



Département d'Informatique

Mémoire présenté par :

Houari AbdeLaziz et Benchabekh Mohamed El Mehdi

Pour l'obtention du diplôme de Master

Domaine : Mathématique et Informatique

Filière : Informatique Spécialité

Spécialité : Traitement Automatique de la Langue

Sujet:

**Conception et réalisation d'un système d'identification
Des textes offensants dans les réseaux sociaux**

Soutenu le : 24/11/2020, devant le jury composé de

Mme. N.Ben Blidia	-Université de Blida 1	Présidente
Mme. M.Mezzi	- Université de Blida 1	Examinatrice
Mr. M.ABBAS	- CRSTDLA	Encadreur
Mme. S.Oukid	- Université de Blida 1	Promotrice

REMERCIEMENTS :

Merci chaleureusement à tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet de fin d'étude.

Nous tenons à remercier Docteur Mourad Abbas pour l'honneur qu'il nous a fait en nous proposant le sujet de ce mémoire de fin d'étude. Nous avons eu aussi l'honneur de travailler avec Monsieur Mohamed Lichouri et de profiter de ses qualités humaines et professionnelles. Ainsi que tous les enseignants de l'université Saad Dahlab – Blida 1 en particulier ceux et celles du département informatique.

Mes remerciements vont également aux membres de jury d'avoir accepté de juger notre travail.

Dédicace

Une dédicace spéciale à Mes chers parents, je ne trouve même pas les mots pour exprimer mes sentiments envers la chose la plus précieuse que dieu

M'a offert.

Merci à mes chers frères, ma famille, mon binôme et ami à la fois Abdelaziz, et tous ceux qui m'ont aidé durant mon passage à Blida.

Mehdi

Dédicace

Je dédie ce travail à:

*A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse,
leur soutien et leurs prières tout au long de mes études,*

*A mes chères sœurs pour leurs encouragements permanents, et leur soutien
moral,*

*A mes chers frères, pour leur appui et leur encouragement, et mon cher
binôme mehdi.*

*A Mes encadreurs Mes sieurs Mourad Abbas et Lichouri Mohamed qui
m'ont donné le courage de continuer.*

*A toute ma famille pour leur soutien tout au long de mon parcours
universitaire,*

*Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de
votre soutien infaillible,*

Merci d'être toujours là pour moi.

Houari

Résumé :

Le texte offensant est omniprésents dans les réseaux sociaux, faire face à ce genre de comportements abusifs est devenu de plus en plus urgent, un des défis majeurs pour le monde informatique.

En se basant sur le corpus qui nous a été fourni nous avons eu affaire à des textes (des commentaires extraits de Twitter) écrits en deux langues (Arabe et Anglais). Notre travail vise à la mise en œuvre d'un système utilisant les techniques de classification supervisées en s'appuyant sur des méthodes de l'apprentissage automatique du domaine d'intelligence artificielle et du traitement automatique de la langue en particulier, comme la machine learning et le deep learning, afin de pouvoir arriver justement à identifier ce genre de textes, en l'occurrence. Les textes offensants selon ces critères : Offensant ou pas, ciblé ou pas, vers un individu ou vers un group ou ni l'un ni l'autre, en ce qui concerne la partie Arabe nous avons traité seulement le premier niveau (Offensant ou pas) .

Après avoir effectué les tests et les comparer à plusieurs niveaux, nous avons obtenu des résultats sur lesquels nous avons constaté que LSTM était le meilleur dans notre cas pour l'apprentissage automatique (Machine learning), du même pour l'apprentissage profond (Deep learning) nous avons constaté que LSTM était le meilleur.

Mots clés :

Réseaux sociaux, textes offensants, classification, Supervisé, Apprentissage .

ملخص :

النص المسيء منتشر في كل مكان في الشبكات الاجتماعية ، وقد أصبحت مواجهة هذا النوع من السلوك أكثر إلحاحًا ، وهو أحد التحديات الرئيسية لعالم تكنولوجيا المعلومات.

استنادًا إلى مجموعة النصوص المقدمة إلينا، تعاملنا مع نصوص مكتوبة بلغتين (العربية والإنجليزية). يهدف عملنا إلى تنفيذ نظام باستخدام تقنيات التصنيف الخاضعة للإشراف التي تعتمد على أساليب التعلم الآلي في مجال الذكاء الاصطناعي والمعالجة الآلية للغة على وجه الخصوص، مثل التعلم الآلي والتعلم العميق ، من أجل التمكن من تحديد هذا النوع من النص بدقة ، في هذه الحالة. النصوص المسيئة وفق هذه المعايير: مسيئة أم لا، مستهدفة أم لا، تجاه فرد أو تجاه مجموعة... فيما يخص اللغة العربية فقد تعاملنا فقط مع المستوى الأول (هجومى أم لا).

بعد إجراء الاختبارات ومقارنتها على عدة مستويات، حصلنا على النتائج حيث وجدنا أن LSVC كانت الأفضل في حالة التعلم الآلي، فيما يخص التعلم العميق. وجدنا أن LSTM هي الأفضل.

الكلمات الدالة :

الشبكات الاجتماعية، النصوص المسيئة، التصنيف، الإشراف، التعلم الآلي.

Summary

Offensive text is omnipresent in social networks, facing this kind of abusive behavior has become more and more urgent, one of the major challenges for the IT world.

Based on the corpus given to us, we will be dealing with texts (comments taken from Twitter) written in two languages (Arabic and English). Our work aims at the implementation of a system using supervised classification techniques relying on machine learning methods from the field of artificial intelligence and automatic language processing in particular, such as machine and deep learning, in order to be able to identify precisely this type of text, in this case Offensive text. According to these criteria: Offensive or not, targeted or not, towards an individual, a group or neither, for Arabic part we have only dealt with the first level (offensive or not).

After performing the tests and comparing them on several levels, we obtained results on which we found that LSTM was the best in our case for Machine learning, the same for Deep learning, we found LSTM to be the best.

Keywords:

Social networks, offensive texts, classification Supervised, Learning .

Table des matières

Introduction générale.....	2
Chapitre 1 Les réseaux sociaux et les textes offensants.....	3
I.1. Introduction.....	4
I.2. Définition des réseaux sociaux.....	4
I.3. Les limites des réseaux sociaux.....	5
I.3.1. Facebook.....	5
I.3.2. Twitter.....	5
I.4. Les techniques et les applications du TAL appliquées sur les réseaux sociaux.....	6
I.4.1. Le TAL et les réseaux sociaux.....	6
I.4.2. Domaines d'application.....	7
I.4.2.1. Secteur industriel.....	7
I.4.2.2. Défense et sécurité nationale.....	8
I.4.2.3. Secteur sanitaire.....	8
I.4.2.4. Politique.....	8
I.5. Les textes offensant nourris par les discours de haine.....	8
I.5.1. Les types des textes offensants.....	9
I.5.2. Exemple des textes offensants.....	10
I.6. Extraction des mots clés et Classification des textes offensants.....	10
I.7. Les travaux précédant.....	11
I.7.1. Corpus anglais.....	11
I.7.2. Corpus arabe.....	14
I.7.3. Corpus Danois.....	14
I.8. Conclusion.....	14
Chapitre II La classification, ses différents aspects et complexités.....	15
II.1. Introduction.....	16
II.2. Définition de la classification.....	16
II.2.1 l'explication formelle.....	17
II.2.2. l'automatisation de la classification.....	17
II.2.3.les méthodes de la classification.....	19
II.2.3.1. Apprentissage non supervisé «Clustering».....	19
II.2.3.2. Apprentissage supervisé «Catégorisation».....	20
II.2.3.3. Avantages et inconvénients.....	21
II.3. Problèmes rencontrés dans la catégorisation de textes.....	22
II.3.1. Sur-apprentissage.....	22
II.3.2. L'homographie.....	22
II.3.3. Polysémie (Ambiguïté).....	22

II.3.4. La graphie	23
II.3.5. Redondance(Synonymie).....	23
II.3.6. Présence-Absence de termes	23
II.3.7. Subjectivité de la décision	23
II.4. Particularité de la langue arabe.....	24
II.4.1. Alphabet.....	24
II.4.2. la classification dans la langue arabe.....	25
II.4.3. Le partitionnement.....	25
II.4.4. Les voyelles	26
II.4.5. L'agglutination.....	26
II.4.6. Ambiguïté	26
II.5. Conclusion	26
Chapitre III Conception et Architecture du système d'identifications des textes offensants.....	27
III.1. Introduction	28
III.1.1. Problématique.....	28
III.1.2. L'objectif	28
III.2. Conception et l'architecture Générale.....	29
III.2.1 « Dataset » (corpus) :	31
III.2.2.Schéma d'étiquetage	34
III.2.2.1. Niveau A (sous tache_A).....	35
III.2.2.2.Niveau B (sous tache_B).....	35
III.2.2.3.Niveau C : (sous tache_C).....	35
III.3. Technique et méthodes des prétraitements	36
III.3.1.1. Passage en minuscule	38
III.3.1.2. « Tokenisation » :.....	38
III.3.1.3. la suppression des mots vides.....	39
III. 3.1.4. Groupement sémantique.....	40
III.3.2. La représentation des expériences.....	42
III.3.3.Représentations vectorielles des Tweets	46
III. 3.3.1 Le Bag of Word(BoW)	46
III. 3.3.2 Calcul du TF-IDF	47
III.3.3.3. Réduction de la taille des vecteurs obtenus.....	48
III.3.3.4. « Mot Embedding » :.....	48
III.4. Les Modèles d'apprentissage automatique et profond.....	51
III.4.1. Machine learning (ML) :	51
III.4.1.1. Algorithmes de régression	52

III. 4.1.2 Algorithmes basés sur des instances.....	53	
III.4.1.3 Algorithmes d'arbre de décision	58	
III.4.1.4. Algorithmes bayésiens.....	59	
III.4.1.5 Algorithmes d'ensemble	60	
III.4.2. Les Algorithmes d'apprentissage profond	65	
III.4.2.1. Réseaux de neurones récurrents (RNN)	65	
III.4.2.2. Long short Term memory (LSTM)	69	
III.4.2.3.les fonctions d'activation.....	71	
III.5. les avantages et les inconvénients	73	
Chapitre 4	Les résultats obtenus et présentation de l'application	77
III.1. Introduction.....	78	
III.2. L'environnement du développement.....	78	
III.2.1. Software	78	
III.2.2. HardWare	86	
III.3. Description du système	86	
III.3.1. Prétraitement	86	
III.3.2. Les métriques d'évaluation :.....	87	
III.3.2.1. Classification binaire <binary>.....	87	
III.3.2. Classification multi-classes.....	88	
III.3.2.2. Classification multi-classes.....	88	
III.3.2.3 sous tache _A	89	
III.3.2.4 sous tache _B	92	
III.3.2.5 Sous tache _C.....	95	
III.3.2.6 Sous tache _Arabic	95	
III.3.3. Les résultats obtenus	97	
III.3.3.1. Corpus d'anglais	97	
III.3.3.2. Corpus arabe	104	
III.3.4. L'apprentissage profond.....	106	
III.3.4.1. Modèle LSTM.....	106	
III.3.3.2. Modèle Simple RNN.....	110	
III.3.3.3.Comparision avec les travaux	111	
III.4. Application :.....	112	
III.5.conclusion	117	
Conclusion générale et perspectives	119	
Références	120	
Références	121	

Liste des Figures

<i>Figure 1 : les mots offensants et les réseaux sociaux (web)</i>	10
<i>Figure 2 Extraction des infos et des connaissances (carin)</i>	17
<i>Figure 3 Extraction Automatique (web)</i>	19
Figure 4 : Apprentissage supervisé/Non supervisé.	21
Figure 5 : Schéma du système général «tweet prédiction»	29
Figure 6 : Architecture de processus du prétraitement	30
Figure 7 graphe montre le nombre du tweets sous tache A.....	31
Figure 8 : graphe montre le nombre des tweets sous tache_B	32
Figure 9 : graphe montre le nombre des tweets sous tache_c	33
Figure 10 : exemple du fitting pour le classificateur KNN	35
Figure 11 : les processus du prétraitement	36
Figure 12 : un exemple de tokenisation des tweets anglais.....	38
Figure 13 : un exemple de « tokenisation » des tweets arabes.....	39
Figure 14 : Mots vides pour la langue anglaise.....	39
Figure 15 : Mots vides pour la langue arabe	40
Figure 16 : « stemmatisation » pour la langue anglaise	41
Figure 17 : la stemmatisation en arabe.....	42
Figure 18 : la lemmatisation en anglais.....	42
Figure 19 : Architecture de processus du prétraitement	45
Figure 20 : Different representation vectorielle	46
<i>Figure 21 : Exemple du bag of Mot</i>	47
<i>Figure 22 : architecture simple du modèle CBOW</i>	49
Figure 23 : architecture simple du modèle Skip-Gram	50
<i>Figure 24 : Modèle logistique régression (LR) Wikipédia</i>	53
<i>Figure 25 : modèle Linéaire Support Machine (wikipédia)</i>	55
<i>Figure 26 : Modèle KNN</i>	57
<i>Figure 27 : Arbre de décision (DT)</i>	59
<i>Figure 28 : Forêt aléatoire</i>	62
Figure 29 : Gradient Boosting (GB) Wikipedia	64
30 : Le processus du RNN	65
<i>Figure 31 : Architecture simple de réseaux de neurones</i>	66
<i>Figure 32 : Architecture pour le processus Du simple RNN</i>	68
<i>Figure 33 : Architecture représente les portes du LSTM</i>	69
<i>Figure 34 : exemple du LSTM avec sigmoïde activation</i>	71
<i>Figure 35 : Logo du python</i>	79
<i>Figure 36 : l'environnement de anaconda</i>	80
<i>Figure 37 : Logo du spyder IDE</i>	80
<i>Figure 38 : Exemple du batshScript</i>	81
<i>Figure 39 : Logo de langage de balise HTML</i>	82
<i>Figure 40 : Logo de langage CSS</i>	82
<i>Figure 41 :Logo de JavaScript</i>	82
<i>Figure 42 : Logo Fire Base</i>	83
<i>Figure 43 : Logo du framework Angular</i>	84
<i>Figure 44 : Framework Flask</i>	85
<i>Figure 50 : Classe réelle et classe de prédiction</i>	88
<i>Figure 45 : Histogramme sous tache A</i>	90
<i>Figure 48 : Les paramètres de configuration LSTM</i>	107
<i>Figure 49: Montre les paramètres du lstm pour multi-classes</i>	108

<i>Figure 51 : Interface d'accueil.....</i>	112
<i>Figure 52 : Détection du tweet dans les textes anglais(offensive)</i>	113
<i>Figure 53 : Catégorisation des tweet (sous tache_B)</i>	114
<i>Figure 54 : Détection de type du tweet arabe (offensive)</i>	115
<i>Figure 55 : Montre le genre du tweet (sous tache_C)</i>	115
<i>Figure 56 : Détection de type du tweet arabe (not offensive)</i>	116

Liste des Tableaux

<i>Table 1 : les statistiques pour sous tache (A)</i>	12
<i>Table 2 : les statistiques pour Sous tache (B)</i>	12
<i>Table 3 :les statistiques pour Sous tache (C)</i>	12
<i>Table 4 : Montre Macro-F1 score pour les modèles du corpus (OLID)</i>	13
<i>Table 5 : Résultats pour arabe sous tache (A) macro-average F1 Score</i>	14
<i>Table 6 : Résultats pour le Danois sous tache (A) macro-Average F1 Score</i>	14
<i>Table 7 : Répartition des combinaisons d'étiquettes pour la tâche A</i>	31
<i>Table 8 : Répartition des combinaisons d'étiquettes pour la tâche B</i>	32
<i>Table 9 : Répartition des combinaisons d'étiquettes pour la tâche C</i>	32
<i>Table 10 : Exemples de Tweet de l'ensemble de données, avec leurs labels correspondants pour chaque sous tache.</i>	34
<i>Table 11 : Les différentes fonctions d'activation</i>	72
<i>Table 12 : les avantages et les inconvénients de différents algorithmes d'apprentissage</i>	75
<i>Table 13 : Représente les Caractéristiques du hardware</i>	86
<i>Table 14 : Les scores avant le prétraitement pour sous tache _A stage zéro</i>	89
<i>Table 15 : Résultats des classificateurs après remove stopMot sous tache _A stage 1</i>	90
<i>Table 16 : Résultats des classificateurs après remove ponctuation sous tache _A stage 1</i>	91
<i>Table 17 : Résultats des classificateurs après remove stopMot sous tache _A stage 1</i>	91
<i>Table 18 : Résultats des classificateurs pour sous tache _A stage 4</i>	92
<i>Table 19 : Résultats des classificateurs avant le prétraitement sous tache _B stage zéro</i>	92
<i>Table 20 : Résultats des classificateurs après remove stopMot sous tache _B stage 1</i>	93
<i>Table 21 : Résultats des classificateurs après « remove ponctuation » sous tache _B stage 1</i> 93	
<i>Table 22 : Résultats obtenus après l'utilisation des Pos Tag sous tache _B stage 1</i>	94
<i>Table 23 : Résultats obtenus pour sous tache _B stage 4</i>	94
<i>Table 24 : résultats des classificateurs avant le prétraitement sous tache _B stage zéro</i>	95
<i>Table 25 : Résultats des classificateurs avant le prétraitement pour sous tache _Arabic stage zéro</i>	96
<i>Table 26 : Résultats des classificateurs pour sous tache _Arabic stage 04</i>	96
<i>Table 27 : Les résultats pour sous tache _A stage 0 expérience 3</i>	98
<i>Table 28 : Les résultats de sous tache _A du stage 5 expérience 3</i>	99
<i>Table 29 : Les résultats de sous tache _B du stage 0 expérience 03</i>	100
<i>Table 30 : Les résultats de sous tache _B du stage 3 expériences 03</i>	101
<i>Table 31 : les résultats de sous tache _B du stage 5 expériences 04</i>	102
<i>Table 32 : Les résultats de sous tache _C du stage 1 expérience 03</i>	104
<i>Table 33 : Les résultats de sous tache _C du stage 5 expériences 03</i>	104
<i>Table 34 : Les résultats de tache arabe stage 4 expérience 11</i>	105
<i>Table 35 : les résultats de tache arabe stage 5 expérience 4</i>	106
<i>Table 36: les résultats d'évaluation du modèle LSTM.</i>	109
<i>Table 37 : Montre les graphes du LSTM en différents taches</i>	110
<i>Table 38 : Les résultats d'évaluation du modèle Simple RNN</i>	110
<i>Table 39 : Montre les graphes du SimpleRNN en différents taches</i>	111
<i>Table 40 : Comparaison avec les travaux précédant</i>	111

Liste des abréviations

TAL :	Traitement Automatique de la Langue
TF :	Fréquence de Terme
IDF :	Fréquence d'Inverse de Document
TF_IDF :	Terme Fréquence – Fréquence Inverse du Document
SVM :	support vector machine
LSVC:	leanier support vector machine
KNN :	Plus Proche Voisin
ML:	Machine Learning
IA:	Intelligence Artificielle
DL:	deep learning
DT:	Arbre de Décision
RF:	Forêt Aléatoire
BNB:	Bernoulli Naïve Bayes
MNB:	Multinomiale Naïve Bayes
GB:	Gradient Boosting
BOW:	Bag-Of-Mot
NB:	Naïve Bayes
LR:	Logistique Régression
MNB:	Multinomial Naïve Bayes
SimpleRNN:	Simple Recurrent neural networks
LSTM:	Long short Term Memory
CNN:	Convolutional neural network
OFF:	Offensive (offensant)
Not:	Not Offensive (non offensant)

TIN:	Target (ciblé)
UNT:	Untarget (pas ciblé)
GRP:	Group (groupe)
IND:	Individual (individuel)
Oth:	Other (autre)
ASMS :	American Society for Mass Spectrometry

Introduction Générale

Introduction générale

Aujourd'hui les réseaux sociaux occupent la place numéro 1 des moyens de communications modernes, il en est d'ailleurs le plus utilisé sur internet. Cependant, grâce à cette popularité et pour des raisons complètement malsaines menés par des comportements méchants voir même interdits, beaucoup d'utilisateurs profitent des options fournissent par ces réseaux comme l'anonymat pour mettre dans les commentaires des textes offensants et des contenus abusifs qui peuvent faire du mal aux gens . le monde informatique et les différentes entreprises technologiques continuent à mettre les moyens pour combattre ce phénomène.

Pour cela Nous avons proposé un travail qui se concentre sur la classification du contenu abusif des commentaires sur twitter en utilisant dix algorithmes d'apprentissage automatique et profond, à savoir l'objectif est de :

- Réaliser un système qui va nous permettre de prédire les cibles des commentaires de twitter et les classer selon des différentes catégories.
- connaître les meilleurs modèle d'apprentissage automatique et profond et quelle sont les meilleures représentations vectorielles (optimale) ?

Afin d'atteindre l'objectif cité précédemment, nous avons proposé cette structure de travail :

Le premier chapitre : Nous allons introduire des notions générales sur les domaines des : Réseaux sociaux, textes offensants, TAL en donnant quelques définitions, les taches principales, les applications de chacune et surtout la relation entre ces différents concepts.

Le deuxième chapitre : vise à présenter le processus de la classification et la catégorisation des textes, avec une illustration des techniques d'apprentissages avec leurs avantages et leurs inconvénients ainsi que les problèmes et les difficultés liées à ces aspects.

Le troisième chapitre : est dédié à la présentation du prétraitement avec ses techniques puis les différents algorithmes d'apprentissage supervisé (automatique et profond). Nous avons également introduit les différents moyens d'évaluation des classificateurs.

Le dernier chapitre : Après avoir évalué et comparer entre nos différents Algorithmes nous allons enfin pouvoir mettre notre Algorithme choisi (Le meilleur) en œuvre, et aussi faire une évaluation des performances des différentes approches implémentées tout en présentant les résultats obtenus avec interprétations.

Et nous allons terminer par une conclusion générale.

Chapitre 1

Les réseaux sociaux et les textes offensants

I.1. Introduction

Le réseau social est devenu un endroit où des gens de tous les coins du monde ont établi une civilisation virtuelle, Dans cette communauté virtuelle, les gens avaient l'habitude de partager leurs points de vue, d'exprimer leurs sentiments, des photos, des vidéos, des blogs, etc. Des sites de réseautage social comme Facebook, Twitter, YouTube, etc... ont donné une plate-forme pour partager d'innombrables contenus en un seul clic. Cependant, aucune restriction n'est appliquée concernant le contenu téléchargé .Ces contenus téléchargés peuvent contenir des mots abusifs, des images explicites qui peuvent ne pas convenir aux normes sociales. En tant que tel, il n'existe aucun mécanisme défini pour empêcher la publication de contenus offensants sur les sites sociaux. Afin de résoudre ce problème, pour cela, nous avons développé un prototype d'essai pour mettre en œuvre notre approche de filtrage automatique des contenus offensants dans les réseaux sociaux.

De nombreux sites de réseaux sociaux populaires ne disposent pas aujourd'hui de mécanisme approprié pour restreindre les contenus offensants .Ils utilisent des méthodes de rapport dans lesquelles l'utilisateur signale si le contenu est abusif.

