Xu, J., Zhu, X., Bellmore, A., 2012. Fast Learning for Sentiment Analysis on Bullying, in: WISDOM'12. International Workshop on Issues of Sentiment Discovery and Opinion Mining. ACM.

Yao, M., Chelmis, C., Zois, D.-S., 2018. Cyberbullying Detection on Instagram with Optimal Online Feature Selection, in: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. pp. 401–408.

Yarkoni, T., 2010. Personality in 100,000 Words: A large-scale analysis of personality and word use among bloggers. J. Res. Pers. 44, 363–373. doi:10.1016/j.jrp.2010.04.001

Yarowsky, D., 1995. UNSUPERVISED WORD SENSE DISAMBIGUATION RIVALING SUPERVISED METHODS, in: The 33rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 189–196.

Yarowsky, D., 1993. One Sense Per Collocation, in: Proceedings of the Workshop on Human Language TechnologyHLT '93. pp. 266–271. doi:10.3115/1075671.1075731

Yatabe, R., Sasaki, M., 2020. Semi-supervised Word Sense Disambiguation Using Example Similarity Graph, in: Proceedings of the Graph-Based Methods for Natural Language Processing (TextGraphs). pp. 51–59.

Yoav Goldberg, 2017. Neural Network Methods for Natural Language Processing.

Yuan, D., Richardson, J., Doherty, R., Evans, C., Altendorf, E., 2016. Semi-supervised Word Sense Disambiguation with Neural Models, in: Proceedings of COLING. pp. 1374–1385.

Zarnoufi, R., Abik, M., 2020. Big Five Personality Traits and Ensemble Machine Learning to Detect Cyber-Violence in Social Media, in: Serrhini M., Silva C., Aljahdali S. (Eds) Innovation in Information Systems and Technologies to Support Learning Research. EMENA-ISTL 2019. Learning and Analytics in Intelligent Systems. pp. 194–202. doi:10.1007/978-3-030-36778-7_21

Zarnoufi, R., Boutbi, M., Abik, M., 2020a. AI to prevent cyber-violence: Harmful behaviour detection in social media. Int. J. High Perform. Syst. Archit. 9, 182–191. doi:10.1504/IJHPSA.2020.113679

Zarnoufi, R., Jaafar, H., Abik, M., 2020b. Machine Normalization : Bringing Social Media Text from Non-Standard to Standard Form. ACMTrans. Asian Low-Resour. Lang. Inf. Process. 19, 30.

Zarnoufi, R., Jaafar, H., Abik, M., 2019. Language Identification for User Generated Content in Social Media, in: Rocha, Á., Serrhini, M. (Eds.), Information Systems and Technologies to Support Learning. EMENA-ISTL 2018. Smart Innovation, Systems and Technologies. Springer, Cham, pp. 672–678. doi:10.1007/978-3-030-03577-8_73

Zarnoufi, R., Jaafar, H., Bachri, W., Abik, M., 2020c. MANorm: A Normalization Dictionary for Moroccan Arabic Dialect Written in Latin Script, in: Proceedings of the Fifth Arabic Natural Language Processing Workshop/COLING 20202. pp. 155–166.

Zhang, W., Clark, R.A.J., Wang, Y., 2016. Unsupervised Language Filtering using the Latent Dirichlet Allocation. Comput. Speech Lang. 39, 47–66.

Zhong, Z., Ng, H.T., 2010. It Makes Sense : A Wide-Coverage Word Sense Disambiguation System for Free Text, in: Proceedings of the ACL 2010 System Demonstrations. pp. 78–83.

Ziamari, K., 2003. Code switching intra-phrastique dans les conversations des étudiants marocains de l'ENSAM: approche linguitique du duel entre l'arabe marocain et le français.