Cela nécessite des efforts humains importants et du temps. Dans notre travail, nous avons appliqué un algorithme pour la détection de contenus offensants à partir des commentaires des réseaux sociaux. Différemment à la méthode conventionnelle de signalement de contenus abusifs par les utilisateurs, notre approche ne nécessite aucune intervention humaine ainsi les mots offensants en les détectant et en les classifiant automatiquement on peut avoir un mécanisme approprié pour restreindre le contenu offensant. [1]

I.2. Définition des réseaux sociaux

Dans le domaine des technologies, un réseau social est un service permettant de regrouper des personnes afin de créer un échange sur un sujet particulier ou non. En quelque sorte, le réseau social trouve ses origines dans les forums, groupes de discussion et salons de chat introduits dès les premières heures d'Internet.

Depuis le début des années 2000, la présence des réseaux sociaux, également appelés réseaux communautaires, devient de plus en plus importante et tend à se multiplier selon diverses caractéristiques.

Les premiers réseaux sociaux de grande envergure (My Space et Facebook ...) se sont positionnés en tant que services généralistes sur lesquels chacun peut partager le contenu de son choix, quel qu'en soit le sujet, avec ses contacts.

Où s'arrête la liberté d'expression sur les médias sociaux ?

La libre communication des pensées et des opinions est un des droits les plus précieux de l'Homme : tout Citoyen peut donc parler, écrire, imprimer librement, sauf à répondre de l'abus de cette liberté, dans les cas déterminés. [2]

I.3. Les limites des réseaux sociaux

Commençons par savoir ce qu'il est permis de dire ou de montrer sur les médias sociaux :

I.3.1. Facebook

Facebook veille à respecter la liberté d'expression de chacun en reflétant la diversité de ses utilisateurs, cependant, certains contenus peuvent être signalés et supprimés :

- Violences et menaces
- Suicide ou automutilation
- Intimidation ou harcèlement
- Discours incitant à la haine
- Contenu explicite

I.3.2. Twitter

Twitter explique que l'exhaustivité des contenus publiés ne permet pas de pouvoir tout contrôler et qu'il ne peut donc être tenu responsable d'un contenu offensant, inexact, inapproprié publié aux yeux de tous.

Cependant, certains contenus sont considérés comme des infractions au règlement :

- Usurpation de l'identité d'une personne ou d'une marque
- Utilisation non autorisée d'une marque déposée
- Informations privées publiées sur Twitter
- Comportement abusif et menaces violentes
- Utilisation non autorisée de documents protégés par copyright

Cette grande liberté d'expression sans limites que nous avons l'impression de trouver sur les différents médias sociaux existants pousse certains d'entre nous à agir sur Internet tel qu'ils ne l'auraient pas fait dans la réalité.

Selon le psychologue « John Suler », il existe plusieurs facteurs pouvant conduire à la désinhibition :

- L'anonymat d'un pseudonyme (plus ou moins réel).
- l'invisibilité aux autres.
- Le laps de temps entre l'envoi du message et la réception du feedback.
- Le fait d'être seul devant son clavier.
- L'absence des figures d'autorité en ligne.

Les signaux physiques que nous envoyons lors d'une conversation en face à face (ton de la voix, expressions faciales, ...) permettent un ajustement social. Sans ces indications, la communication en ligne est source de mauvaises interprétations et facilite les débordements.

De plus, on observe sur les médias sociaux le phénomène de regroupement en communauté.

le débat. Au contraire, cela ne fait que cloisonner les avis des uns et des autres, ne laissant apparaître que des confrontations peu constructives la plupart du temps. [3]

I.4. Les techniques et les applications du TAL appliquées sur les réseaux sociaux

I.4.1. Le TAL et les réseaux sociaux

Le traitement automatique de données extraites des réseaux sociaux doit arriver à déterminer les méthodes les plus appropriées pour l'extraction d'information.

La classification automatique, l'indexation de données pour la recherche documentaire ou la traduction automatique par exemple. On sait que le seul volume des données et la vitesse à laquelle de nouveaux contenus sont créés suffisent à rendre irréalisable toute tentative de veille ou d'analyse manuelle significative. La veille des médias sociaux est l'une des plus importantes applications de l'ASMS. Comme dans sa définition traditionnelle, la veille consiste en l'activité de surveillance et de suivi, le contenu en ligne et les médias de diffusion, notamment dans un but politique, commercial ou scientifique L'importante quantité

d'information accessible dans les médias sociaux représente une manne de renseignements. Ces derniers ajoutent une dimension absente des médias traditionnels en nous informant sur les opinions et sentiments des auteurs.

L'objectif de la recherche documentaire dynamique et de la recherche d'événements en temps réel est de mettre en place des stratégies de recherche efficaces à partir de différentes fonctionnalités qui tiennent compte de multiples dimensions, y compris les liens spatiaux et temporels. Dans un tel cas, certaines méthodes de TAL, par exemple, la recherche documentaire et le résumé automatique de données sous forme de documents de diverses sources, deviennent essentielles à la recherche d'événements et à la détection d'information pertinente.

I.4.2. Domaines d'application

L'important volume d'information accessible sur les réseaux sociaux peut être mis à profit dans certains secteurs d'activités, notamment le secteur industriel, sanitaire, sécurité publique Ici nous allons présenter quelques intégrations innovantes dans le domaine de la veille des médias sociaux ainsi que des scénarios types d'applications utilisées pour la communication entre les décideurs et les utilisateurs visant à être au fait des situations et d'en assurer la coordination. Les outils de TAL Permettent également d'interpréter des données en temps réel, ou presque, favorisant Ainsi la prise de décision aux plans stratégiques et opérationnels.

I.4.2.1. Secteur industriel

L'intérêt pour la surveillance de données extraites des médias sociaux est considérable dans le secteur industriel. En effet, ces données sont Susceptibles d'aider en optimisant de manière importante l'efficacité de la veille stratégique. L'intégration de telles données aux systèmes de veille stratégique déjà en Place permet aux entreprises d'atteindre différents objectifs, notamment concernant La stratégie de marque et la notoriété, la gestion des clients actuels et potentiels et L'amélioration du service à la clientèle. Le marketing en ligne, la recommandation de Produits et la gestion de la réputation ne sont que quelques exemples d'applications Concrètes de l'ASMS.

I.4.2.2. Défense et sécurité nationale

Ce secteur s'intéresse beaucoup à l'étude de ce type de sources d'information et de résumés pour comprendre différentes situations, Procéder à l'analyse des sentiments d'un groupe de personnes partageant des intérêts communs et s'assurer d'être à l'affût de menaces potentielles dans leur domaine D'intervention. Certaines méthodes d'extraction d'information à partir du Web 2.0 y Sont présentées pour établir des liens entre différentes données dénotant des entités et Analyser les caractéristiques et le dynamisme des réseaux au sein desquels évoluent Des organismes et des discussions. Dans ce contexte, les agrégats de comportements Sociaux offrent de précieux renseignements en matière de sécurité nationale.

I.4.2.3. Secteur sanitaire

Au fil du temps, les médias sociaux se sont intégrés aux soins de Santé. Ce secteur y recourt pour favoriser l'implication citoyenne et améliorer ses relations avec la clientèle. L'utilisation de Twitter comme plateforme de discussion sur des Sujets tels que les maladies, les traitements, les médicaments ou les recommandations À l'intention des fournisseurs et des bénéficiaires (patients, familles et aidants) illustre Bien la pertinence des médias sociaux dans ce domaine. On a donné au phénomène le Nom de « santé sociale ».

I.4.2.4. Politique

La veille des médias sociaux permet d'assurer le suivi des mentions Faites par différents citoyens d'un pays ainsi que de l'opinion internationale à l'égard D'un parti politique. Le nombre d'abonnés que compte un parti est essentiel au déroulement de sa campagne électorale. L'extraction d'opinions et le suivi des déclarations Publiées sur les forums de discussion permettent à un parti politique de mieux saisir la Teneur de certains événements, lui donnant ainsi l'occasion de s'ajuster pour améliorer Sa position.

I.5. Les textes offensant nourris par les discours de haine

Le discours de haine fait partie de la famille des textes offensants cependant La définition de ce discours n'est ni universellement acceptée, ni les facettes individuelles de la

définition sont pleinement acceptées. Ross et al estiment qu'une définition claire du discours de haine peut aider à l'étude de la détection du discours de haine en facilitant l'annotation des discours de haine, et donc en rendant les annotations plus fiables.

-Nous résumons les principales définitions du discours de haine provenant de diverses sources, ainsi que certains aspects des définitions qui rendent la détection du discours de haine difficile.

1. Encyclopédie de la Constitution américaine: "Le discours de haine est un discours qui attaque une personne ou un groupe sur la base d'attributs tels que la race, la religion, l'origine ethnique, l'origine nationale, le sexe, le handicap, l'orientation sexuelle ou l'identité de genre."
2. Facebook: «Nous définissons le discours de haine comme une attaque directe contre les personnes en fonction de ce que nous appelons des caractéristiques protégées - race, ethnicité, origine nationale, appartenance religieuse, orientation sexuelle, caste, sexe, genre, identité de genre et maladie grave ou handicap. Nous offrons également des protections pour le statut d'immigration. Nous définissons l'attaque comme un discours violent ou déshumanisant, des déclarations d'infériorité ou des appels à l'exclusion ou à la ségrégation. »
3. Twitter: "Conduite haineuse: Vous ne pouvez pas promouvoir la violence contre ou attaquer directement ou menacer d'autres personnes sur la base de la race, de l'ethnie, de l'origine nationale, de l'orientation sexuelle, du sexe, de l'identité de genre, de l'appartenance religieuse, de l'âge, du handicap ou d'une maladie grave." [7]

I.5.1. Les types des textes offensants

Le texte offensant représente l'ensemble des mots et des phrases malveillantes visant une ou plusieurs personnes (groupe, communauté ...) pour leur caractéristiques, parmi ces caractéristiques nous pouvons avoir :

- la couleur de la peau
- l'appartenance ethnique
- le sexe
- la religion
- les origines

- le physique

Ces textes offensants peuvent être vu sous différentes formes nous pouvons citer quelques une :

- discours de haine
- harcèlement
- intimidation
- mensonges et préjugés
- violence verbale

I.5.2. Exemple des textes offensants

Le partage des images est devenu une ressource renouvelable pour les harceleurs qui prennent ces images sans demander l'autorisation et les mettre remplis de textes offensants, beaucoup d'internautes notamment des enfants mineurs sont tombé victimes de ce genre de pratiques. Ce qui a poussé Instagram à lancer une nouvelle fonctionnalité pour lutter contre le harcèlement en ligne. Dès le 17 décembre 2019, les utilisateurs recevront une alerte s'ils s'apprêtent à publier du contenu offensant. Pour détecter les légendes blessantes, Instagram base son analyse sur tous les contenus offensants précédemment signalés par les internautes.



Figure 1 : Les mots offensants et les réseaux sociaux (web)

I.6. Extraction des mots clés et Classification des textes offensants

La classification de texte est un domaine avec un espace de caractéristiques dimensionnelles élevées. L'extraction des mots clés car les fonctionnalités peuvent être extrêmement utiles dans la classification de texte. L'extraction automatique des mots clés est une direction de recherche importante dans l'exploration de texte, le traitement du langage

naturel et la recherche d'informations. L'extraction de mots clés nous permet de représenter des documents texte de manière condensée. La représentation compacte des documents peut être utile dans plusieurs applications, telles que l'indexation automatique, la synthèse automatique, la classification automatique, le « clustering » et le filtrage

La classification de texte est un domaine avec un défi d'espace d'entité dimensionnel élevé. Par conséquent, l'extraction des mots les plus importants / pertinents sur le contenu du document et l'utilisation de ces mots clés comme fonctionnalités peuvent être extrêmement utiles. À cet égard, cette étude examine les performances prédictives de cinq méthodes d'extraction de mots clés statistiques sur les algorithmes de classification et les méthodes d'ensemble pour la classification des documents de texte scientifique (catégorisation). [8]

I.7. Les travaux précédant

La plupart des recherches se concentrent sur les langues les plus parlées, comme l'Anglais et le français ... les autres langues comme la langue arabe souffre encore de la rareté des travaux les langues peu dotées en ressources sont ignorées.

D'après notre recherche nous avons trouvés beaucoup de gens et des chercheurs qui ont travaillé sur notre thème donc nous allons citer certaines de leur affaire :

La taxonomie proposée dans OLID permet de représenter différents types de contenus offensants comme le « discours de haine » et la « cyber intimidation » en fonction du type et de la cible d'un message. Par exemple, de nombreux messages offensants ciblant un groupe sont probablement un « discours de haine » alors que de nombreux messages ciblant un individu sont susceptibles d'être « cyber intimidation ». La taxonomie OLID est devenue populaire en raison de sa simplicité et de sa flexibilité. Elle a été utilisée pour annoter des ensembles de données dans d'autres langues telles que l'arabe [9] et le grec [10], permettant ainsi un apprentissage multilingue et une analyse.

I.7.1. Corpus anglais

Pour ce corpus, ils ont travaillé (Mubarak et al, 2020) [9], (Pitenis et al, 2020) [10]. Avec deux corpus comme suivant:

L'ensemble de données d'identification de la langue offensive (OLID) contient une collection de 14 200 tweets anglais annotés utilisant un modèle d'annotation qui englobe les trois niveaux suivants:

A: Détection des langues offensantes (OFF/NOT)

B: Catégorisation du langage offensant (TIN/UNT)

C: Identification des cibles (IND/GRP /OTH)

OLID : l'ensemble de données officiel participé à des projets d'étudiants dans différentes universités. À notre connaissance, il a jusqu'à présent été utilisé par des étudiants de l'Université d'Arizona (États-Unis), de impériale Collège de Londres (Royaume-Uni) et de l'Université de Leeds (Royaume-Uni). Certains des articles du système étudiant sont disponibles.

SOLID : L'ensemble de données d'identification des langues offensives semi-supervisées contient plus de 9 millions de tweets annotés selon la taxonomie à trois niveaux d'OLID l'ensemble de données anglais officiel [11].

Ils ont entraîné cinq langues (corpus) pour la sous tache(A) et training seulement corpus anglais pour sous tache (B,C).

Le tableau en dessous il va monter les nombres des tweets qui sont entraînés en différents langues.

Langue	Entraîné			Testé		
	OFF	NOT	Total	OFF	NOT	Total
Anglais	1448861	7640279	9089140	1090	2807	3897
Arabe	1589	6411	8000	402	1598	2000
Danois	384	2577	2961	41	288	329
Grec	2486	6257	8743	425	1119	1544
Turque	6131	25625	31756	716	2812	3528

Table 1 : les statistiques pour sous tache(A)

Langue	Entraîné			Testé		
	TIN	UNT	Total	TIN	UNT	Total
Anglaise	149550	39424	188974	850	1072	1922

Table 2 : les statistiques pour Sous tache(B)

Langue	Entraîné				Testé			
	IND	GRP	OTH	Total	IND	GRP	OTH	Total
Anglais	120330	22176	7043	149549	580	190	80	850

Table 3 : les statistiques pour Sous tache(C)

Pour les modèles ils ont utilisé quatre modèles (PMI, fast text, bert, LSTM)

Modèle	Sous tache _A	Sous tache _B	Sous tache _C
Bert	0.816	0.705	0.568
PMI	0.684	0.498	0.461
LSTM	0.681	0.657	0.585
Fast Text	0.662	0.470	0.590

Table 4 : Montre Macro-F1 score pour les modèles du corpus (OLID)

I.7.2. Corpus arabe

Quelque résultat pour le processus du traitement et de test pour la langue arabe

Nombre	Équipe	Score
1	ALAMIHAMZA	0.9017
2	ALT	0.9016
3	Galileo	0.8989
4	KUISAIL	0.8972
5	AMR-KELEG	0.8958

Table 5 : Résultats pour arabe sous tache (A) macro-average F1 Score

I.7.3. Corpus Danois

Nombre	Équipe	Score
1	LT@HELSSINKI	0.8119
2	Galileo	0.8021
3	NLPDOVE	0.7923
4	FBK-DH	0.7766
5	KS@LTH	0.7750

Table 6 : Résultats pour le Danois sous tache (A) macro-Average F1 Score

I.8. Conclusion

Dans ce premiers chapitre nous avons présenté une vue générale sur notre travail qui consiste à appliquer les techniques et les méthodes du TAL que nous avons étudié. Pour justement détecter les textes offensants avec leurs diverses formes tout en rappelant les aspects et la définition des bases de chacun de ces concepts.

Dans le chapitre qui suit, nous discutons plus en détail sur la classification et ses différentes techniques de l'identification du texte notamment celui qui est offensants. Après avoir consulté les travaux précédents nous avons remarqué que les données pour la langue Anglaise sont beaucoup plus grandes que les autres langues Par la suite nous allons utiliser les résultats des tests de ces travaux précédents pour les comparer avec les miens nous avons aussi pris quelques idées sur les algorithmes avec lesquelles ils ont travaillé.

Chapitre II

La classification, ses différents aspects et complexités

II.1.Introduction

Dans ce chapitre nous allons parler sur les problèmes et les difficultés de la classification automatique et l'identification de texte avec ces différents appellations : classification, clustering, ou catégorisation (la méthode avec laquelle nous allons opter pour notre travail) , nous allons également spécifier les différentes caractéristiques des langues sur lesquelles nous avons travaillé notamment la langue arabe avec ces différentes règles et variations ainsi que l'impact de cette dernière sur notre traitement.

II.2. Définition de la classification

La classification automatique de documents est l'un des fameux problèmes en informatique, il consiste à attribuer un document à une ou plusieurs catégories ou classes. Le problème se diffère et se multiplie en fonction des types des documents en question, en effet la classification de textes est différent de celle de documents images, vidéo ou encore son. Nous pouvons aussi envisager des classifications selon des règles propre aux documents par exemple l'auteur, la date de parution... Dans le cadre de ce projet et dans la suite de rapport nous nous concentrons sur la classification de documents de type texte selon leur contenu. La classification de textes est une tâche générique qui consiste à mettre en groupe des documents qui se ressemblent selon des critères bien définis à savoir les critères observables tels que le type du document, le genre, la nature, etc... Assigner à une ou plusieurs catégories, parmi une liste prédéfinie, ou non à un document. La classification de textes est définie comme une opération qui identifie des classes d'équivalence entre des segments de textes en tenant compte de leur contenu informationnel (mots, n-gram, etc.).

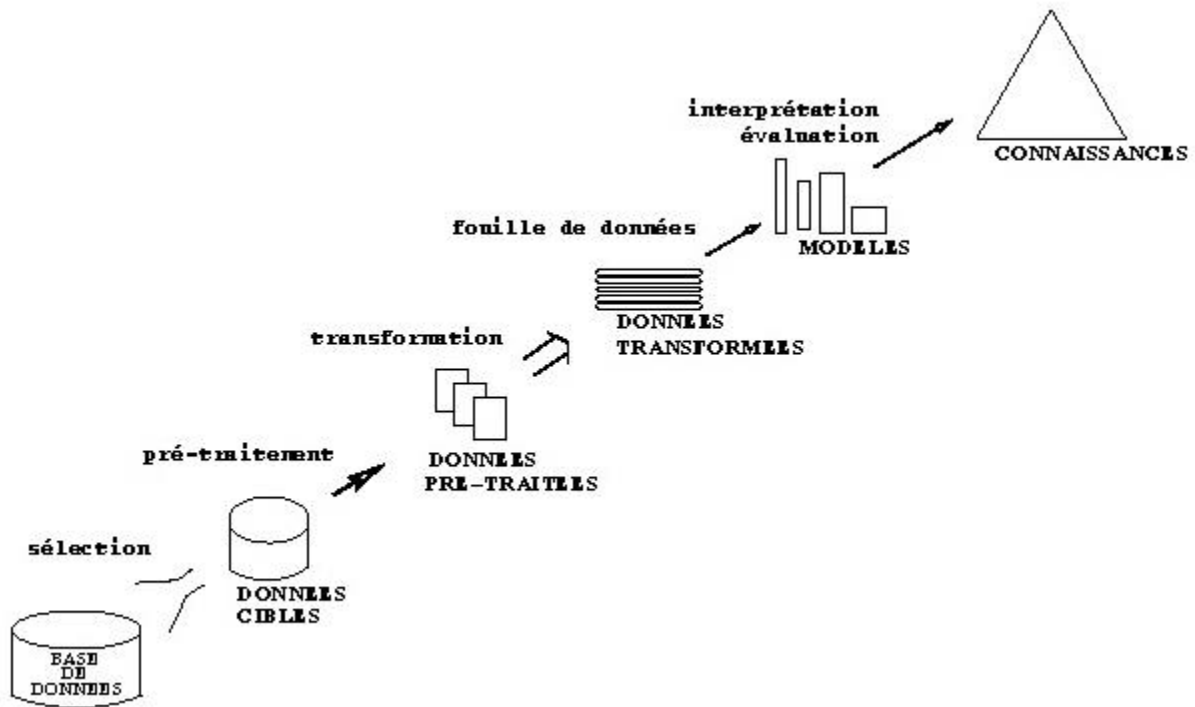


Figure 2 Extraction des infos et des connaissances (carin).

II.2.1 l'explication formelle

Formellement, la catégorisation de texte consiste à associer une valeur booléenne à chaque paire $(d_j, c_i) \in D \times C$, où D est l'ensemble des textes et C est l'ensemble des catégories selon.

Que $d_j \in c_i$, ou non. Le but de la catégorisation de texte est de construire une procédure (modèle, classificateur) $\Phi : D \times C \rightarrow B$ qui associe une ou plusieurs étiquettes (catégories) à un document d_j avec la fonction $F : D \rightarrow C$, la vraie fonction qui retourne pour chaque vecteur d_j une valeur c_i [12].

II.2.2. l'automatisation de la classification

Nous assistons ces dernier temps à un accroissement de la quantité d'information textuelle disponible et accessible d'une manière assez particulière. D'après les derniers chiffres, plus de 200 millions de serveurs hôtes sur Internet et plus de 3 milliards de pages, la taille des corpus tests utilisés est passée en quelques années de quelques Méga-octets à plusieurs Giga-octets. Le nombre de textes à classificateur est énorme, il est très dur de connaître la période pour laquelle a besoin un spécialiste (expert) pour attribuer un texte à une catégorie ? En pratique, il s'agit d'une question difficile à répondre. Assurément, plusieurs

variables influencent le phénomène et les lignes qui suivent porteront sur certaines d'entre elles. Certainement une grande partie du temps consommé pour classer un document est employé dans sa lecture, puis éventuellement à sa relecture. On peut aussi imaginer que la longueur des textes à classer est assez déterminante du temps qui va être requis pour cette opération, et sans doute, d'une personne à une autre, la vitesse de lecture varie. Une fois cette étape achevée, il faut trancher à quelle(s) catégorie(s) ce texte appartient. Au temps de réflexion exigé s'ajoute, certainement, le temps de se référer à la description des classes et éventuellement de consulter d'autres textes préalablement associés à certaines classes, pour valider la décision. D'autres facteurs interviennent également comme, comme par exemple le nombre de classes qui peut faire la différence : plus il y a de classes différentes, autrement dit plus il y a d'étiquettes possibles pour un texte donné, plus il est difficile de faire un choix parmi celles-ci. Aussi, plus la sémantique des catégories est précise, fine, détaillée, plus il faut faire attention avant d'associer un document. À cet égard, classer des documents appartenant soit à la catégorie «informatique» soit à la catégorie «mathématiques» est vraisemblablement plus aisée que celle de classer des documents appartenant à l'une ou l'autre des catégories «Intelligence artificielle», «Génie logiciel» et «Système d'information». En résultat, nous illustrons dans les trois points suivants les problèmes et les difficultés cruciales qui font que la méthode classique manuelle de classification des documents textuels n'est plus intéressante :

- L'opération manuelle de cette tâche par un spécialiste du domaine est extrêmement coûteuse en termes de temps et personnel car il s'agit de lire attentivement chaque texte, au vu de la quantité phénoménale de textes aujourd'hui accessibles (par le biais du réseau Internet en particulier)
- Les traitements manuels sont peu flexibles et leur généralisation à d'autres domaines est pratiquement impossible; c'est pour cela qu'on cherche à installer des méthodes automatiques
- Cette opération peut être perçue comme subjective puisque basée sur l'interprétation du document, deux experts peuvent classer différemment un même document, ou encore un même expert peut classer différemment un même document soumis à deux instants différents [14].

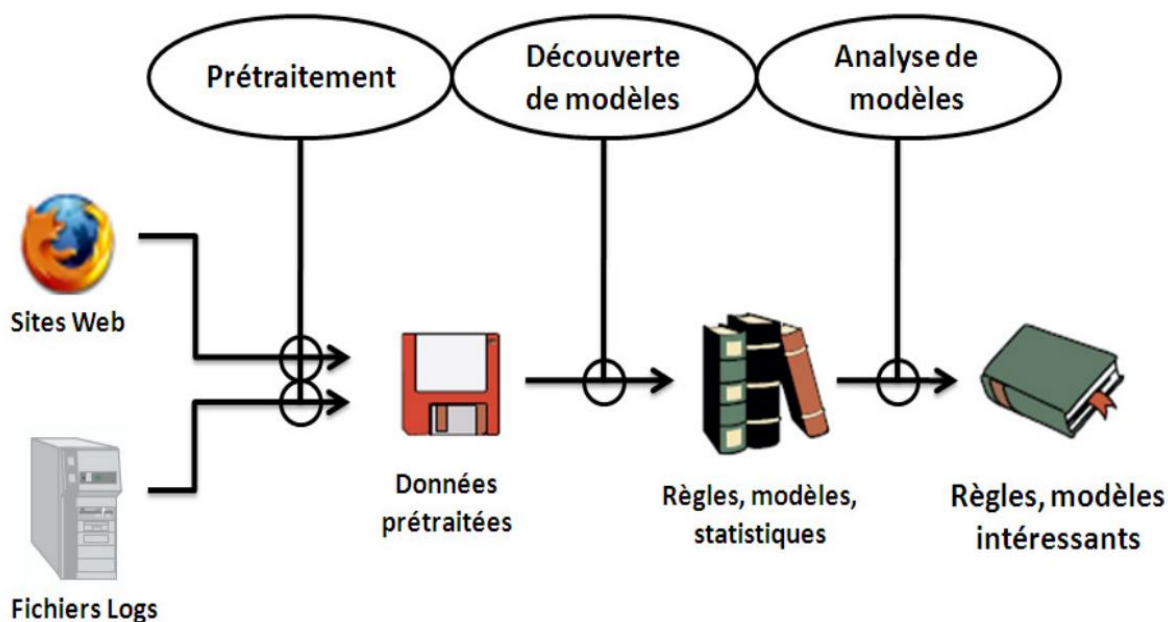


Figure 3 Extraction Automatique (web).

II.2.3.les méthodes de la classification

L'objectif de la CT est de classer de façon automatique les documents dans des catégories qui ont été définies soit préalablement par un expert, il s'agit alors de classification supervisée ou catégorisation, soit de façon automatique, il s'agit alors de classification non supervisée ou encore «clustering».

II.2.3.1. Apprentissage non supervisé «Clustering»

L'apprentissage non supervisé consiste à apprendre à classer sans supervision. Au début de processus nous ne disposons ni de la définition des classes, ni de leurs nombres. C'est l'algorithme de classification qui va déterminer ces informations. Nous ne disposons pas non plus de données en entrée qui sont déjà classées, c'est aussi à l'algorithme de découvrir par lui-même la structure plus ou moins cachée des données et de former des groupes d'individus dont les caractéristiques sont communes. [16] La classification non supervisée consiste à trouver de manière automatique une organisation cohérente à un groupe de documents homogènes pour construire des regroupements cohérents (des classes ou clusters), elle correspond en statistiques au « clustering », qui est également le terme utilisé en recherche d'informations. Le «clustering» consiste donc, à diviser les objets (dans notre cas des textes) en groupes sans connaître à priori leurs classes d'appartenance. Les techniques pour réaliser de tels regroupements constituent un domaine d'étude très riche, qui a donné lieu

à de multiples propositions dont le recensement n'est pas l'objet de ce document. L'apprentissage non supervisé est utilisé dans plusieurs domaines tels que :

- Médecine : Découverte de classes de patients présentant des caractéristiques physiologiques communes.
- Le traitement de la parole : construction de système de reconnaissance de la voie humaine.
- Archéologie : regroupement des objets selon leurs époques.
- Traitement d'images - Classification de documents.

Dans la littérature il existe plusieurs types d'algorithmes d'apprentissage non supervisé tels que les algorithmes de partitionnements et les algorithmes de classification hiérarchique :

- Le partitionnement: consiste au regroupement des données suivant leur degré de similarité. L'algorithme le plus célèbre appartenant à cette classe est K-means : c'est un algorithme qui permet de partitionner un ensemble de données automatiquement en K clusters. Il consiste tout d'abord à choisir k points qui représentent les centres des groupes à créer, puis à affecter les autres points aux centres les plus proches. Cette affectation est faite par le calcul de distance entre les points. Plusieurs distances peuvent être définies telles que la distance euclidienne ou la distance de Manhattan. Par la suite nous procédons à une étape de raffinement des groupes de façon itérative, le raffinement se fait par le recalcul des centres des groupes après chaque itération et par une réaffectation des points aux groupes. L'algorithme s'arrête quand aucun point ne bouge. [16]
- La classification hiérarchique : il existe deux types de classification hiérarchique : Ascendante et descendante. La classification ascendante consiste à utiliser une matrice de similarité afin de partir d'une répartition fine vers un groupe unique. Donc, il s'agit de fusionner les groupes jusqu'à ce qu'on obtient un seul groupe englobant tous les autres. Cette classification peut être représentée par un arbre hiérarchique ou dendrogramme. La classification descendante se présente comme l'inverse de la classification ascendante. Donc il s'agit de décomposer un cluster unique en sous-groupes jusqu'à l'obtention des singletons. [15]

II.2.3.2. Apprentissage supervisé «Catégorisation»

Contrairement à l'apprentissage non supervisé, nous commençons ici par un ensemble de classes connues et définies à l'avance. Nous disposons aussi d'une sélection initiale de données dont la classification est connue. Ces données sont supposées indépendantes et identiquement distribuées. Elles nous servent pour l'apprentissage de l'algorithme. La classification se fait par l'algorithme selon le modèle qu'il a appris. [15] La catégorisation de

textes correspond à la procédure d'affectation d'une ou de plusieurs catégories ou classes prédéfinies à un texte. Elle correspond à la classification supervisée pour l'apprentissage automatique et à la discrimination en statistiques alors que la recherche d'informations utilise des termes plus proches de l'application concernée : filtrage ou routage. Cette problématique utilise largement des méthodes issues de l'apprentissage automatique et beaucoup d'algorithmes d'apprentissage supervisé lui ont été appliqués (Naïve bayes, K plus proches voisins, arbres de décision, machines à vecteurs support, réseaux de neurones, etc...).

II.2.3.3. Avantages et inconvénients

Parmi les avantages et inconvénients liés aux deux approches, on peut citer :

- Les groupes ou clusters obtenus par la technique supervisée est de meilleure qualité et plus précise que la technique non-supervisée.
- Dans la technique supervisée, on sait ce qui est attendu favorisant de meilleurs résultats par rapport au non supervisée.
- Un avantage des techniques non supervisées, est qu'elles accomplissent la tâche de similarité sans avoir besoin des données expertisées.
- Un inconvénient des approches supervisées, repose sur le fait qu'il peut être difficile de se procurer des données expertisées.
- L'inconvénient majeur des approches non supervisées qu'elle demande dans l'étape d'évaluation des résultats l'intervention d'un expert.

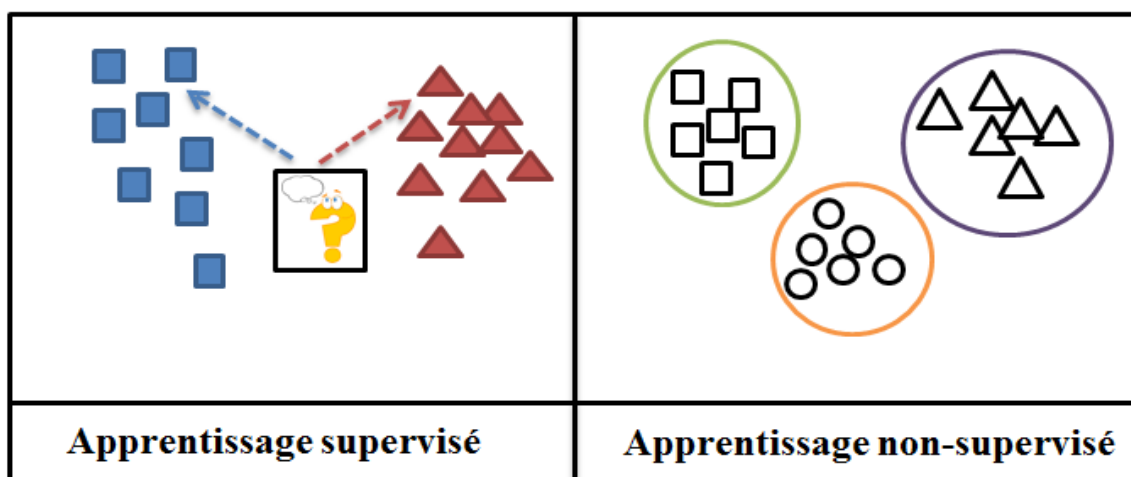


Figure 4 : Apprentissage supervisé/Non supervisé.

II.3. Problèmes rencontrés dans la catégorisation de textes

Beaucoup de difficultés peuvent s'opposer au processus de catégorisation de textes. Des problèmes connus dans la discipline liés à l'apprentissage automatique supervisé comme la subjectivité de la décision prise par les experts, le sur-apprentissage, etc.. mais aussi des problèmes particuliers liés à la nature des données traitées à savoir des données textuelles comme la polysémie, la redondance, Les variations morphologiques ou même L'homographie, etc.. Nous allons signaler les huit principales Dans ce qui suit :

II.3.1. Sur-apprentissage

Le sur-apprentissage s'explique par le fait que le modèle de prédiction n'arrive pas à bien classer les nouveaux textes, pourtant il l'a bien fait dans la phase d'apprentissage en classant correctement les textes de la base d'apprentissage.

Pour limiter le sur-apprentissage, on doit sélectionner des termes pour réduire la dimensionnalité. D'après les expériences antérieures, le nombre de termes doit être limité par rapport au nombre de textes de la base d'apprentissage. Quelques auteurs recommandent d'utiliser au moins 50 à 100 fois plus de textes que de termes. En général le nombre de textes d'apprentissage est limité, c'est pour cela on cherche à agir sur le nombre des termes utilisés en les diminuant, pour éviter ce sur-apprentissage. Sans bien sûr pénaliser le système en supprimant des termes pertinents.[17]

II.3.2. L'homographie

Deux mots sont dits homographes si 'ils s'écrivent de la même façon sans forcément avoir la même prononciation. L'homographie est une sorte d'ambiguïté supplémentaire. (Ex : avocat en tant que fruit et avocat en tant que juriste).

II.3.3. Polysémie (Ambiguïté)

Un mot possède, dans différents cas, plus d'un sens et plusieurs définitions lui sont associées. Par conséquent, à cause de la polysémie, les mots seuls sont parfois de mauvais descripteurs ; exemple le mot livre peut désigner une unité monétaire, ou un bouquin. 2.10.4. Les mots composés Le non prise en charge des mots composés comme : comme Arc-en-ciel, peut-être, sauve-qui-peut, etc. Dont le nombre est très important dans toutes les langues, et traiter le mot Arc-en-ciel par exemple en étant 3 termes séparés réduit considérablement la performance d'un système de classification néanmoins l'utilisation de la technique des n-grammes pour le codage des textes atténue considérablement ce problème des mots composés.

II.3.4. La graphie

Un terme peut comporter des fautes d'orthographe ou de frappe comme il peut s'écrire de plusieurs manières ou s'écrire avec une majuscule. Ce qui va peser sur la qualité des résultats. Parce que si un terme est orthographié de deux manières dans le même document exemple : ن'وورد , ca peut être l'abréviation d'un nom irlandais (nill woord), mais ca peut être aussi (N'Word) qui est un tabou en Amérique car il représente l'un des slogans anti noir les plus connus dans le monde. la simple recherche de ce terme avec une seule forme graphique néglige la présence du même terme sous d'autres graphies, ce qui va influencer les résultats puisque les différentes graphies vont être traitées séparément. Néanmoins du point de vue pratique, le fait qu'un terme inconnu est proche d'un autre terme prouve qu'il a été mal orthographié.

II.3.5. Redondance(Synonymie)

La redondance et la synonymie permettent d'exprimer le même concept par des expressions différentes, plusieurs façons d'exprimer la même chose. Cette difficulté est liée à la nature des documents traités exprimés en langage naturel contrairement aux données numériques. LE FEVRE illustre cette difficulté dans l'exemple du chat et l'oiseau : mon chat mange un oiseau, mon gros matou croque un piaf et mon félin préféré dévore une petite bête à plumes.[17] La même idée est représentée de trois manières différentes, différents termes sont utilisés d'une expression à une autre mais en fin compte c'est bien le malheureux oiseau qui est dévoré par ce chat. Lors d'une représentation vectorielle d'un document, ces termes sont représentés séparément, et les occurrences du concept sont dispersées. Il est alors important de rassembler ces terme sen un groupe sémantique commun.

II.3.6. Présence-Absence de termes

La présence d'un mot dans le texte indique un propos que l'auteur a voulu exprimer, on a donc une relation d'implication entre le mot et le concept associé, quoique on sait très bien qu'il y a plusieurs façons d'exprimer les mêmes choses, dès lors l'absence d'un mot n'implique pas obligatoirement que le concept qui lui est associé est absent du document. Cette réflexion pointue nous amène à être attentifs quant à l'utilisation des techniques d'apprentissage se basant sur l'exclusion d'un mot particulier

II.3.7. Subjectivité de la décision

Parmi les problèmes classiques usuels dans le domaine de l'apprentissage supervisé c'est la subjectivité de la décision prise par les experts qui décident de la classe à laquelle le texte va être attribué. Certainement après la lecture du texte à classer, l'expert va trancher à

quelle(s) catégorie(s) ce texte appartient en se basant sur le contenu sémantique et le contexte du texte et même en consultant d'autres textes préalablement associés à certaines classes, pour valider la décision prise qui ne peut être que subjective. Les experts humains ne lisent pas de la même manière ! Ne réfléchissent pas de la même manière ! Donc ne classent pas de la même manière ! Ainsi un même document peut être classé différemment par deux experts, ou encore un même document peut être classé différemment par le même expert, soumis à deux instants différents.[17]

II.4. Particularité de la langue arabe

Dans les travaux précédents, nous avons remarqué une certaine efficacité dans les langues comme l'anglais, grâce à la multiplication des essais et des tentatives et les moyens qui assurent la continuité de ces travaux dans ce domaine, quant à la langue arabe classique il n'existe qu'un peu d'efforts, Et qui n'est pas du tout suffisant. Cependant, dans notre travail nous allons nous intéresser à classer les textes selon des critères un peu plus avancés offensants ou pas, direct ou indirect, vers un individu ou vers un groupe ou autres (ni l'un ni l'autre).

La langue arabe appartient à la famille des langues sémitiques, et plus précisément au rameau méridional de ces langues, avec un nombre de locuteurs estimé entre 315 millions 1 et 375 millions de personnes au sein du monde arabe et de la diaspora arabe², elle est utilisée comme vecteur de transmission religieux pour tous les croyants musulmans au nombre de 1 milliard et demi à travers les cinq continents du globe. Le fait que la langue arabe est la langue du coran elle s'est étendue au-delà du golfe arabo-persique, atteignant l'Afrique du nord et l'Asie mineure. De plus, l'expansion territoriale de l'empire musulman a fait de l'arabe une langue de culture et de sciences. Par ailleurs, la diversité des populations arabes et de leurs cultures ont fait émerger différentes variantes de l'arabe allant de l'arabe classique utilisé dans le coran, à l'arabe standard moderne.

II.4.1. Alphabet

L'alphabet arabe comporte vingt-huit lettres (si l'on exclut la hamza, qui se comporte soit comme une lettre à part entière soit comme un diacritique). De nombreuses lettres sont similaires, cela résulte directement du fait que l'écriture est cursive : les formes possibles des lettres s'en sont trouvées diminuées. Pour distinguer les différents sons notés par une même lettre, on utilise des points placés sur ou sous la lettre. Il ne dispose, en réalité, que d'une quinzaine de caractères pour les noter. Ces lettres peuvent être rangées selon l'ordre traditionnel des (ع، ح، ط، ي، ك، ل، م، ن، س، ا، ب، ج، د، هـ، و، ز، ح، ط، ي، ك، ل، م، ن، س، ع) sémitiques alphabets.

، ف ، ص ، ق ، ر ، ش ، ت ، ث ، غ ، ظ ، ض ، ذ ، خ ، د ، ز ، س ، ش ، ص ، ض ، ح ، ج ، ت ، ث ، ج ، ح ، خ ، د ، ذ ، ر ، ز ، س ، ش ، ص ، ض ، ط ، ظ ، ع ، غ ، ف ، ق ، ك ، ل ، م ، ن ، هـ ، و ، ي .

Mais dès les toutes premières époques est apparu un ordre mnémotique, dans lequel des regroupements rapprochent les lettres dont les formes sont semblables. Voici l'ordre traditionnel de l'alphabet arabe, présenté d'une manière qui en fait apparaître les regroupements entre (أ، ب، ت، ث، ج، ح، خ، د، ذ، ر، ز، س، ش، ص، ض، ط، ظ، ع، غ، ف، ق، ك، ل، م، ن، هـ، و، ي) semblable graphisme de lettres.

II.4.2. la classification dans la langue arabe

Le traitement de la langue arabe représente un enjeu très difficile pour les chercheurs du domaine non seulement par rapport à la pauvreté des travaux et les ressources mais aussi par rapport ses caractéristique qui sont extrêmement spéciale et propre à cette langue dans ce qui suit nous allons citer quelques-unes caractéristique avec une petite explication pour chacune de ces caractéristiques.

II.4.3. Le partitionnement

Connu aussi sous le terme « segmentation » c'est une étape importante en prétraitement , le partitionnement consiste à diviser un texte en partition (petite unité) de plus grand au plus petit (chaine de caractère en mots ou un mot en caractères ou un texte en paragraphes) commencer une analyse d'un texte sans le segmenter en phrases(ou mot ou caractères) mène à des résultat moins bons , de même avoir une mauvais partitionnement conduit à accumuler les erreurs du traitement automatique du texte. Pour la langue arabe il y a peu de travaux sur la segmentation de texte et li n'existe pas des segmentations fonctionnelles et spécifiques à la langue arabe [60] Le partitionnement est une source d'ambiguïtés, vu que d'une part la ponctuation est rarement utilisée dans les textes arabes et d'autre part cette ponctuation, lorsqu'elle existe, n'est pas toujours déterminante pour guider la segmentation. De plus, certains mots outils peuvent marquer le début d'une nouvelle phrase, ce qui nécessite des analyses de surface afin de pouvoir segmenter le texte. Exemple : ولد هذا العالم والباحث في مصر : Entrée Dans cette phrase, la particule 'و' [w] joue le rôle de séparateur entre propositions et segmente l'énoncé en deux propositions, par contre dans la phrase suivante : Sortie : ولد العالم والباحث في مصر La même particule ' و ' [w] ne joue pas le rôle de séparateur entre propositions mais plutôt celui d'une conjonction de coordination entre les mots العالم[AIEAlm] (Savant) et الباحث[AlbAHv] (chercheur) et donc ne segmente pas la phrase. [61] ..

II.4.4. Les voyelles

Les voyelles sont des signes diacritiques placés au-dessus ou en-dessous des lettres, La plupart des documents arabes sont non voyelles. En effet, les voyelles ne sont utilisées que dans certains ouvrages scolaires pour débutants et dans le Coran, Un texte arabe non voyelle est fortement ambigu. [61], ce qui peut poser de problèmes lors du traitement.

II.4.5. L'agglutination

L'ambiguïté c'est le facteur qui rend le plus le traitement automatique de la langue plus difficile car il touche à la fois la mémoire du stockage mais aussi la puissance du processeur. Nous pouvons voir l'ambiguïté sous différentes formes et selon différents niveaux de traitement que ce soient : lexical morphologique, syntaxique et même sémantique [62] Exemple : Par exemple le mot " علم " peut représenter un nom «ilm» علم (science) ou un verbe «aalima» (il a appris), ça peut même avoir plus de deux ambiguïtés.

II.4.6. Ambiguïté

L'ambiguïté c'est le facteur qui rend le plus le traitement automatique de la langue plus difficile car il touche à la fois la mémoire du stockage mais aussi la puissance du processeur. Nous pouvons voir l'ambiguïté sous différentes formes et selon différents niveaux de traitement que ce soient : lexical morphologique, syntaxique et même sémantique [62] Exemple : Par exemple le mot " علم " peut représenter un nom «ilm» علم (science) ou un verbe «aalima» (il a été informé), ça peut même avoir plus de deux ambiguïté .

II.5. Conclusion

L'identification du texte a toujours été un enjeu en même temps difficile et intéressant pour les amoureux de l'intelligence artificielle et le traitement automatique de langage en particulier, le langage naturel humain avec ces différentes variation linguistiques et morphologiques a besoin non seulement d'une énorme masse de données structurés (pour faire un bon apprentissage) mais aussi des algorithmes pertinents et performants qui peuvent à partir de cet apprentissage tirer le meilleur des résultats . dans notre cas nous avons travaillé avec 10 algorithmes 8 pour l'apprentissage automatique (BNB, MNB, DT, KNN, RF, LR, GB ? LSVC) et 2 pour l'apprentissage profond (simple RNN, LSTM) , nous en avons parlé en détailles dans les prochaines chapitres .

Chapitre III

Conception et Architecture
du système d'identifications
des textes offensants

III.1. Introduction

La définition et la modélisation d'une architecture dédiée aux activités de L'analyse des « big data », comme celles produites par les réseaux sociaux comme Twitter, est actuellement encore dans un stade précoce de son développement et de sa consolidation. Contrairement à l'entrepôt de données traditionnel où les systèmes de business intelligence, dont l'architecture est conçue pour les données structurées, les systèmes dédiés au « big data » ont fonctionné plutôt avec des données semi-structurées, ou dites "brutes", c'est-à-dire sans structure Particulière.

De nos jours, une énorme quantité de données, produites quotidiennement par les réseaux sociaux, peut être traitée et analysé à des fins différentes. Ces données sont dotées de plusieurs fonctionnalités, parmi lesquelles (la dimension, les particularités, la source...).