Annexe A

Tableau A.1 Règles de conversion de l'IPA adaptée à l'écriture latine utilisée pour le dialecte AM, et son équivalent en phonème AM avec le symbole IPA correspondant

Script IPA adapté utilisé dans le dictionnaire AM	Script Latin adopté en SM	Phonème AM	Symbole API
ḥ	7	ح	ħ
ḍ/ đ	d	ض	dˤ
ɛ / Ɛ	3	ع	ʕ
ġ	gh	غ	ɣ
ħ	h	ه	h
ḫ / x	kh	خ	x
ḷ	l	ل (géminé)	l
ṛ / ṛ / ř	r	ر	r
ṣ	s	ص	sˤ
š / š	ch	ش	š
ṭ / ṭ	t	ط	tˤ
ž	j	ج	ʒ
ẓ / ż	z	ز	z
â	a	ا	ʔ / a
ə	e	-	-
î	i	ي	i
û	ou	و	u

Annexe B

Algorithme de traduction-désambiguïsation des textes CS :

Input: target words *wt // wt* is the word to translate from a Source into a Target Language
Output: the translation of target words.
Method:
- for each *wt* in input text do
    - extract all context words for each occurrence of *wt* in the entire text // *construction of context words set Wc*
    - if the *wt* language is Standard Language then
        - extract *wt* translations into Target Language from the bilingual semantic dictionary according to its P.O.S.
    - else (it is a dialect)
        - get their translations from the dialect bilingual dictionary according to its P.O.S.
    - extract the set of Wc
    - for each wc in Wc do
        - if the wc language = Source Language then
            - if the wc language is Standard Language then
                - extract wc *translations* to Target Language from the bilingual semantic dictionary according to its P.O.S.
            - else if (it is a dialect)
                - get its translations from the dialect bilingual dictionary according to its P.O.S.
            - else (wc language is the Target Language)
                - if the wc language is Standard Language then
                    - extract its *senses* in Target Language from the semantic dictionary according to its P.O.S.
                - else (it is a dialect)
                    - get its *senses* from the dialect dictionary according to its P.O.S.
- for each wt synset translation in N translation synsets do
    - get the glosses, examples, hypernyms and hyponyms.
    - for each wc in Wc (launch in parallel) do
        - for each wc synset translation in N translation synsets do
            - get the glosses, examples, senses, hypernyms and hyponyms.
            - count the overlapping score between each wt translation glosse, example, hypernym and hyponym and those of wc translation.
- find the maximum overlapping score within wt translations.
- if the scores are not null then
    - select the wt translation with the maximum score.
- else
    - select the Most Frequent Translation of wt senses.
- return the translation of the target word.

Annexe C

Exemple des glosses des synsets fournis par BabelNet pour le mot "preuve" :

Tableau C.1 Synsets BabelNet du mot 'evidence' avec le premier gloss en Anglais et en Français

Partie du discours	Babel Synsets (synonyms)	Traductions en français
Nom	Evidence, ground: Your basis for belief or disbelief; knowledge on which to base belief.	Preuve, évidence : Une évidence est ce qui s'impose à l'esprit comme une vérité, ou une réalité, sans qu'il soit besoin d'aucune preuve ou justification.
	Evidence : An indication that makes something evident	Preuve : pas de définition disponible.
	Evidence : (law) all the means by which any alleged matter of fact whose truth is investigated at judicial trial is established or disproved	Preuve, preuve en droit civil français : La preuve définit tout moyen utilisé pour établir l'existence d'un fait ou droit dont on se prévaut.
	Sign, mark, evidence: A perceptible indication of something not immediately apparent (as a visible clue that something has happened)	Sign : pas de définition disponible.
	Evidence (policy debate) : Evidence in policy debate consists mainly of two parts.	Non disponible
Verbe	Demonstrate, attest, certify, evidence : Provide evidence for; stand as proof of; show by one's behavior, attitude, or external attributes	Assurer, attester, démontrer, exprimer, indiquer : pas de définition disponible.
	Testify, evidence, bear witness : Provide evidence for	Témoigner : pas de définition disponible.
	Evidence tell : give evidence	Dire, démontrer, montrer, preuver, témoigner : pas de définition disponible.