Au moment où la nécessité d'obtenir les informations et la manière dont ces informations doivent être traitée a changé. Jusqu'à récemment, on pensait que les données devraient être les premières Traitées et mises à disposition ultérieurement, quel que soit l'aspect temporel.

L'architecture du traitement pourrait conduire à une précision faible (si nous utilisons les sous taches séparément). Une solution possible consiste à fusionner entre «Dataset » et les tweets en une seule architecture (tweet-prédiction).

III.1.1. Problématique

Notre problématique se base sur comment :

- Comment identifier les textes offensants ???
- Comment les classer automatiquement ???
- Selon quels critères ???
- Les Algorithmes et les représentations choisis ???

Une solution possible à ce problème est l'architecture (tweet-prédiction) □

III.1.2. L'objectif

L'objectif c'est de réaliser un bon système qui va nous permettre de prédire les cibles des tweets et les classer selon des différentes catégories.

Savoir quelles sont les meilleures représentations vectorielles parmi les quatre représentations que nous allons présenter après, et quel est le meilleur modèle parmi les modèles d'apprentissage automatique ou bien profond qui va nous aider à la prédiction.

III.2. Conception et l'architecture Générale

L'architecture du système est une réponse aux difficultés conceptuelles et pratiques de la description et de la conception de notre problématique.

La figure 5 représente notre architecture de manière générale qui inclut un corpus avec ses trois niveaux et aussi la condition qui permet le passage d'un niveau à un autre.

Au niveau A: le tweet soit il est de type offensif soit non.

Au niveau B: le tweet soit il est classé dans la catégorie target ou bien untarget.

Au niveau C: le tweet soit il est destiné à une personne spécifique (individual), groupe ou autre chose (other).

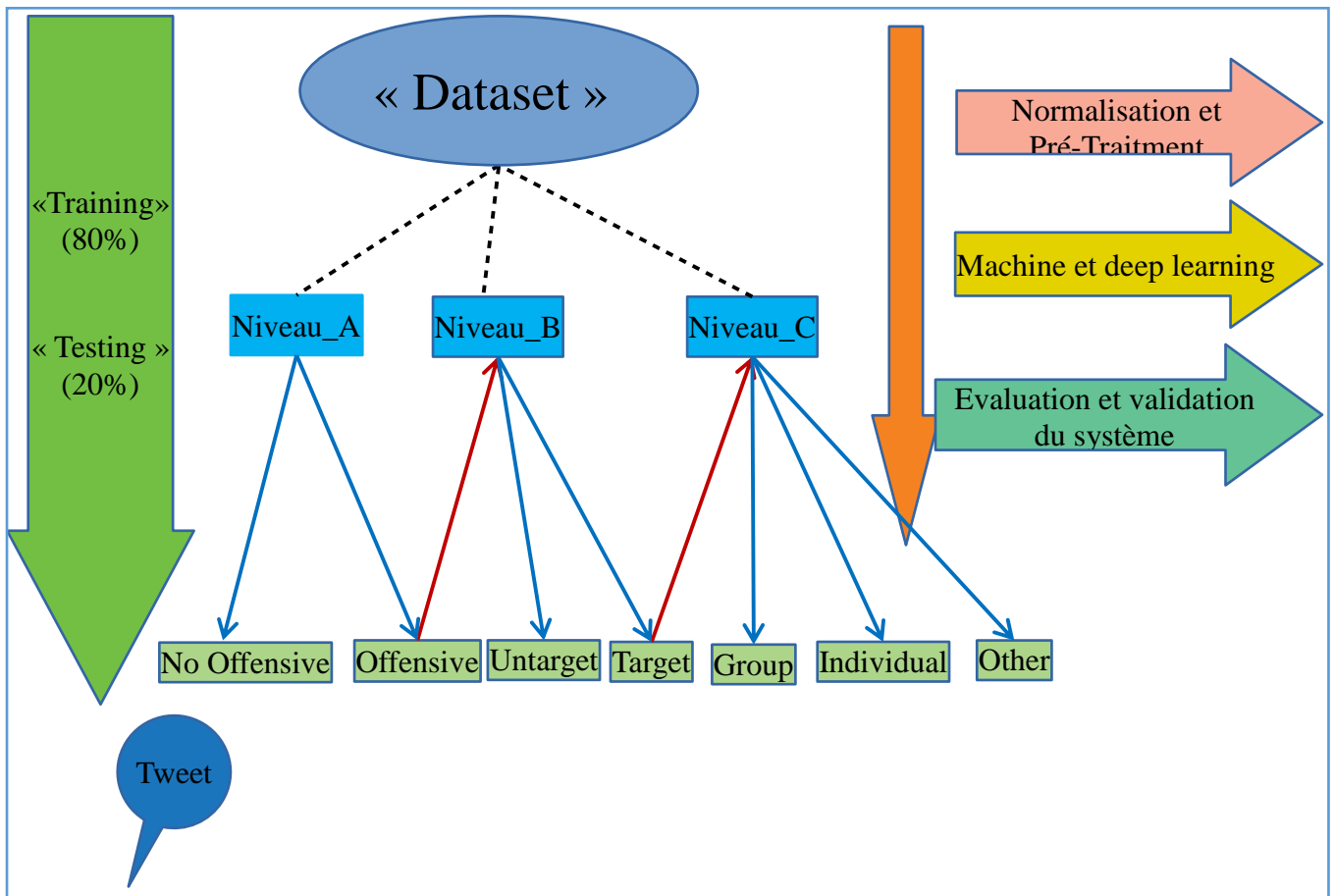


Figure 5 : Schéma du système général «tweet prédiction ».

Architecture plus détaillée pour le sous-tâche _A

Sous Tache_A

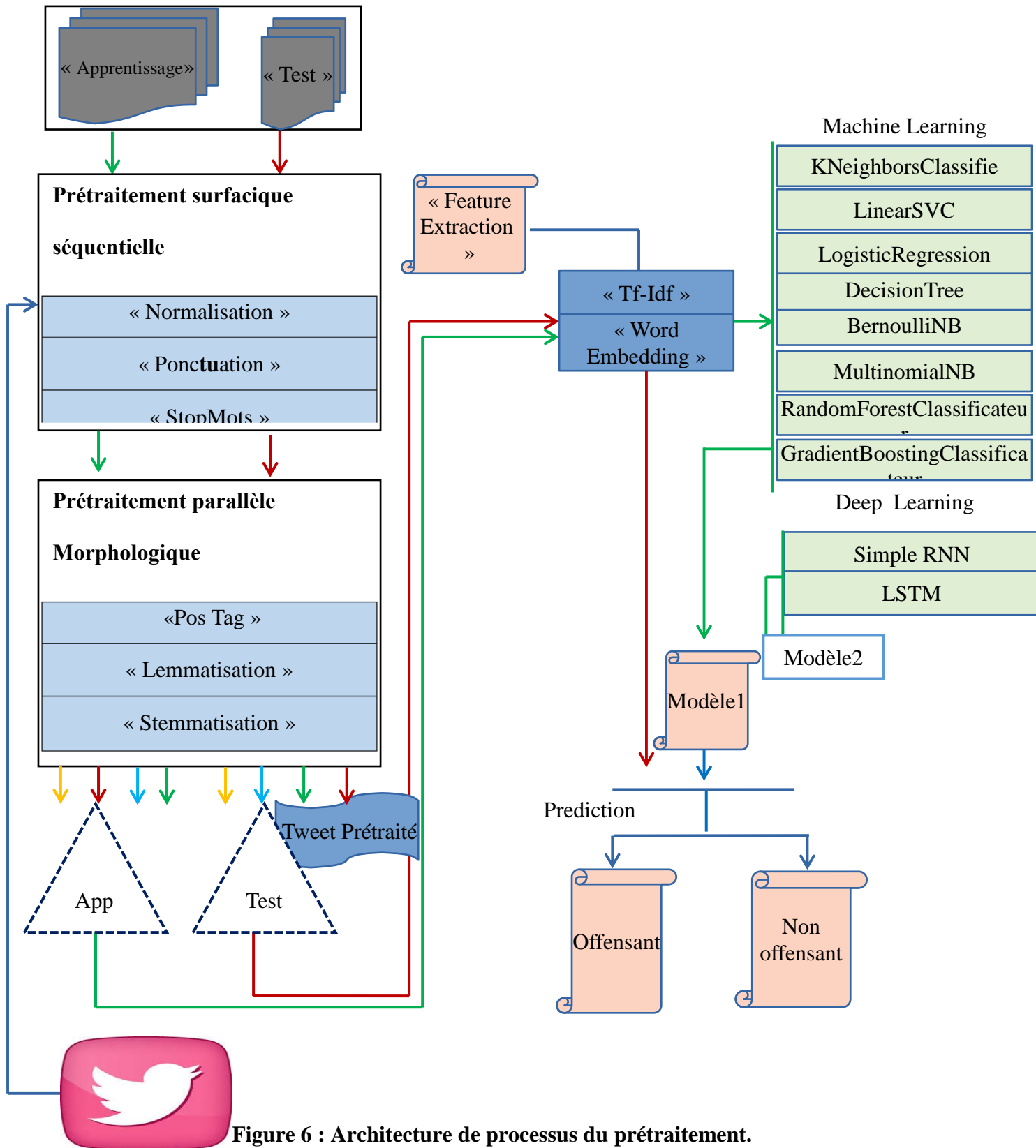


Figure 6 : Architecture de processus du prétraitement.

III.2.1 « Dataset » (corpus) :

C'est un ensemble de données (des tweets offensives ou non offensives) avec ces types pour la détection d'un langage offensant en utilisant un schéma d'étiquetage hiérarchique<tweet-prédiction>.

Dans cette section, nous décrivons nos ensembles de données pour les deux langues : anglais, arabe, Tous les langages suivent le schéma d'annotation <tweet-prédiction> et tout le jeu de données a été traité en utilisant les mêmes méthodes, par exemple, toutes les mentions d'utilisateurs ont été remplacées par @USER pour l'anonymisation. Cette stratégie est conforme aux meilleures pratiques actuelles en matière de collecte de données linguistiques abusives ,Toutes les deux langues contiennent des données pour Sous-tâche A, et seul l'anglais contient des données pour les tâches B et C. La distribution des données pour toutes les langues pour la sous-tâche A est indiquée dans le tableau 7.

Niveau A	« Training »			« Testing »		
Langue	OFF	NOT	TOTAL	OFF	NOT	TOTAL
Anglaise	3485	7107	10592	915	1733	2648
Arabe	675	3325	4000			

Table 7 : Répartition des combinaisons d'étiquettes pour la tâche A

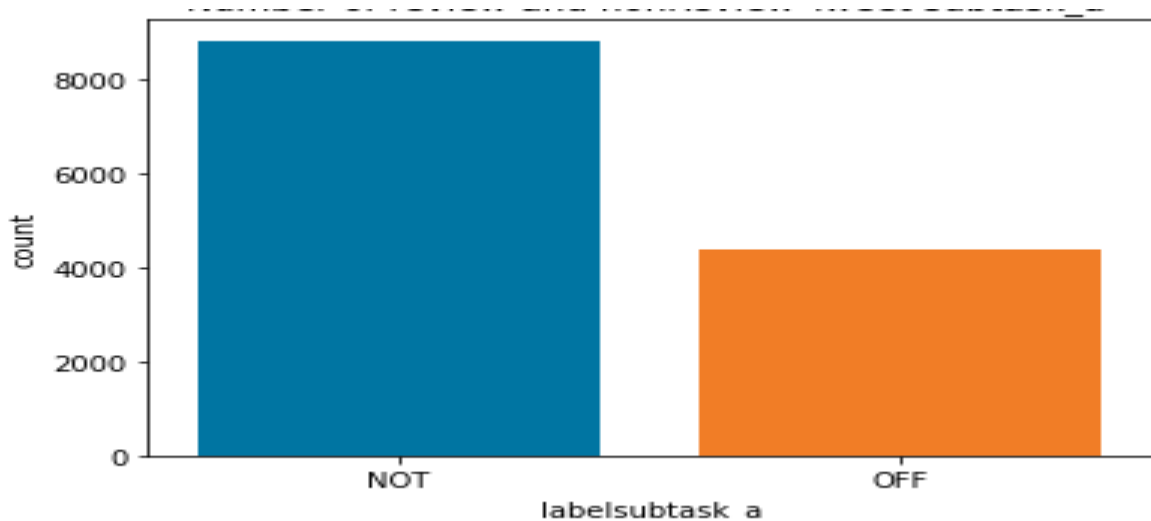


Figure 7 graphe montre le nombre du tweets sous tache A

Niveau B	Training				Testing			
Langue	TIN	UNT	NULL	TOTAL	TIN	UNT	NULL	TOTAL
Anglais	3084	401	7107	10592	792	128	1728	2648

Table 8 : Répartition des combinaisons d'étiquettes pour la tâche B

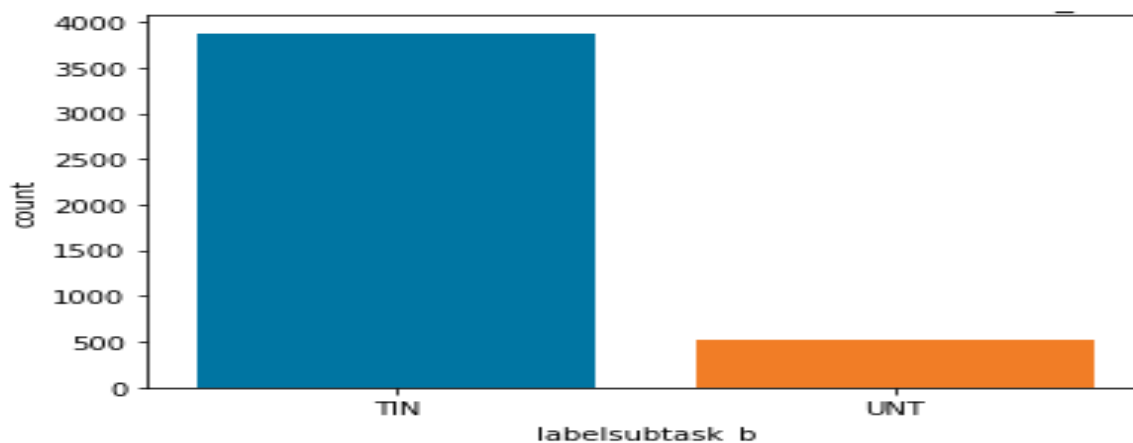


Figure 8 : graphe montre le nombre des tweets sous tache_B

NiveauC	Training					Testing				
Langue	IND	GRP	OTH	NULL	TOTAL	IND	GRP	OTH	NULL	TOTAL
Anglais	1903	871	310	7508	10592	504	202	85	1847	2648

Table 9 : Répartition des combinaisons d'étiquettes pour la tâche C

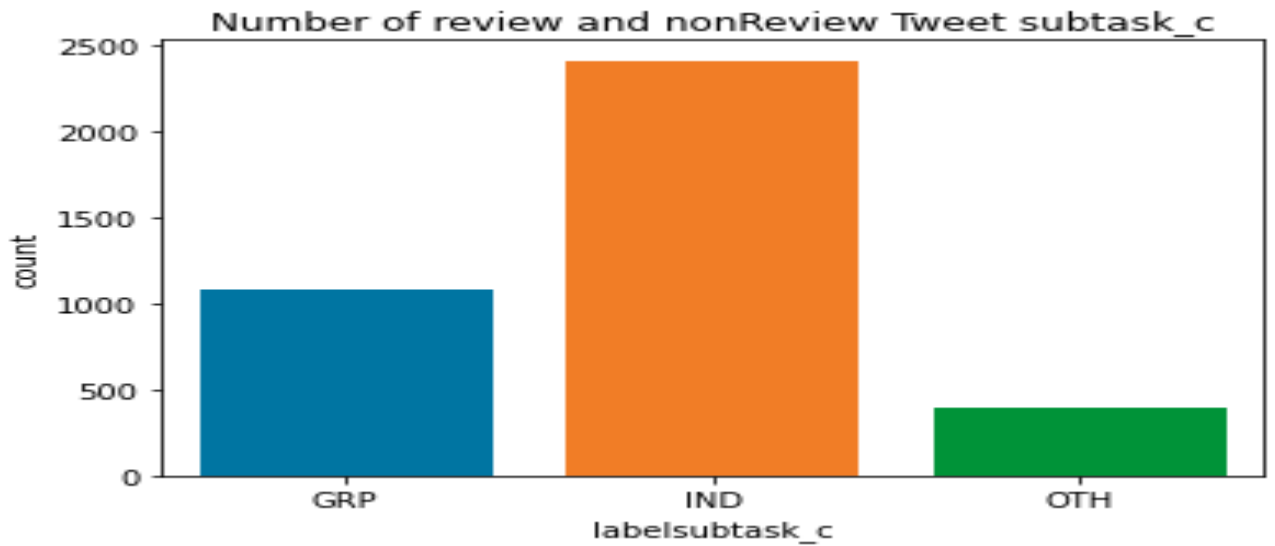


Figure 9 : graphe montre le nombre des tweets sous tache_c

Quelque exemple qui sont représentés des types et les catégories pour les deux langues arabes et anglaises.

Langue	Tweet	Niveau A	Nive au B	Nive au C
Anglais	-@USER She should ask a few native Americans what their take on this is.	OFF	—	—
Anglais	- @USER Go home you're drunk!!! @USER #MAGA #Trump2020 □□□□ URL	OFF	TIN	IND
Anglais	@USER was literally just talking about this lol all mass shootings like that have been set ups. it's propaganda used to divide us on major issues like gun control and terrorism	OFF	TIN	GRP
Anglais	@USER @USER @USER It's not my fault you support gun control	NOT	—	—
Anglais	ANTI-ANTIFA IS BALLS	OFF	UNT	—
Arabe	بتعمل حلقة صغيرة عشان عندي امتحان بكرة ومتضيعليش اليوم الله عليك	NOT	—	—

	يا فخر اليوتيوب			
Arabe	هذي مجرد أحداث شغب من فيءة قليلة من الجردان ذو العقلية ***** لكن شففت قبل ماتش كورة ماتوا فية 84 روح بشرية	OFF	—	—

Table 10 : Exemples de Tweet de l'ensemble de données, avec leurs labels correspondants pour chaque sous tache.

Anglais :

Pour l'anglais, nous avons trois ensembles (trois sous taches <A, B et C>) de données chacun contient 10 592 Tweets et 2648 pour le test en anglais, ce qui en fait le plus grand ensemble de données du genre. Nous avons collecté des commentaires de tweeter aléatoires en utilisant les 20 mots vides anglais les plus courants tels que the, of, and, to, ... etc.

Ensuite, nous avons étiqueté ces commentaires de tweeter de manière supervisée et nous avons utilisé pour cela huit modèles (KNeighborsClassificateur, LinearSVC, LogisticRegression, DecisionTreeClassificateur, MultinomialNB, BernoulliNB, RandomForestClassificateur, GradientBoostingClassificateur pour la « machine learning » et aussi avec simpleRNN et LSTM pour le « deep learning ») et Nous avons sélectionné les tweets offensifs pour l'ensemble de test en utilisant la technique supervisée .

Arabe :

L'ensemble de données se compose de 4000 tweets fournies par le centre de recherche scientifique et technique pour le développment de la langue arabe (CRSTDLA) ce corpus lié aux dialectes des pays arabes comme <Qatar, Oman et l'Arabie Saoudite>, et elle est largement observée Dans les communications offensives dans presque tous les dialectes arabes orientaux. Nous l'avons appliqué les technique du prétraitement avec l'utilisation des modèles que nous avons cités précédemment.

III.2.2.Schéma d'étiquetage

Nous avons proposé une hiérarchie de Modélisation du langage offensant, qui classe Chaque exemple utilisant la hiérarchie à trois niveaux schéma de l'OLID [21]

Dans notre « Dataset », nous avons décomposé le contenu offensant en trois sous-tâches suivantes en tenant compte du type et de la cible des infractions :

- Sous-tâche A - Identification d'un langage offensant.
- Sous-tâche B - Catégorisation automatique des types d'infraction.

- Sous-tâche C - Identification de la cible de l'infraction.

Et pour chaque sous tâche nous avons entraîné notre « Dataset » sur 80% et après la création des classificateurs nous avons les tests sur 20%

Voilà un exemple qui fait le « fitting » de notre « Dataset » pour le classificateur KNN :

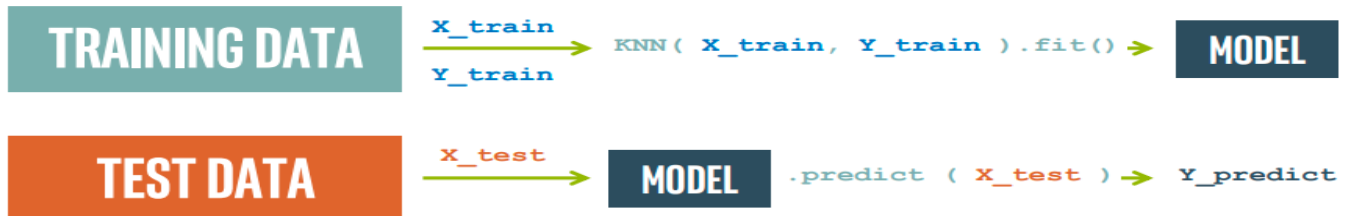


Figure 10 : exemple du fitting pour le classificateur KNN

III.2.2.1. Niveau A (sous tâche_A)

Détection des langues offensantes Le niveau A demande si le texte est offensant (OFF) ou pas (NOT) :

NOT : contenu qui n'est pas offensant.

OFF : contenu offensant.

III.2.2.2. Niveau B (sous tâche_B)

Catégorisation de l'offensive Langue Le niveau B demande si le texte offensant est ciblé (TIN) ou non (UNT).

TIN : ciblée

UNT : non ciblées

III.2.2.3. Niveau C : (sous tâche_C)

Le niveau C catégorise la cible du contenu offensant :

IND : la cible est un individu mentionné explicitement ou implicitement dans la conversation ;

GRP : la cible est un groupe de personnes basé sur l'appartenance ethnique, le sexe, l'orientation sexuelle, croyance religieuse ou autre caractéristique commune

OTH : cibles qui n'entrent dans aucune des catégories précédentes, par exemple, les organisations, les événements et problèmes.

III.3. Technique et méthodes des prétraitements

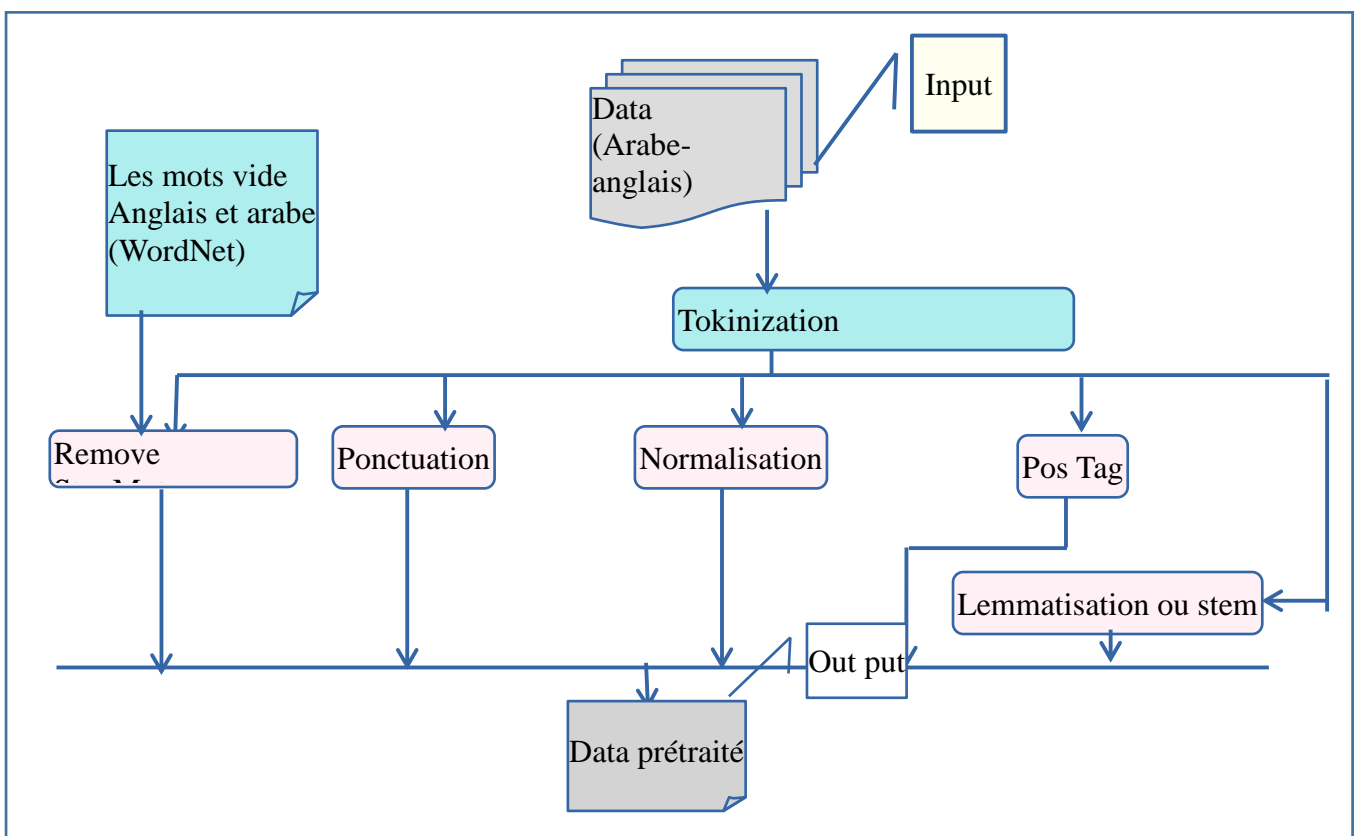
III.3.1. Prétraitement

Le prétraitement une étape qui cherche à standardiser du texte afin de rendre son usage plus facile.

Dans cette partie nous avons effectué une sorte de prétraitement et de normalisation de texte, qui sont des étapes typiques lorsque nous travaillons avec des tweets. Normalisation du texte

Inclus diverses transformations de texte telles que la conversion d'émojis en texte brut, segmentation des hashtags, correction d'erreurs, minuscules, radicalisation et / ou lemmatisation. D'autres techniques comprenaient l'enlèvement de mentions @user, URL, hashtags, émojis, e-mails, dates, chiffres, ponctuation, caractère consécutif Répétitions, mots offensants et / ou mots vides.

Représentation du texte comme un vecteur : Cette étape peut être effectuée via des techniques de sac de mots (« Bag of Mots ») ou « Term Frequency-Inverse Document Frequency » (Tf-IdF). On peut également apprendre des représentations vectorielles (embedding) par apprentissage profond.



Pour le processus du prétraitement après la « tokenisation », Nous avons testé quatre différentes représentations vectorielles de textes (mot, caractère, n-gramme de caractère, combinaison entre les trois <mot>).

Par exemple: <@ We NEED GUN CONTROL >

- Au niveau du mot:

Dans ce niveau nous avons traité les mots de notre Tweet entre un range <1,5>

Mot _represent= ['control', 'gun', 'gun control', 'need', 'need gun', 'need gun control', 'we', 'we need', 'we need gun', 'we need gun control'].

- Au niveau du caractère:

Dans ce niveau nous avons traité les caractères des mots de notre Tweet entre un range <1,5>

char_represent = [' ', 'c', 'co', 'con', 'cont', 'g', 'gu', 'gun', 'gun ', 'n', 'ne', 'nee', 'need', 'c', 'co', 'con', 'cont', 'contr', 'd', 'd ', 'd g', 'd gu', 'd gun', 'e', 'e ', 'e n', 'e ne', 'e nee', 'ed', 'ed ', 'ed g', 'ed gu', 'ee', 'eed', 'eed ', 'eed g', 'g', 'gu', 'gun', 'gun ', 'gun c', 'l', 'n', 'n ', 'n c', 'n co', 'n con', 'ne', 'nee', 'need', 'need ', 'nt', 'ntr', 'ntro', 'ntrol', 'o', 'ol', 'on', 'ont', 'ontr', 'ontro', 'r', 'ro', 'rol', 't', 'tr', 'tro', 'trol', 'u', 'un', 'un ', 'un c', 'un co', 'w', 'we', 'we ', 'we n', 'we ne']

- Au niveau du caractère N-gramme :

Dans ce niveau nous avons traité les caractères des mots de notre Tweet entre un range <1,5> de manière N-gramme (char-wb) <caractère Mot bord> cette option

'char_wb' crée des caractères n_gramme uniquement à partir du texte.

```
from sklearn.feature_extraction.text import TfidfVectorizer
corpus = ['We NEED GUN CONTROL']
vectorizer = TfidfVectorizer(analyzer='char_wb', ngram_range=(1,5))
x = vectorizer.fit_transform(corpus)
print([(w) for w in vectorizer.get_feature_names()])
```

char_wb_represent = [' ', 'c', 'co', 'con', 'cont', 'g', 'gu', 'gun', 'gun ', 'n', 'ne', 'nee', 'need', 'w', 'we', 'we ', 'c', 'co', 'con', 'cont', 'contr', 'd', 'd ', 'e', 'e ', 'ed', 'ed ', 'ee', 'eed', 'eed ', 'g', 'gu',

'gun', 'gun ', 'l', 'l ', 'n', 'n ', 'ne', 'nee', 'need', 'need ', 'nt', 'ntr', 'ntro', 'ntrol', 'o', 'ol', 'ol ', 'on', 'ont', 'ontr', 'ontro', 'r', 'ro', 'rol', 'rol ', 't', 'tr', 'tro', 'trol', 'trol ', 'u', 'un', 'un ', 'w', 'we', 'we ']

- Au niveau du tout <combinaison> :

Dans ce niveau nous avons fait une combinaison à partir de la classe « FeatureUnion » entre les trois représentations précédentes.

```
all_represent = ['char_wb_represent', 'char_represent', 'mot_represent']
```

III.3.1.1. Passage en minuscule

➤ anglais

Tout d'abord Dans un premier temps, nous avons transformé les majuscules en minuscules car les étapes suivantes sont sensibles à la casse : elles considèrent cette phrase “@USER We NEED GUN CONTROL” et “@user we need gun control” comme différents par exemple.

III.3.1.2. « Tokenisation » :

➤ Anglais

Dans cette étape nous allons commencer par découper notre texte en mots. C'est ce qu'on appelle la « tokenisation » ou « tokeniser » en anglais. L'idée est de prendre les phrases puis de les découper en « tokens ». Un « token » peut être un mot, une ponctuation, un chiffre...etc., cependant de nombreux cas ne sont pas facile à traiter : Les dates et heures qui peuvent être séparées par des points, des slashes, des deux points, Les apostrophes ; Les caractères spéciaux (: @, émoji)...etc.

La figure (12) la présente la « «tokenisation » pour la phrase Le passage en minuscule est aussi intégré.

@USER I know! She is still alive....but wondering if her career was railroaded? Odd!
Loved her...especially that day!

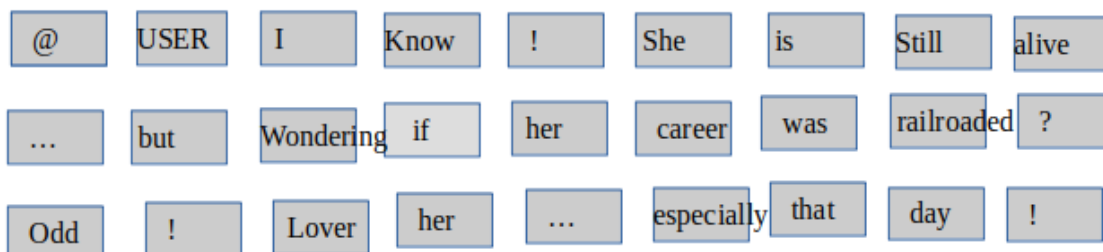


Figure 12 : un exemple de tokenisation des tweets anglais

➤ Arabe

La « tokenisation » pour l'arabe c'est la même pour l'Anglais elle consiste à découper les espaces blanches et les caractères spéciaux avec la segmentation des tweets en unités significative.

On prend ce Tweet par exemple : << user @ يا اخي كلماتك كلها روعة ليس فيها اي مضيعة وقت ♥♥♥♥>>

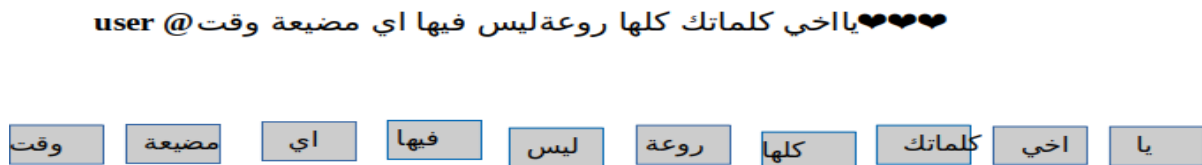


Figure 13 : un exemple de « tokenisation » des tweets arabes

III.3.1.3. la suppression des mots vides

➤ Anglais

Ensuite, nous avons retiré les mots appartenant aux « stopMots » par l'utilisation des librairies existantes en python (Nltk → Motnet). Ces listes se composent de mots qui n'apportent aucune information, qui sont en général très courants et donc présents dans la plupart des documents,

La suppression de ces « stopMots » permet de ne pas polluer les représentations des documents afin qu'elle ne contienne que les mots représentatifs et significatifs. Ce "nettoyage" du texte peut aussi s'accompagner de la suppression d'autres éléments comme les nombres, les dates, la ponctuation etc.

@USER I know! She is still alive...but wondering if her career was railroaded? Odd! Loved her...especially that day!

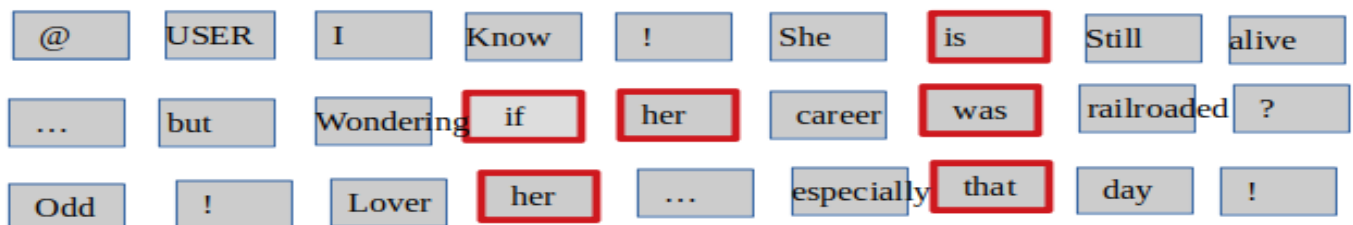


Figure 14 : Mots vides pour la langue anglaise

Arabe



Figure 15 : Mots vides pour la langue arabe

Comme nous l'avons vu pour l'élimination des « stopMots » en anglais

III.3.1.4. la suppression des ponctuations

➤ anglais

La suppression de ponctuation a pour but d'organiser et de rendre le texte facile a traité grâce à une ensemble des fonctions parmi les fonctions que nous avons utilisé :

```
def removePunctuation(text):
    out = text.str.replace('[^\w\s]', '')
    return out
```

Quelques ponctuations parmi cela :

[le point (.), les accolades({}), la virgule (,), point-virgule(;), le point d'interrogation(?) , les opérateurs(+/~/*) et les crochet([]) ..etc.]

III. 3.1.4. Groupement sémantique

Il y a des mots porteurs de sens et séparés en « tokens ». Mais un mot peut être écrit au pluriel, au singulier ou avec différents accords et les verbes peuvent être conjugués aux différents temps et personnes.

Donc nous avons réduire les différences grammaticales des mots Pour le faire, nous disposons de deux méthodes distinctes :

- La « stemmatisation », qui ne prend pas en compte le contexte de la phrase.
- La lemmatisation, qui prend en compte le contexte.

I. La « stemmatisation »

Pour l'arabe La sémantique est l'étude du sens des expressions linguistiques. La quantité de recherche scientifique dans les modèles computationnels de la sémantique est beaucoup plus petite que d'autres domaines du TAL. Cela est peut-être dû à sa plus grande complexité et subtilité. La recherche sur la sémantique en TAL arabe n'est pas différente [22]

➤ **Anglais**

La « stemmatisation » (ou « racinisation ») réduit les mots à leur radical ou racine. Le résultat n'est pas forcément un mot existant, comme vous pouvez le constater dans notre phrase.

Nous avons utilisé la bibliothèque NLTK pour la « stemmatisation » des mots .

@USER I know! She is still alive...but wondering if her career was railroaded? Odd! Loved her...especially that day!

@	USER	I	Know	!	She	is	Still	aliv
...	but	Wonder	if	her	career	was	railroad	?
Odd	!	Love	her	...	especi	that	day	!

Figure 16 : « stemmatisation » pour la langue anglaise

Cependant, celle-ci peut parfois réduire deux mots à l'orthographe proche, mais aux sens différents, à une même racine.

➤ **Arabe**

Le morphème racine en arabe est une séquence de trois, de quatre ou très rarement de cinq consonnes (appelées radicaux). La racine signifie un sens abstrait partagé par tout ses dérivés. En effet, à chaque racine correspond un champ sémantique et à l'aide de différents patrons, on peut générer une famille de mots appartenant à ce champ sémantique, par exemple la racine ب-ت-ك peut engendrer quinze mots autour de la notion d'écriture

Pour le même tweet : << user @ يا اخي كلماتك كلها روعة ليس فيها اي مضيعة وقت >>

@USER يا اخي كلماتك كلها روعة ليس فيها اي مضيعة وقت ❤️❤️❤️"

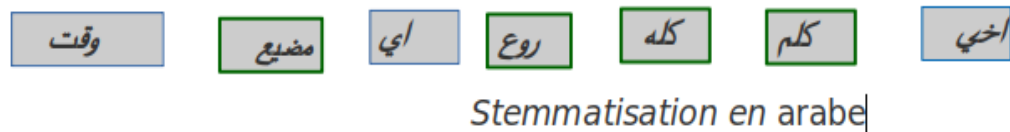


Figure 17 : la stemmatisation en arabe

II. La Lemmatisation

La lemmatisation, qui prend en considération le contexte dans lequel le mot est écrit, a pour but de trouver la forme canonique du mot, le lemme. Par conséquent, elle doit se faire après la transformation des lettres majuscules en minuscules et avant la « tokenisation » car les mots présents avant et après sont importants pour déterminer la nature du mot.

Le lemme correspond à l'infinitif des verbes et à la forme au masculin singulier des noms, adjectifs et articles. La lemmatisation de la phrase d'exemple est présentée dans la figure suivante.

@USER I know! She is still alive...but wondering if her career was railroaded? Odd! Loved her..especially that day!

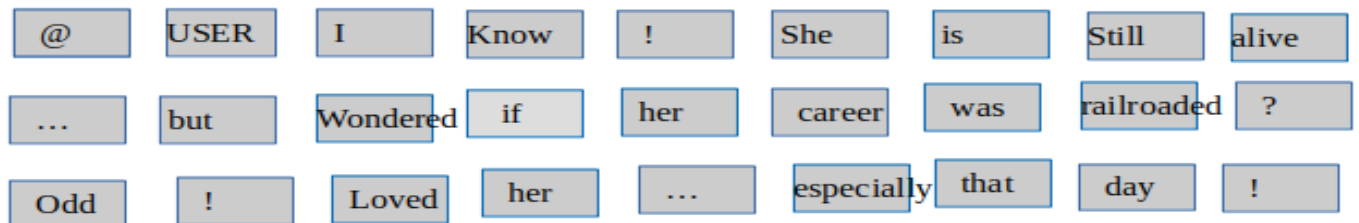


Figure 18 : la lemmatisation en anglais

En conclusion, la « stemmatisation » peut être considérée comme une forme brute et rapide alors que la lemmatisation tente de garder au maximum le sens des phrases.

Quelle que soit la méthode avons-nous choisi (« stemmatisation » ou « lemmatisation ») nous sommes maintenant en possession de mots utiles et réduits pour chaque Tweet. Ils constituent ce qu'on appelle le dictionnaire du document à analyser.

III.3.2. La représentation des expériences

Expérience : la détection et l'identification des Tweets (soit en langue anglaise ou en langue arabe) pour dire que ce tweet il est offensif ou non. Nous avons les testé sur tout les

cas possibles par <mot, caractère, caractère N-gramme et tous>et aussi nous avons combiné ce derniers avec les technique du prétraitement<remove punctuation, remove stopMot , postag, stemmatisation et lemmatisation >et nous avons répété le même traitement sur toutes les taches <A,B et C> (pour anglais et pour l'arabe) .

➤ **Stage zéro**

Nous avons effectué aucune technique du prétraitement sauf les quatres représentations vectorielles (mot, caractère, c caractère N-gramme, tout<all>) ce qui nous donne quatre expériences. L'entrée et la sortie de cette étape comporte les huit classificateurs avec leurs scores.

➤ **Stage 1**

Nous avons effectué cinq prétraitement soit (la suppression du mot vide ou la suppression de ponctuation ou Pos tag ou lemmatisation ou stemmatisation) avec les quatres représentations vectorielles (mot, caractère, caractère n-gramme, tous) après la combinaison ça nous donne 20 expériences. L'entrée de cette étape comporte les huit classificateurs et la sortie comporte seulement six classificateurs les deux restants seront éliminés à cause du score (ils ont les valeurs les plus basses).

➤ **Stage 2**

Nous avons appliqué les cinq prétraitements à la fois avec les quatre représentations vectorielles sur les six classificateurs de la sortie de l'étape précédente qui nous donnent quarante(40) expériences, la sortie de cette étape comporte quatre classificateurs.

➤ **Stage 3**

Nous avons appliqué les cinq prétraitements à la fois avec les quatre représentations vectorielles sur les quatre classificateurs de la sortie de l'étape précédente qui nous donnent quarante(40) expériences, la sortie de cette étape comporte deux classificateur, les deux restants nous avons les éliminés.

➤ **Stage 4**

Nous avons appliqué aussi les cinq prétraitements à la fois avec les quatre représentations vectorielles sur les deux classificateurs de la sortie de l'étape précédente, la sortie de cette étape comporte un seul classificateur.

➤ Stage 5

Nous avons appliqué aussi les cinq prétraitements avec les quatre représentations vectorielles sur un seul classificateur. de la sortie de l'étape précédente ça vous dire que pour chaque représentation vectorielle nous avons appliqué les cinq prétraitements.

La figure en dessous représente l'architecture en détail concernant le filtrage pour une seule sous tâche (A), nous avons obtenu 128 expériences pour chaque sous tâches

Donc en total nous aurons 512 expériences.

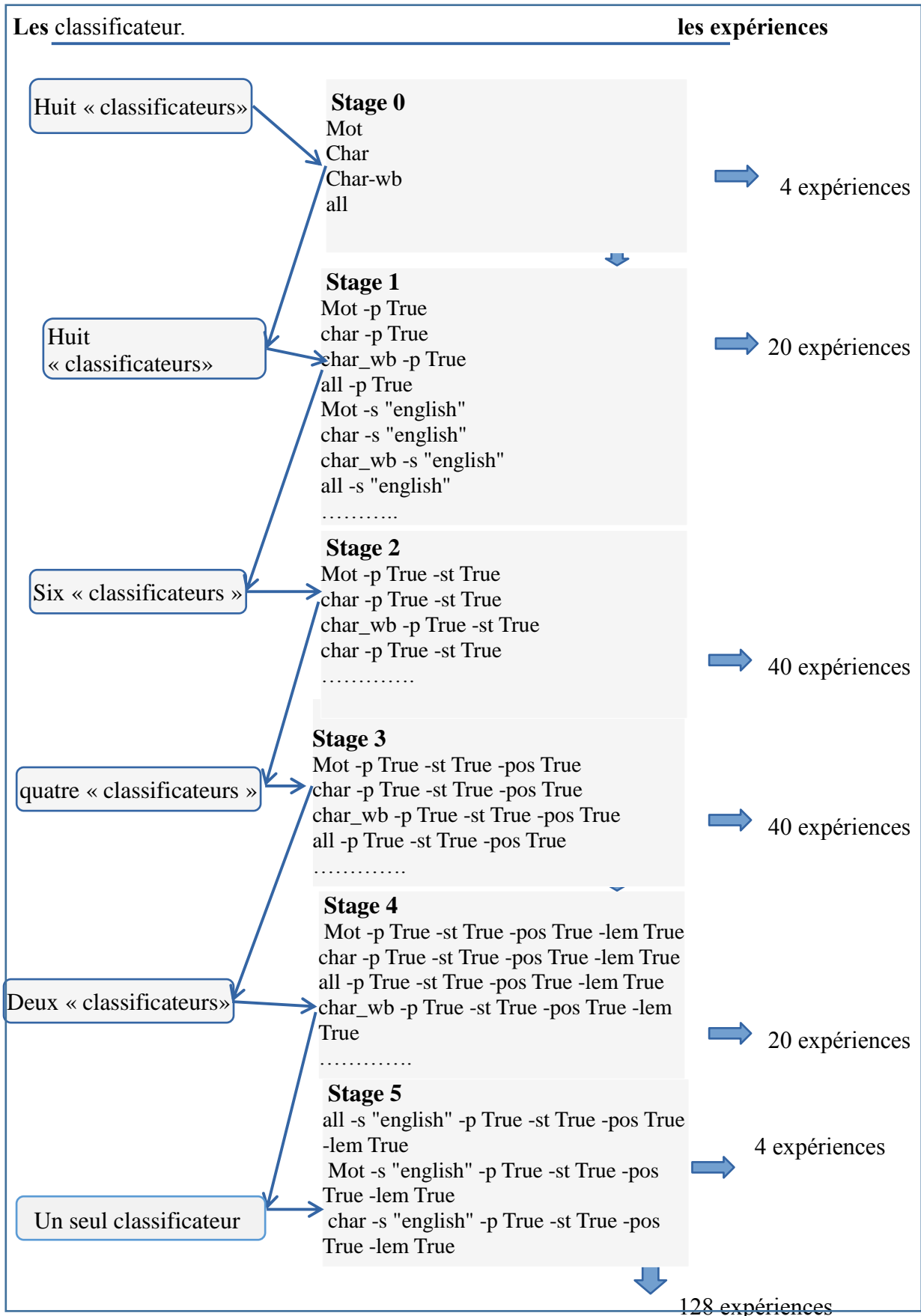


Figure 19 : Architecture de processus du prétraitement

III.3.3.Représentations vectorielles des Tweets

Dans cette étape nous avons affaire à la représentation des tweets de notre corpus (codage des textes) de manière consécutive mais avant tout pourquoi la représentation vectorielle des mots ?

- Permettre de Représenter le conceptuel « Signification » des mots.
- les vecteurs de mots proches ont une signification similaire.

Pour les algorithmes d'apprentissage automatique nous avons traité les représentations vectorielles des tweets par la technique du Tf-idf, et Pour les algorithmes d'apprentissage automatique en profond par la technique de Word Embedding.

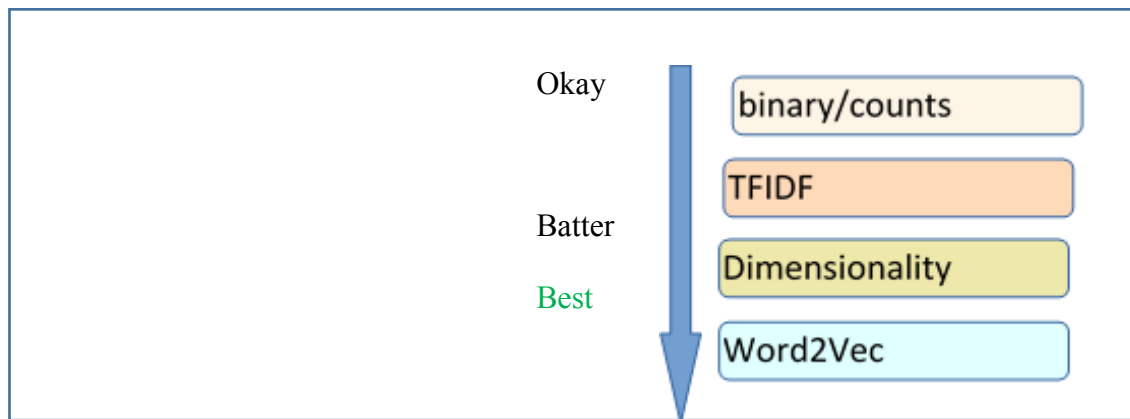


Figure 20 : Different representation vectorielle

III. 3.3.1 Le Bag of Word(BoW)

Ici, nous souhaitons représenter des tweets par un vecteur. Pour cela, on fait correspondre chaque composante du vecteur-document à un mot du dictionnaire du corpus. Une composante contient donc une valeur pour chacun des mots existant dans l'ensemble des textes que nous traitons. Cette valeur peut être, par exemple, le nombre d'occurrences du mot dans le document. Si un mot n'y est pas présent on lui donnera la valeur 0. Il s'agit d'une approche dite « bag of Word » ou (sac des mots). La figure suivante illustre la vectorisation :

Tweet 1 : @USER I know She is still alive but wondering if her career was railroaded

Tweet 2: @USER You are dead to me!

dictionary	I	Know	she	still	alive	but	Wonder	career	was	railroaded	you	dead
Tweet 1:	1	1	1	1	1	1	1	1	1	1	0	0
Tweet 2 :	0	0	0	0	0	0	0	0	0	0	1	1

Figure 21 : Exemple du bag of Mot

Cependant d'autres approches sont possibles et permettent d'être plus pertinent sur la représentation de nos documents. Elles se basent sur les hypothèses suivantes :

- Plus le mot est présent dans un grand nombre de documents, moins il apporte d'information permettant de les distinguer
- Un mot présent dans peu de document permet de bien les caractériser
- Plus un mot est présent dans un document, plus il a de poids dans ce document.

III. 3.3.2 Calcul du TF-IDF

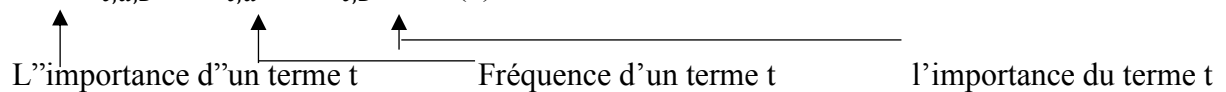
Nous avons utilisé le TF-IDF. Cet acronyme anglais correspond à un poids calculé et affecté pour chaque mot de tweets du corpus. Il se décompose en deux parties :

La fréquence d'apparition d'un mot dans un tweet

Le nombre de tweets dans lequel le mot apparaît une fois ou plus par rapport au nombre de tweets total du notre corpus.

La combinaison de ces deux indicateurs donne le TF-IDF. Ce score présente l'importance d'un mot dans un document et prend en compte sa rareté dans l'ensemble du corpus. Les termes les moins présents dans le corpus ont donc un poids plus important car ils sont plus discriminants. Donc on a utilisé ce score comme valeur des vecteurs représentant nos tweets.

$$TFIDF_{t;d;D} = TF_{t;d} * IDF_{t;D} \quad (9)$$



Dans un document d

dans un document d

dans l'ensemble des documents

III.3.3.3. Réduction de la taille des vecteurs obtenus

Les vecteurs qui nous ont obtenu dans le bag of Mots classique ou le TF-IDF ont la taille du vocabulaire du nos corpus. Il peut être intéressant, afin de limiter le temps de calcul et le stockage, de limiter cette taille, Pour ce faire :

Nous avons Éliminé les mots non discriminants, c'est à dire ceux qui ont un TF-IDF trop faible dans l'ensemble du corpus ;

Et nous ont Défini des bornes correspondant au nombre de documents maximum et minimum dans lequel un mot doit se trouver pour être gardé dans le vocabulaire ;

On a fait ces étapes parce que Ces vecteurs réduits nous permettent d'améliorer les performances d'outils de Machine Learning qui on va les utilisés dans les étapes suivantes et aussi ils ont limité le bruit présent dans les données de nos corpus.

III.3.3.4. « Mot Embedding » :

Dans la partie d'apprentissage automatique profond (deep learning) :

Nous avons travaillé avec « word embedding » (le « word embedding » est une représentation apprise pour un texte ou les mots qui ont la même signification ont une représentation similaire.) il est aussi la clé avancée de l'apprentissage profond sur les problèmes complexes de traitement du langage naturel.

Mot2Vec:

est l'une des techniques les plus populaires pour apprendre l'intégration de mots à l'aide d'un réseau neurone peu profond .Il a été développé par Tomas Mikolov en 2013 chez Google [23].

Mot2Vec est une méthode pour construire une telle embedding .Il peut être obtenu en utilisant deux méthodes (toutes deux impliquant des réseaux de neurones): Skip Gram et Common « Bag Of Mots » <CBOW>.

I.e Modèle CBOW:

Cette méthode prend le contexte de chaque mot comme entrée et tente de prédire le mot correspondant au contexte.

Par exemple ce Tweet :<<Antifa is a terrorist organization>>

Alors l'entrée au notre réseau de neurones soit les deux mots « Antifa » et « terrorist » et nous essayons de prédire un mot cible (organization) en utilisant deux contextes à l'entrée.

Nous utilisons le codage « one hot vector » pour chaque contexte.

Antifa= [1, 0, 0]

Terrorist= [0, 1, 0]

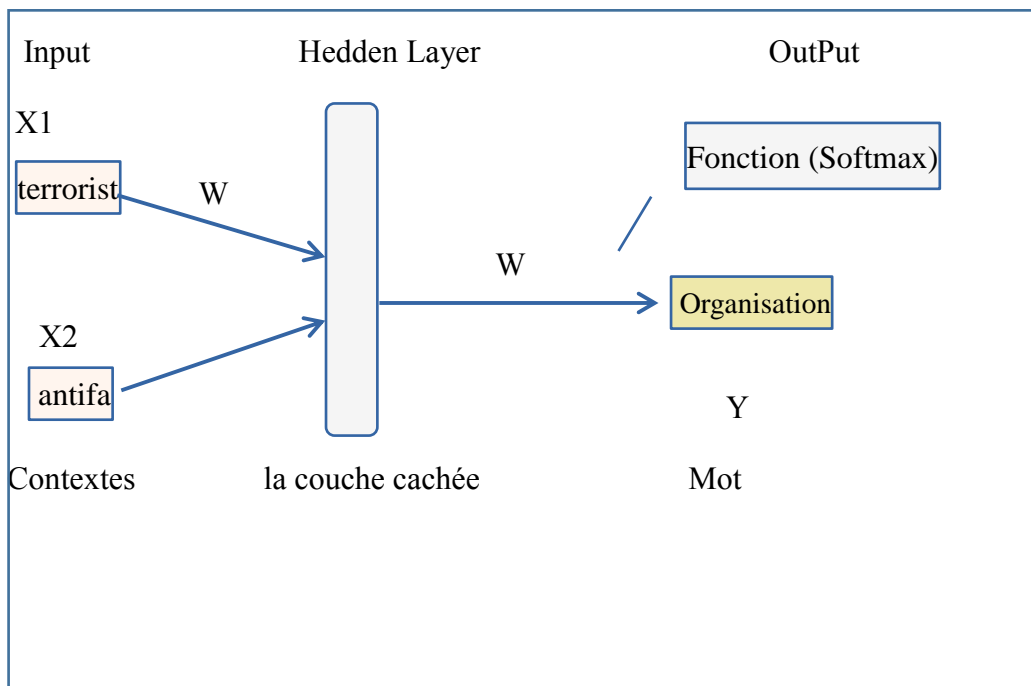


Figure 22 : architecture simple du modèle CBOW

L'entrée ou les contextes sont des vecteurs codé à one hot vecteur de taille V. La couche cachée contient N neurones et la sortie c'est un nouveau vecteur de longueur V.

Les neurones de la couche cachée copient simplement la somme pondérée des entrées dans la couche suivante. Il n'y a pas d'activation comme sigmoïde, « tanh » ou « ReLU ». La seule non-linéarité concerne les calculs « softmax » dans la couche de sortie.

Le Modèle Skip-gram:

Cette méthode prend les mots en entrée et tente de prédire les contextes correspondant à ce mot.

Par exemple si nous prenons l'exemple précédent : <<Antifa is a terrorist organization>>

Alors l'entrée au notre réseau de neurones soit un mot < Antifa > et nous essayons de prédire les contextes cible ('terrorist', 'organization') en utilisant un seul mot à l'entrée.

Nous utilisons le codage « one hot vector » pour chaque mot.

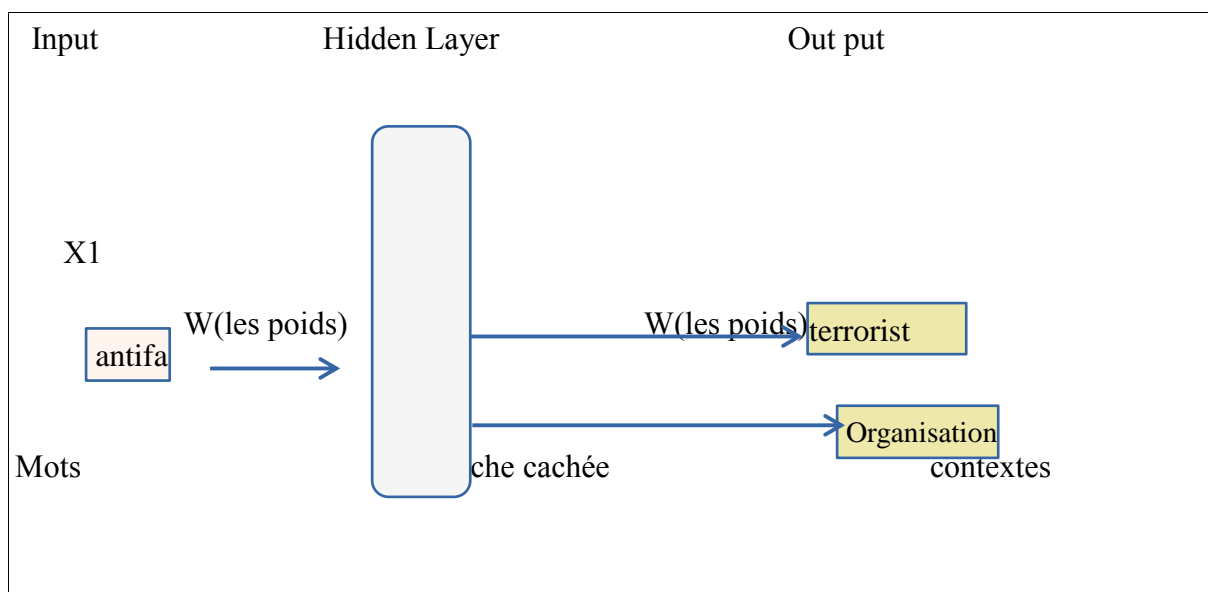


Figure 23 : architecture simple du modèle Skip-Gram

Selon Mikolov Skip-gram fonction bien avec une petite quantité de données et représente bien les mots rares, par contre Cbow est plus rapide et à de meilleures représentation pour les mots les plus fréquents.[24]

Une petite différence entre les techniques utilisées :

Word Embedding	Tf-Idf
vecteur multi dimensionnel qui tente à Capturer une relation de mots avec d'autres mots.	Les TF-IDF sont des vecteurs clairsemés où le nombre de valeurs non nulles dans le vecteur est égal au nombre de mots uniques dans le document.
Entraîner sur un grand corpus externe	Entraîner sans données externes.
doit être appliqué chaque mot individuellement.	peut être appliqué à chaque document formé à la fois.
plus intensif en mémoire	moins intensif en mémoire

En résumé : nous avons fait les différentes étapes nécessaires au traitement des données textuelles afin de les rendre utilisables. Elles nous permettent d'obtenir des représentations vectorielles de nos tweets, exploitables dans la suite des processus de NLP. Bien entendu, les différentes méthodes de vectorisation existantes.

III.4. Les Modèles d'apprentissage automatique et profond

III.4.1. Machine learning (ML) :

Est un sous-ensemble de l'intelligence artificielle (IA) qui offre aux systèmes la possibilité d'apprendre et de s'améliorer automatiquement à partir de l'expérience sans être explicitement programmés. En ML, il existe différents algorithmes (logistique régression, arbre de décision, La forêt aléatoire.....etc.) qui aident à résoudre des problèmes. [25]

➤ Deep learning (DL) :

Est un sous-ensemble de l'apprentissage automatique, qui utilise les réseaux de neurones pour analyser différents facteurs avec une structure similaire au système neurone humain.

L'apprentissage profond structure les algorithmes en couches pour créer un «réseau neurone artificiel» capable d'apprendre et de prendre des décisions intelligentes par lui-même.

L'apprentissage en profondeur est un sous-domaine de l'apprentissage automatique. Alors que les deux relèvent de la grande catégorie de l'intelligence artificielle, l'apprentissage en profondeur est ce qui alimente l'intelligence artificielle la plus humaine. [25]

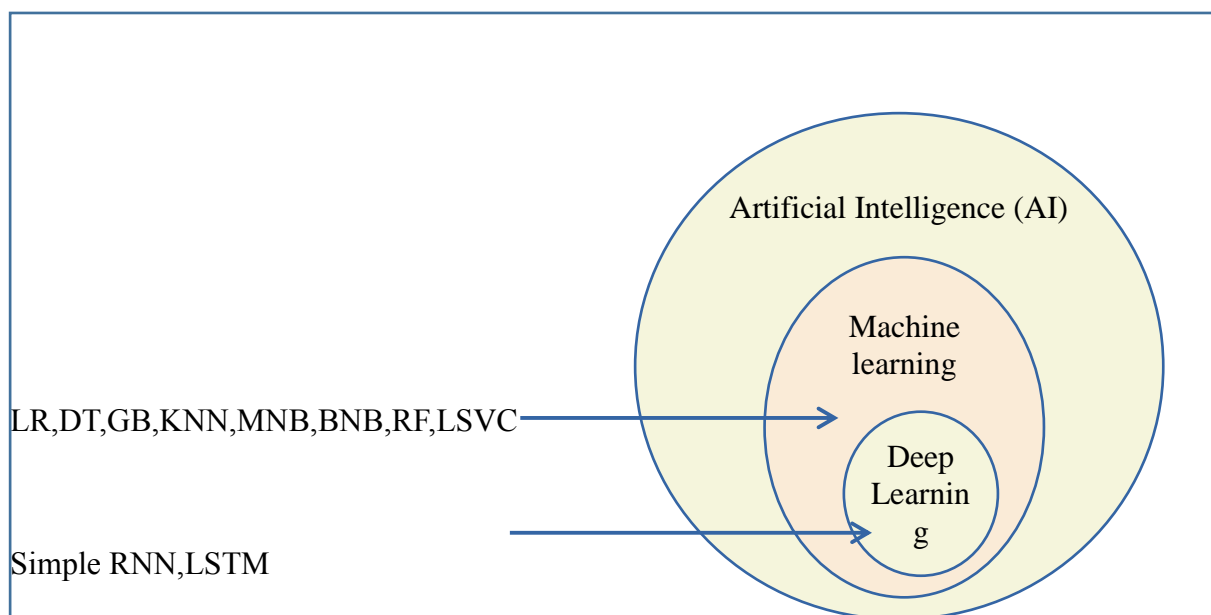


Figure 23 : Présentation des différents apprentissages

Dans cette étape de la modélisation nous ferons un tour et nous avons prend quelques algorithmes d'apprentissage automatique les plus populaires d'un côté supervisé .À partir d'un échantillon de population qui représente nos données, on répartit les données en deux fichiers, les données d'entraînement (80%) et les données de test (20%). La première catégorie de données servira pendant la phase d'apprentissage du modèle alors que le second (test)sera utilisé pour évaluer la qualité de prédiction du modèle. Le but n'est donc pas de construire une fonction qui prédira avec une précision optimale les valeurs des variables cibles mais une fonction qui se généralisera au mieux pour prédire des valeurs de données qui n'ont pas encore été observées.

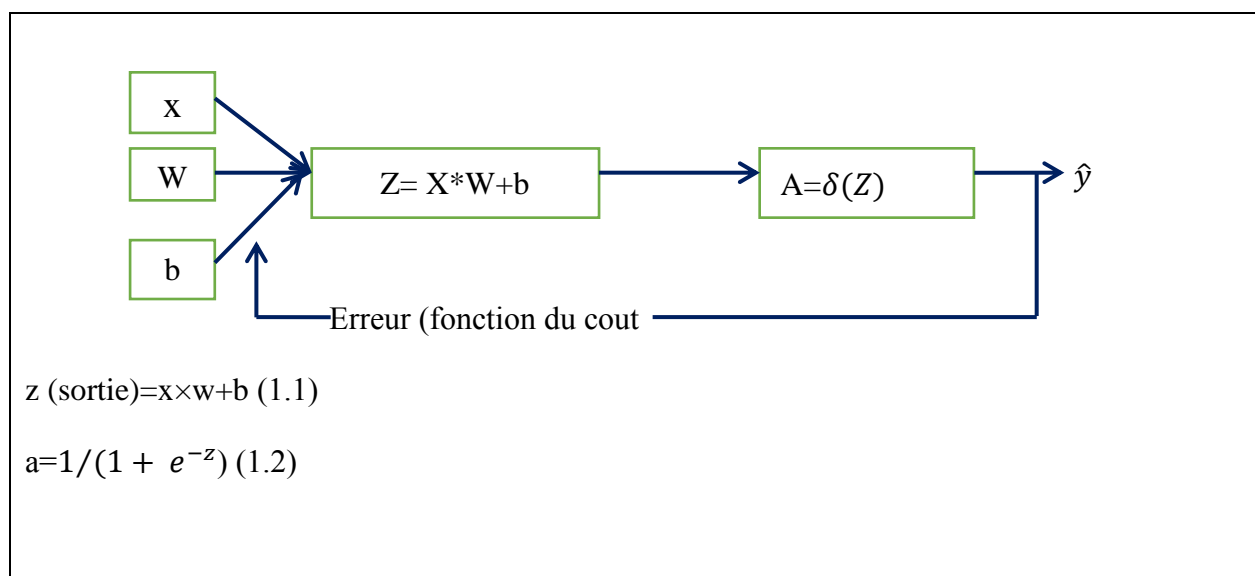
III.4.1.1. Algorithmes de régression

La régression c'est un processus concerne la modélisation de la relation entre les variables qui est affinée de manière itérative à l'aide d'une mesure d'erreur dans les prédictions faites par le modèle pour cela on a choisi l'un des modèles (algorithmes) suivant :

III.4.1.1.1. Régression logistique (LR)

Dans ce processus nous avons donné (X, Y) , X étant une matrice de valeurs avec m exemples et n caractéristiques (des tweets) et Y étant un vecteur avec m exemples (pour sous tâche _A c'est off ou Not off).

L'objectif est de former le modèle à prédire à quelle classe appartiennent les valeurs futures. Principalement, nous créons une matrice de poids avec une initialisation aléatoire. Ensuite, nous le multiplions par fonctionnalités.



Le but de l'algorithme de régression logistique est de créer une frontière de décision linéaire séparant deux classes par exemple (classe pour les mots offensive et autre pour les mots non offensive). Cette frontière de décision est donnée par une probabilité conditionnelle.

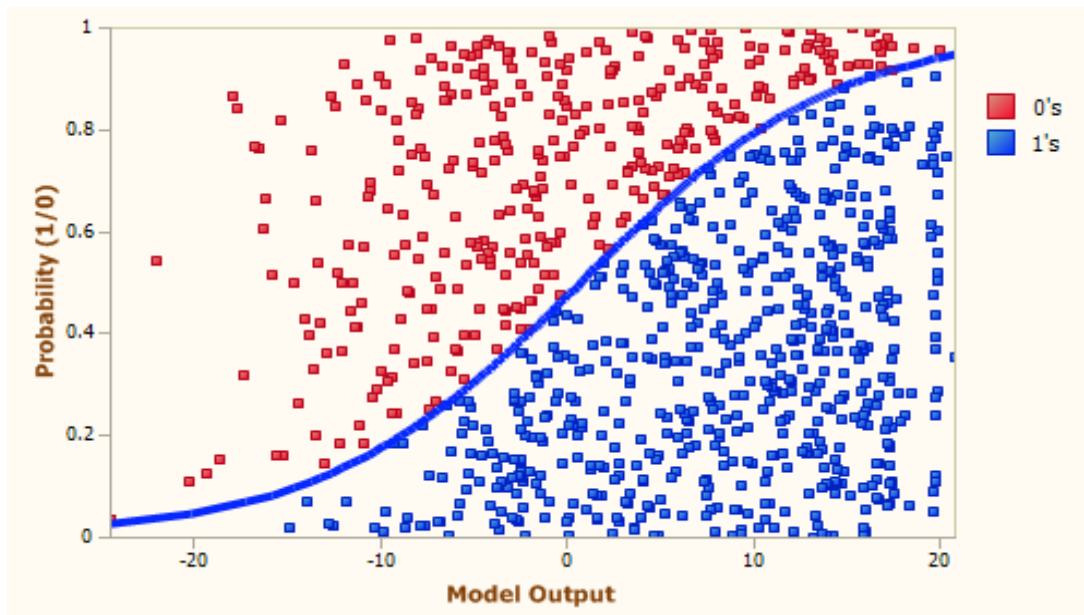


Figure 24 : Modèle logistique régression (LR) Wikipédia

On suppose pour la sous tâche _A que la classe au-dessus de la ligne bleue (limite de décision) c'est-à-dire le « les taches Bleues » (des mots offensives) soit classée comme «1 » et la classe en dessous de la limite de décision « les taches en rouge » (not offensive) soit définie comme «0 ». Ce que fait la régression logistique, c'est qu'elle calcule une probabilité conditionnelle

$$P(Y=1 | x; w)$$

Probabilité pour la classe '1'

$$P(Y=0 | x; w)$$

Probabilité pour la classe '0'

III. 4.1.2 Algorithmes basés sur des instances

III. 4.1.2 .1. Support vector machine (svm)

Les machines à vecteurs de support ou SVM (Séparateurs à vastes marges) sont des algorithmes qui séparent les données en classes. Pendant l'entraînement, un SVM trouve une ligne qui sépare les données d'un jeu en classes spécifiques et maximise les marges (les distances entre les frontières de séparation et les échantillons les plus proches) de chaque classe. Après avoir appris les lignes de classification, le modèle peut ensuite les appliquer aux nouvelles données.

Les spécialistes placent le SVM dans la catégorie des « classificateurs linéaires » : l'algorithme est idéal pour identifier des classes simples (classe des mots offensive ou not offensive...Etc.) qu'il sépare par des vecteurs nommés hyperplans. Il est également possible de programmer l'algorithme pour les données non linéaires, que l'on ne peut pas séparer clairement par des vecteurs. Mais, avec des données d'entraînement hypercomplexes – visages, traits de personnalité, génomes et matériel génétique

Les systèmes de classes deviennent plus petits et plus difficiles à identifier et nécessitent un peu plus d'assistance humaine.

Les SVM dits non linéaires sont souvent mis à contribution pour classificateur des images ou des mots, des phrases et des entités, et parmi les méthodes du SVM comme (SVRegression et svelclassification) pour notre nous avons travaillé avec linearSVC.

III. 4.1.2.2. Linear support vector classificateur (LSVC)

La méthode SVC (Linear Support Vector Classificateur) applique une fonction de noyau linéaire pour effectuer la classification et fonctionne bien avec un grand nombre d'échantillons. Si nous le comparons avec le modèle SVC, le SVC linéaire a des paramètres supplémentaires tels que la normalisation de pénalité qui applique «L1» ou «L2» et la fonction de perte. La méthode du noyau ne peut pas être modifiée dans SVC linéaire, car elle est basée sur la méthode linéaire du noyau.

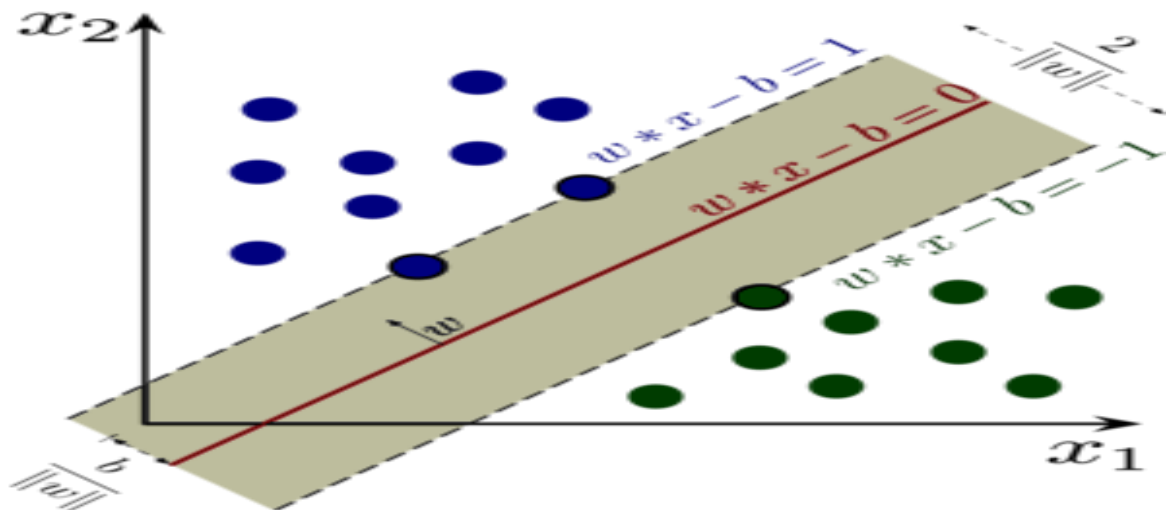


Figure 25 : modèle Linéaire Support Machine (wikipédia)

Nous avons donné un ensemble de données de formation de n points de forme:

$$(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$$

Où y_i sont soit 1 (offensive <avec des points bleus>), soit -1 (Not offensive <avec des points verts>), et x un vecteur de dimensionnelle réel

- $W * x - b = 1$ → pour la classe (OFF) support 1
- $W * x - b = -1$ → pour la classe (NOT OFF) support 2
- $W * x - b = 0$ → hyperplan
- Et la distance entre support 1 et support 2 égale $\|W\| = 2$
- $b / \|W\|$ → détermine le décalage de hyperplan par rapport à l'origine le long du vecteur normale \vec{w} .

III. 4.1.2.3. Plus proche voisin (KNN) :

L'algorithme KNN (K-nearest Neighbors) est une méthode d'apprentissage supervisé. Il peut être utilisé aussi bien pour la régression que pour la classification.

L'algorithme de regroupement k-Nearest Neighbor est une solution de regroupement populaire pour les tâches TDT (topic detection and tracking) [26].

Principe de K-NN : “dis-moi qui sont tes voisins, je te dirais qui tu es...”.

Pour effectuer une prédiction, l'algorithme K-NN ne va pas calculer un modèle prédictif à partir d'un Training Set comme c'est le cas pour la régression logistique ou la régression linéaire. En effet, K-NN n'a pas besoin de construire un modèle prédictif. Ainsi, pour K-NN il n'existe pas de phase d'apprentissage proprement dite. Pour pouvoir effectuer une prédiction, K-NN se base sur le jeu de données pour produire un résultat.[27]

-Pour effectuer une prédiction, l'algorithme K-NN va se baser sur le jeu de données (les tweets dans notre cas) en entier. En effet, pour une observation, qui ne fait pas parti du jeu de données, qu'on souhaite prédire, l'algorithme va chercher les K instances du jeu de données (label) les plus proches de notre observation. Ensuite pour ces voisins, l'algorithme se basera sur leurs variables de sortie (output variable) pour calculer la valeur de la variable de l'observation qu'on souhaite prédire.

Par ailleurs :

- Si K-NN est utilisé pour la régression, c'est la moyenne (ou la médiane) des variables des plus proches observations qui servira pour la prédiction
- Si K-NN est utilisé pour la classification, c'est le mode des variables des plus proches observations qui servira pour la prédiction.[33]

On peut schématiser le fonctionnement de K-NN en l'écrivant en pseudo-code suivant :

Début Algorithme

Données en entrée :

Un ensemble de données D.

Une fonction de définition de distance d.

Un nombre entier K

Pour une nouvelle observation X (tweet) dont on veut prédire sa variable de sortie y Faire :

Calculer toutes les distances de cette observation X (tweet) avec les autres observations du jeu de données D

Retenir les K observations du jeu de données D les proches de X en utilisation la fonction de calcul de distance d

Prendre les valeurs de y des K observations retenues :

3.1. Si on effectue une régression, calculer la moyenne (ou la médiane) de y retenues

3.2. Si on effectue une classification, calculer le mode de y retenues

4. Retourner la valeur calculée dans l'étape 3 comme étant la valeur qui a été prédite par KNN pour l'observation X.

Fin Algorithme

- **La distance euclidienne :**

Distance qui calcule la racine carrée de la somme des différences carrées entre les coordonnées de deux points : $D_e(x, y) = \sqrt{\sum_{j=1}^n (x_j - y_j)^2}$ (2.1)

- **Distance Manhattan :**

La distance de Manhattan : calcule la somme des valeurs absolues des différences entre les coordonnées de deux points : $D_m(x, y) = \sum_{i=1}^k |x_i - y_i|$ (2.2)

Le choix de la valeur à utiliser pour effectuer une prédiction avec K-NN, varie en fonction du jeu de données.

En règle générale, moins on utilisera de voisins (un nombre petit) plus on sera sujette aux sous apprentissage (underfitting). Par ailleurs, plus on utilise de voisins (un nombre K grand) plus, sera fiable dans notre prédiction (overfitting).

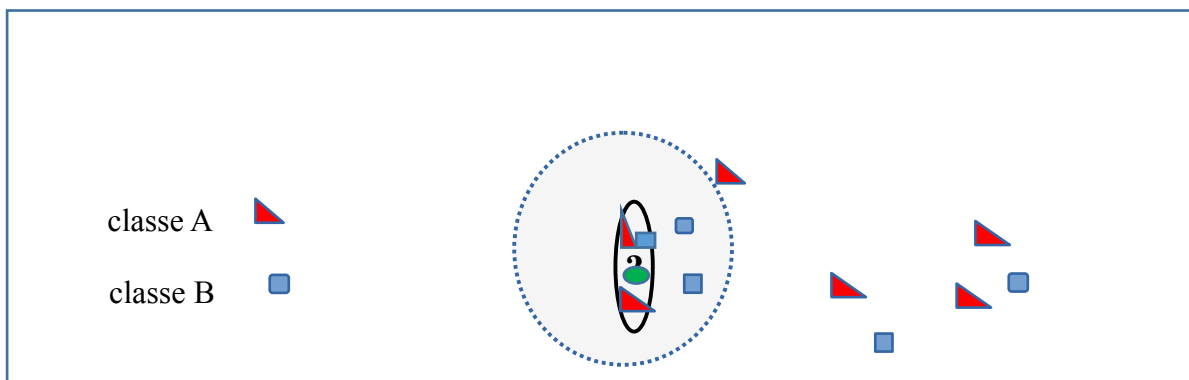


Figure 26 : Modèle KNN

Exemple:

- les triangles représentent la classe des tweets Not offensive
- Les carrés représente la classe des tweets offensive

Quelle est la classe du point vert ?

Celle des triangles rouges (délimitée par le cercle continu) ou celle des carrés bleus (cercle tracé en pointillés) ? Si le nombre de plus proches voisins, k , est fixé à 3, la classe du point vert est celle des triangles rouges ici le point vert sera un tweet not offensif, car ces derniers sont au nombre de 2 contre un seul carré bleu. Si k vaut 5, la classe du point vert est celle des carrés bleus, au nombre de 3 contre 2 triangles rouges donc le point vert sera un tweet offensif.

III.4.1.3 Algorithmes d'arbre de décision

Les arbres de décision sont d'un type d'apprentissage automatique supervisé (c'est-à-dire que vous expliquez ce qu'est l'entrée et quelle est la sortie correspondante dans les données d'apprentissage) où les données sont continuellement divisées en fonction d'un certain paramètre. L'arbre peut être expliqué par deux entités, à savoir les nœuds de décision et les feuilles. Les feuilles sont les décisions ou les résultats finaux. Et les nœuds de décision sont l'endroit où les données sont divisées.

III.4.1.3.1. Arbre de décision (DT) :

Un algorithme d'arbre de décision Représente graphiquement les données en branches pour montrer les résultats possibles de diverses actions. Il classe et prédit les variables de réponse en fonction des décisions passées.

On considère d'abord le problème de classement. Chaque Tweet (x) de la base de données (Dataset) est représenté par un vecteur multidimensionnel $((x_1, x_2, \dots, x_n))$ correspondant à l'ensemble de variables descriptives du point. Chaque nœud interne de l'arbre correspond à un test fait sur une des variables (x_i) :

- Variable catégorielle : génère une branche (un descendant) par valeur de l'attribut ;
- Variable numérique : test par intervalles (tranches) de valeurs.

Les feuilles de l'arbre spécifient les classes (off, Not, IND, GRP, TINetc.).

Une fois l'arbre construit, classer un nouvel candidat se fait par une descente dans l'arbre, de la racine vers une des feuilles (qui encode la décision ou la classe). A chaque niveau de la descente on passe un nœud intermédiaire où une variable (x_i) est testée pour décider du chemin (ou sous arbre) à choisir pour continuer la descente.

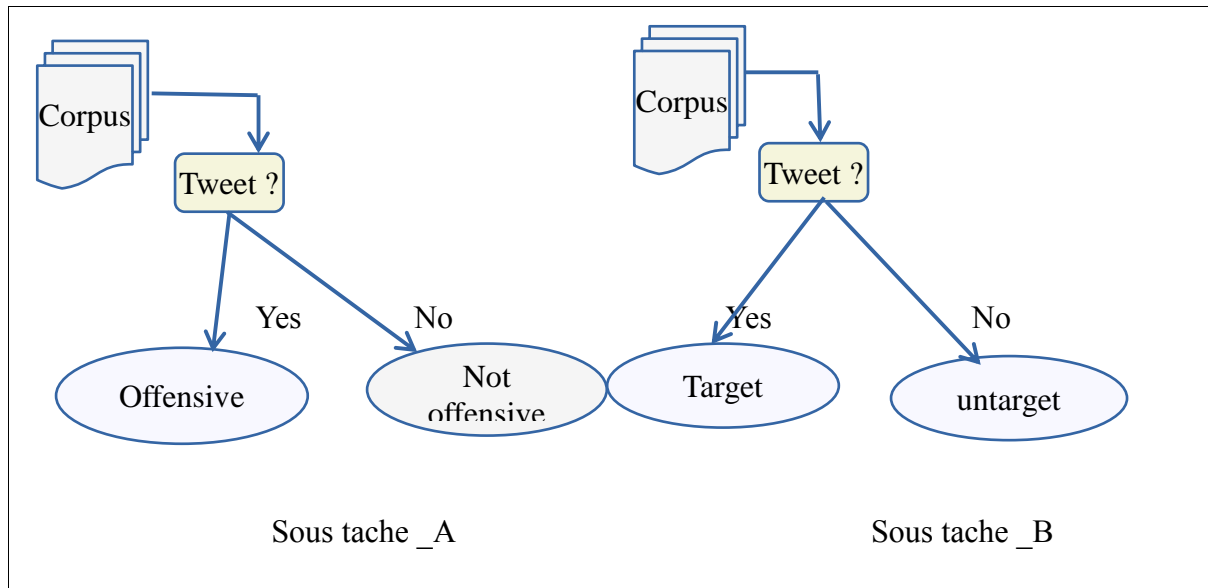


Figure 27 : Arbres de décision (DT)

III.4.1.4. Algorithmes bayésiens

III.4.1.4.1 Naïve bayes

Nous avons travaillé avec théorème de Bayes permet de calculer la probabilité postérieure $P(c | x)$ à partir de $P(c)$, $P(x)$ et $P(x | c)$. Considérez l'équation suivante :

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \quad (3.1)$$

$$P(c | x) = P(x_1 | c) * P(x_2 | c) * \dots * P(x_n | c) * P(c) \quad (3.2)$$

$P(c | x)$: probabilité postérieure de classe (c, cible) donnée prédicteur (x, attributs). Cela représente la probabilité que c soit vrai, à condition que x soit vrai.

$P(c)$: est la probabilité de classe a priori. Il s'agit de la probabilité de classe observée parmi toutes les observations.

$P(x | c)$: est la vraisemblance qui est la probabilité de la classe donnée par le prédicteur. Cela représente la probabilité que x soit vrai, à condition que x soit vrai.

$P(x)$: est la probabilité a priori du prédicteur. Il s'agit de la probabilité observée du prédicteur parmi toutes les observations.

On prend cet **exemple**:

Tweet 1: @USER @USER He raised a hand ----->Not offensive

Tweet 2: Alaye turn if your maga don pay ---->offensive

Tweet 3: @USER @USER He is losing his mind ---> offensive

Tweet 4: @USER She is your sister!lol ---> Not offensive

$$P(\text{offensive} | \text{maga}, \text{losing}) = P(\text{maga} | \text{offensive}) * P(\text{losing} | \text{offensive}) * P(\text{offensive}) / P(\text{losing}) * P(\text{maga})$$

$$P(\text{not offensive} | \text{hand}, \text{sister}) = P(\text{hand} | \text{not offensive}) * P(\text{sister} | \text{not offensive}) * P(\text{not offensive}) / P(\text{hand}) * P(\text{sister}).$$

$$P(\text{offensive}) = \text{cardinalité}(\text{offensive}) / \text{cardinalité}(\text{tous les Label} < \text{offensive}, \text{not offensive})$$

L'algorithme naïf de Bayes donne des performances utiles malgré les variables corrélées dans l'ensemble de données, même s'il a une hypothèse de base d'indépendance entre les caractéristiques. La raison en est que dans un ensemble de données donné, deux attributs peuvent dépendre l'un de l'autre,

Mais la dépendance peut se répartir uniformément dans chacune des classes. Dans ce cas, l'hypothèse d'indépendance conditionnelle de Bayes naïf est violée, mais c'est toujours le classificateur optimal.

Et pour la vitesse il est rapide qu'elle converge vers sa précision asymptotique à un rythme différent de celui d'autres méthodes, comme la régression logistique, les machines vectorielles de support, etc.

III.4.1.5 Algorithmes d'ensemble

III.4.1.5.1. Forêt aléatoire (RF)

Nous avons travaillé aussi avec Les forêts aléatoires (RF) qui construisent de nombreux arbres de décision individuels lors de la formation. Les prédictions de tous les arbres sont regroupées pour faire la prédiction finale ; le mode des classes pour la classification ou la prévision moyenne pour la régression. Comme ils utilisent une collection de résultats pour prendre une décision finale, ils sont appelés techniques d'Ensemble.

La différence entre lui et l'arbre de décision c'est que l'arbre de décision est construit sur un ensemble de données entier, en utilisant toutes les caractéristiques / variables d'intérêt, tandis qu'une forêt aléatoire sélectionne au hasard des observations / lignes et des

caractéristiques / variables spécifiques pour construire plusieurs arbres de décision à partir de puis fait la moyenne des résultats.

Importance :

L'importance de la caractéristique est calculée comme la diminution de l'impureté du nœud pondérée par la probabilité d'atteindre ce nœud. La probabilité de nœud peut être calculée par le nombre d'échantillons qui atteignent le nœud, divisé par le nombre total d'échantillons. Plus la valeur est élevée, plus la fonction est importante.

- Pour chaque arbre de décision en supposant seulement deux nœuds enfants (arbre binaire) --
→ (OFF or NOT) :

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)} \quad (8.1)$$

- $ni_{sub}(j)$ = l'importance du nœud j
- $w_{sub}(j)$ = nombre pondéré d'échantillons atteignant le nœud j
- $C_{sub}(j)$ = la valeur d'impureté du nœud j
- $left(j)$ = nœud enfant de gauche divisé sur le nœud j
- $right(j)$ = nœud enfant de la division droite sur le nœud j

Ensuite on va calculée Fi :

$$fi_i = \frac{\sum j:node\ j\ splits\ on\ feature\ i\ ni_j}{\sum k \in all\ nodes\ ni_k} \quad (8.2)$$

$fi_{sub}(i)$ = l'importance de la caractéristique i

$ni_{sub}(j)$ = l'importance du nœud j

-Ceux-ci peuvent ensuite être normalisés à une valeur comprise entre 0 et 1 en divisant par la somme de toutes les valeurs d'importance des caractéristiques :

$$normfi_i = \frac{fi_i}{\sum j \in all\ feature\ fi_j} \quad (8.3)$$

-L'importance finale de la caractéristique, au niveau de la forêt aléatoire, est sa moyenne sur tous les arbres. La somme de la valeur d'importance de l'entité sur chaque arbre est calculée et divisée par le nombre total d'arbres :

$$RFfi_i = \frac{\sum_{j \in \text{all tree}} normfi_{ij}}{T} \quad (8.4)$$

RFfi sub (i) = l'importance de la caractéristique i calculée à partir de tous les arbres du modèle Random Forest.

normfi sub (ij) = l'importance normalisée des caractéristiques pour i dans l'arbre j

T = nombre total d'arbres.

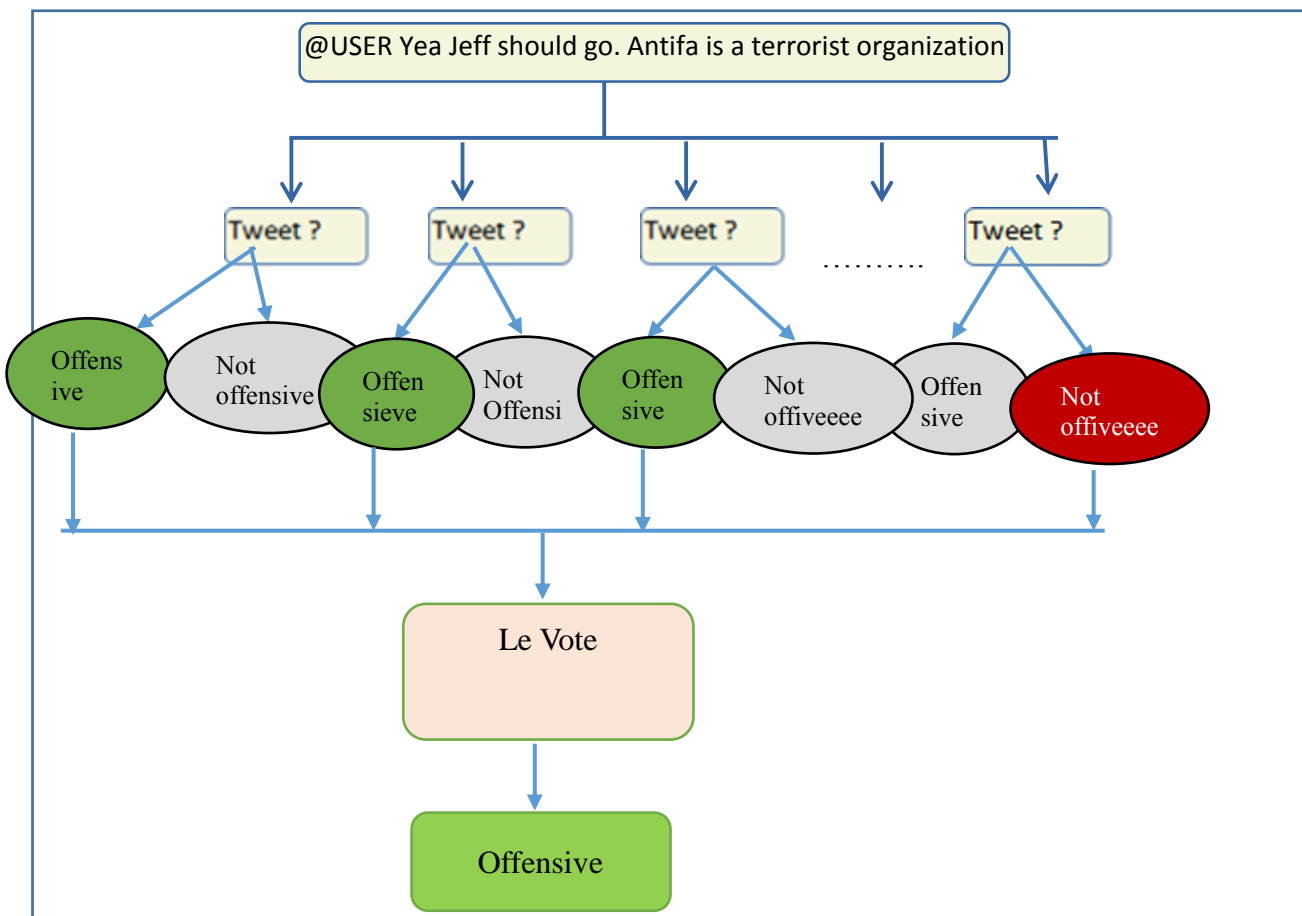


Figure 28 : Forêt aléatoire

III.4.1.5.2. Gradient Boosting Classificateur (GB)

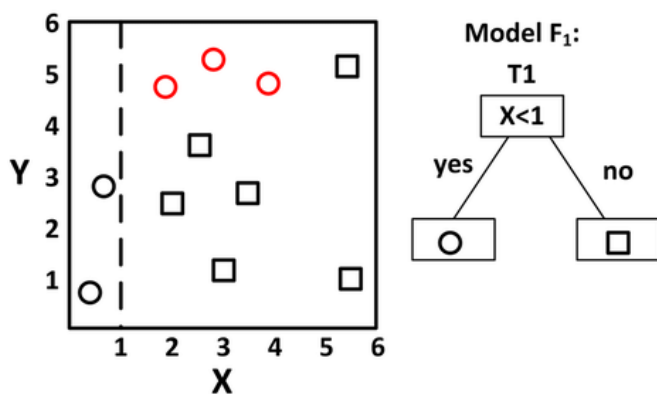
L'idée derrière le « gradient boosting » est de prendre une hypothèse faible ou un algorithme d'apprentissage faible et d'y apporter une série de modifications qui amélioreront la force de l'hypothèse / de l'apprenant. Ce type de stimulation d'hypothèse est basé sur l'idée d'apprentissage approximativement correct des probabilités [28]

Cet algorithme utilise le gradient de la fonction de perte pour le calcul des poids des individus lors de la construction de chaque nouveau modèle. Cela ressemble un peu à la descente de gradient pour les réseaux de neurones pour ceux qui connaissent

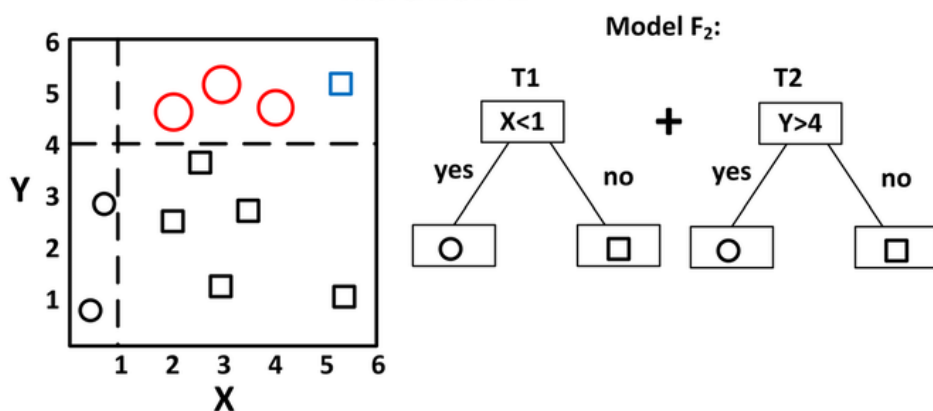
Le principe de gradient boosting

L'idée de base ressemble à celle du bagging. Plutôt que d'utiliser un seul modèle, nous en utilisons plusieurs que nous agrégeons ensuite pour obtenir un seul résultat. Dans la construction des modèles, le Boosting travaille de manière séquentielle. Il commence par construire un premier modèle qu'il va évaluer. A partir de cette mesure, chaque individu va être pondéré en fonction de la performance de la prédiction. L'objectif est de donner un poids plus important aux individus pour lesquels la valeur a été mal prédite pour la construction du modèle suivant. Le fait de corriger les poids au fur et à mesure permet de mieux prédire les valeurs difficiles. [29]

Iteration 1



Iteration 2



Iteration 3

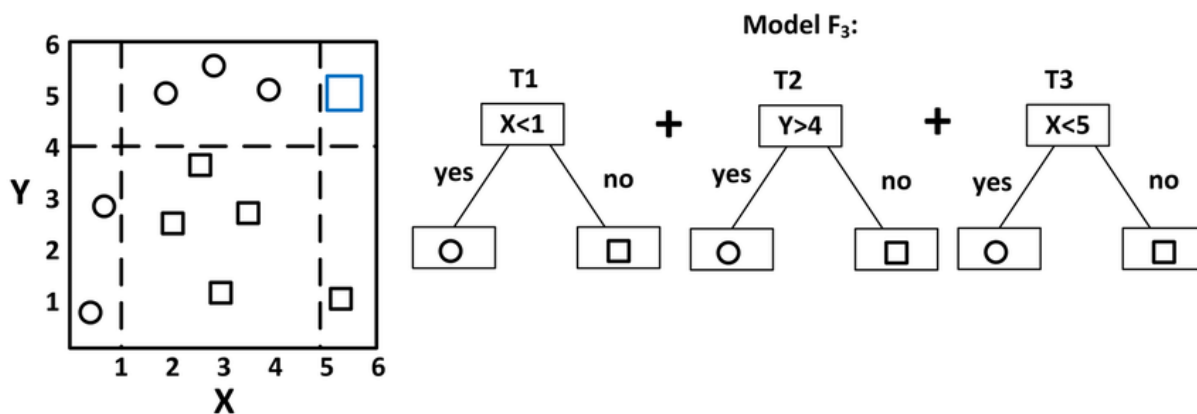


Figure 29 : Gradient Boosting (GB) Wikipedia

III.4.2. Les Algorithmes d'apprentissage profond

Dans cette partie nous avons travaillé avec deux algorithmes d'apprentissage profond (Réseaux de neurones récurrents (Simple RNN) et Réseaux de mémoire à long terme (LSTM)

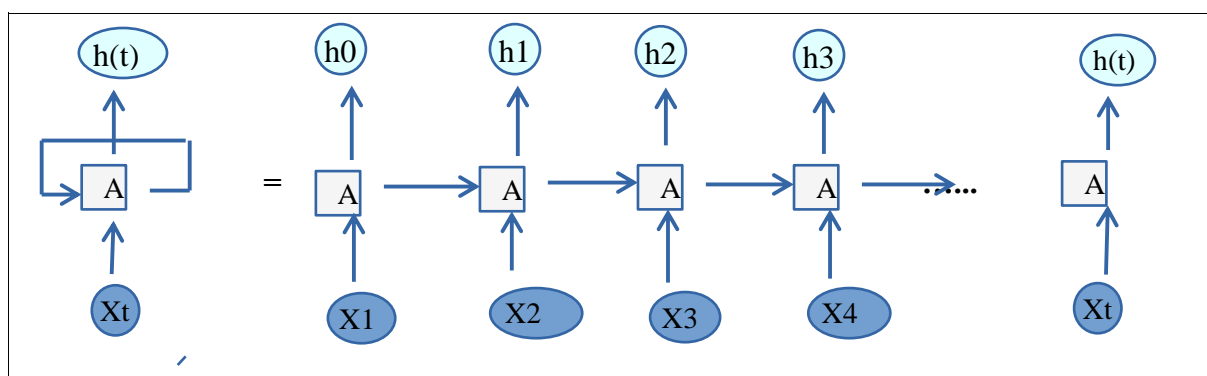
Les méthodes d'apprentissage en profondeur sont une mise à jour moderne des réseaux de neurones artificiels qui exploitent de nombreux calculs (un bon marché).

III.4.2.1. Réseaux de neurones récurrents (RNN)

Le réseau neurone récurrent est une généralisation du réseau neurone à réaction qui possède une mémoire interne. RNN est de nature récurrente car il remplit la même fonction pour chaque entrée de données tandis que la sortie de l'entrée actuelle dépend du dernier calcul. Après avoir produit la sortie, elle est copiée et renvoyée dans le réseau récurrent. Pour prendre une décision, il considère l'entrée actuelle et la sortie qu'il a apprise de l'entrée précédente (Toutes les entrées sont liées les unes aux autres).[30]

Nous avons travaillé avec récurrents neurones car il a des points d'avantages :

- RNN peut modéliser la séquence de données afin que chaque échantillon puisse être supposé dépendant des précédents
- Les réseaux de neurones récurrents sont même utilisés avec des couches convolutives pour étendre le voisinage effectif des pixels.



30 : Le processus du RNN

Exemple 1 : << @USER You must have an entire closet dedicated to college teams. >> Tout d'abord, il prend le $X(0)$ <YOU> de la séquence d'entrée, puis il sort $h(0)$, avec $X(1)$ <must>, est l'entrée pour l'étape suivante. Ainsi, $h(0)$ et $X(1)$ sont l'entrée pour

l'étape suivante. De même, h (1) du suivant est l'entrée avec X (2) <have> pour l'étape suivante et ainsi de suite. De cette façon, il se souvient du contexte pendant l'entraînement.

La formule de l'état actuel est

La formule de l'état actuel est $h_t = f(h_{t-1}; X_t)$ (4.1)

Application de la fonction d'activation : $h_t = \tanh(W_{hh}h_{t-1} + W_{xh}X_t)$ (4.2)

W est le poids

h est le vecteur caché unique

W_{hh} est le poids à l'état caché précédent

W_{hx} est le poids à l'état d'entrée actuel

tanh est la fonction d'activation qui implémente une non-linéarité qui écrase les activations à la plage [- 1.1]

Et production en final $Y_t = W_{hy}h_t$ (4.3)

Y_t est l'état de sortie

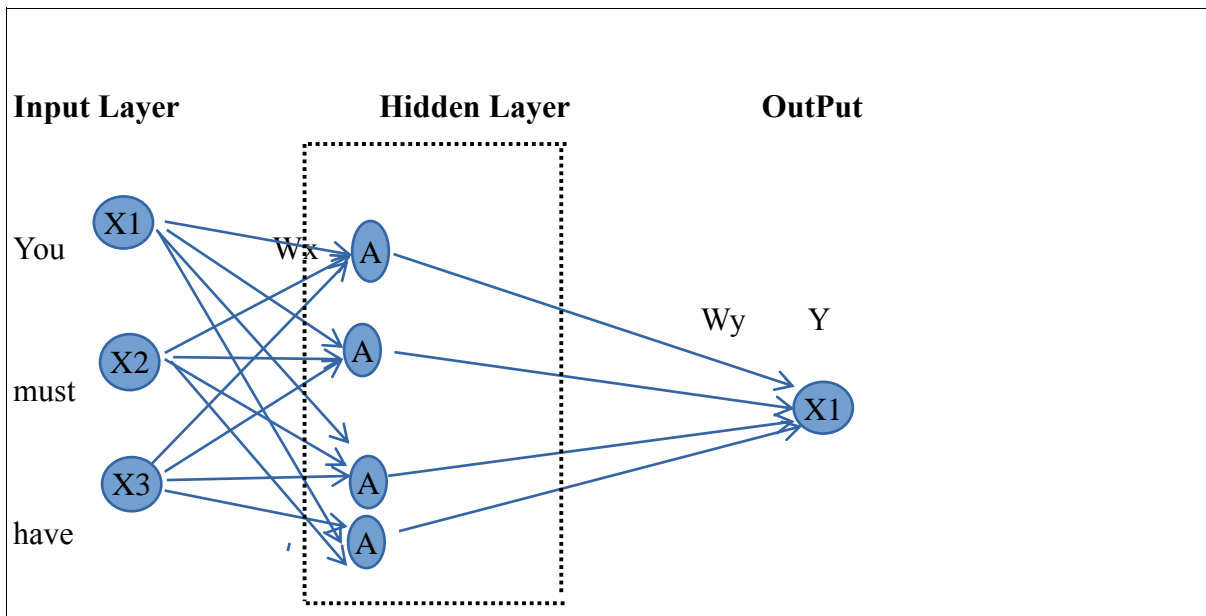


Figure 31 : Architecture simple de réseaux de neurones

Exemple 2 : «You must have »

Pour utiliser ce Tweet dans un RNN nous d'abord la convertir sous forme numérique (on peut travailler avec soit « one Hot vector » ou bien « Mot embedding »), dans notre cas nous avons travaillé avec « Mot embedding » pour convertir chaque mot en deux nombre.

Pour le <E1 et E2> nous prenons des valeurs aléatoire.

Mot	E1	E2
you	0.5	0.4
must	0.3	0.1
have	0.7	0.5

Tableau : Représentation du chaque mots en deux nombres

Maintenant, pour passer ces mots dans un RNN, nous traitons chaque mot comme Time-step embedding as features.

Model = Sequential ()

model.add(SimpleRNN(4, input_shape=(3, 2)))

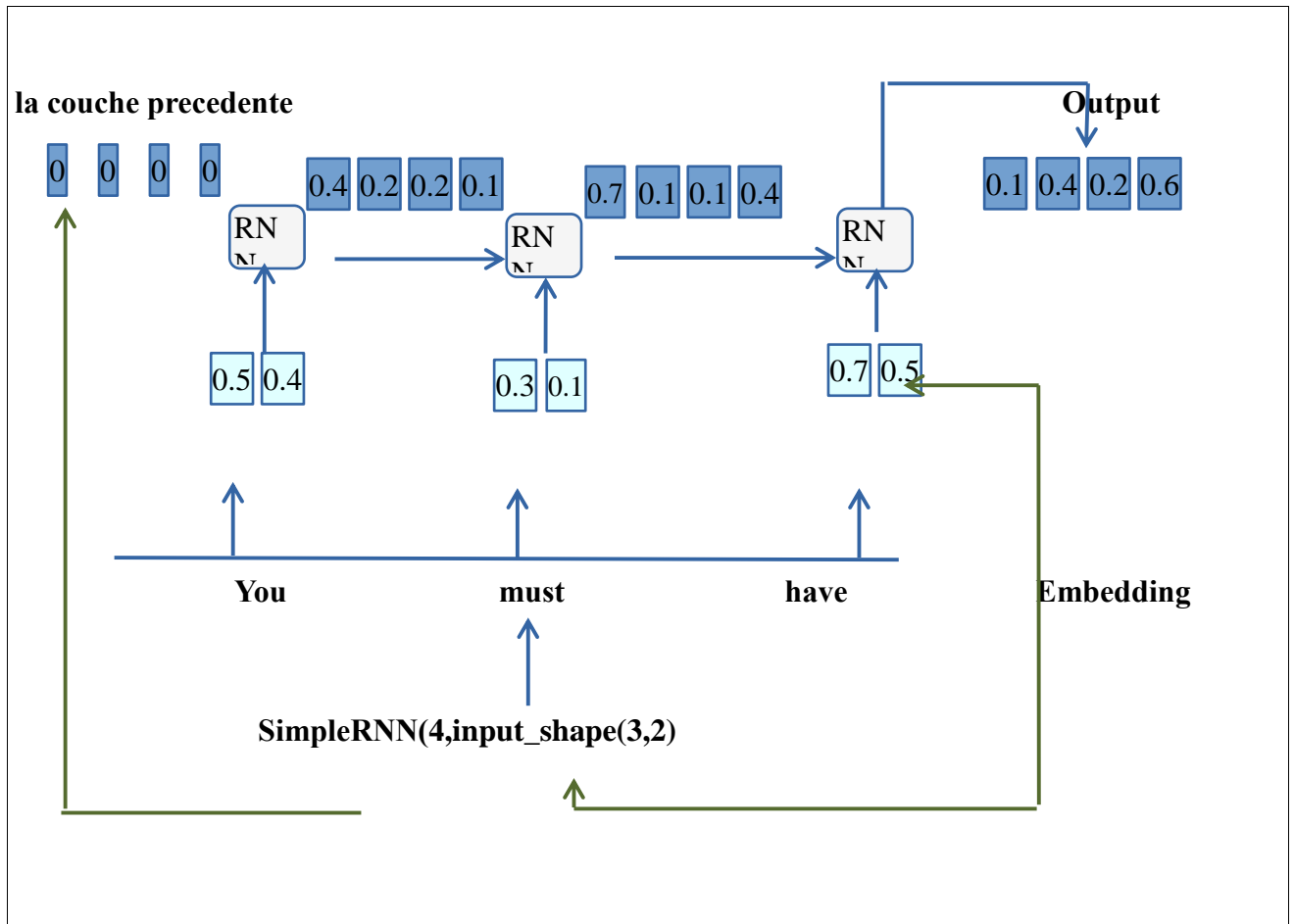


Figure 32 : Architecture pour le processus Du simple RNN

input_shape = (3,2):

- Nous avons 3 mots (you, must, have) Ainsi, le nombre time-step 3, et nous voyons donc 3 blocs sur la figure.
- Pour chaque mot, nous transmettons en Mot embedding de taille 2 au réseau.

SimpleRNN (4,...):

- Cela signifie que nous avons 4 unités dans la couche cachée.
- Pour le premier bloc, puisqu'il n'y a pas de sortie précédente, l'état caché précédent est défini sur [0, 0, 0, 0]

On peut dire que le réseau neurone récurrent n'est vraiment efficace sur tous les traitements (la taille des données) il a des points faibles comme la suite.

- le problème du gradient de disparition, dans lequel la performance du réseau neurone souffre du fait qu'il ne peut pas être formé correctement
- La formation (entraînement) d'un RNN est une tâche très difficile.
- Il ne peut pas traiter de très longues séquences si vous utilisez « tanh » ou relu comme fonction d'activation

A cause de ce point (le point faible en RNN) «longues séquences » nous avons passé à utiliser le modèle LSTM.

III.4.2.2. Long short Term memory (LSTM)

Les réseaux de mémoire à long terme (LSTM) sont une version modifiée des réseaux de neurones récurrents, ce qui facilite la mémorisation des données passées en mémoire. Le problème du gradient de fuite (disparition) de RNN est résolu ici. LSTM est bien adapté pour classer, traiter et prédire des séries temporelles en fonction de décalages temporels de durée inconnue. Il entraîne le modèle en utilisant la rétro-propagation. Dans un réseau LSTM, trois portes sont présentes.

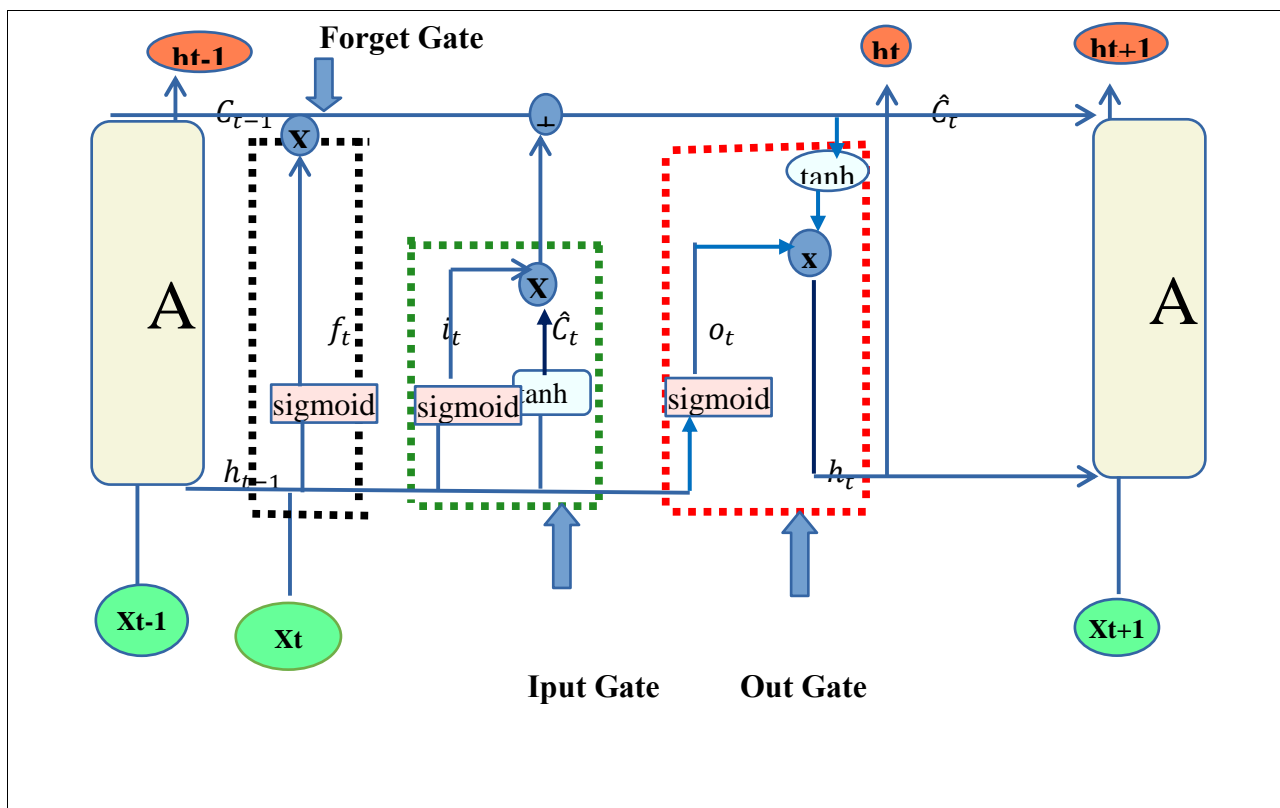


Figure 33 : Architecture représente les portes du LSTM

1. Input Gate (Porte d'entrée) : découvrez quelle valeur de l'entrée doit être utilisée pour modifier la mémoire. La fonction **sigmoïde** décide des valeurs à laisser passer par **0,1**. et la fonction **tanh** donne un poids aux valeurs qui sont passées en décidant de leur niveau d'importance allant de **-1** à **1**(plage **[-1, 1]**)

$$i_t = \delta(w_i \cdot [h_{t-1}, x_t] + b_i) \quad (5.1)$$

$$\hat{C}_t = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \quad (5.2)$$

2. Forget Gate (Oubliez la porte) : découvrez les détails à supprimer du bloc. Il est décidé par la fonction **sigmoïde**. Il regarde l'état précédent (h_{t-1}) et l'entrée de contenu (x_t) et sort un nombre entre **0** (**omit**) et **1** (**keep**) pour chaque nombre dans l'état de cellule $C_t - 1$.

$$f_t = \delta(w_f \cdot [h_{t-1}, x_t] + b_f) \quad (5.3)$$

3. Output Gate (Porte de sortie) : l'entrée et la mémoire du bloc sont utilisées pour décider de la sortie. La fonction **sigmoïde** décide des valeurs à laisser passer par 0,1. et la fonction **tanh** donne un poids aux valeurs qui sont passées en décidant leur niveau d'importance allant de -1 à 1 et multipliées par la sortie de **Sigmoïde**.

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad (5.4)$$

$$o_t = \delta(w_o \cdot [h_{t-1}, x_t] + b_o) \quad (5.5)$$

$$h_t = o_t * \tanh(C_t) \quad (5.6)$$

Exemple 1 :

On supposant que l'entrée comme suivante pour:

Fonction sigmoïde $[0.3 \ 0.7 \ 0.1] \rightarrow [S(0.3) \ S(0.7) \ S(0.1)]$ **output** $[0.57] \ 0.61 \ 0.52$

Fonction tanh $[0.3 \ 0.7 \ 0.1] \rightarrow [\tanh(0.3) \ \tanh(0.7) \ \tanh(0.1)]$ **output** $[0.29] \ 0.6 \ 0.01$

X c'est une multiplication.

+ C'est une addition.

Example 2:

<< You must have an entire >>

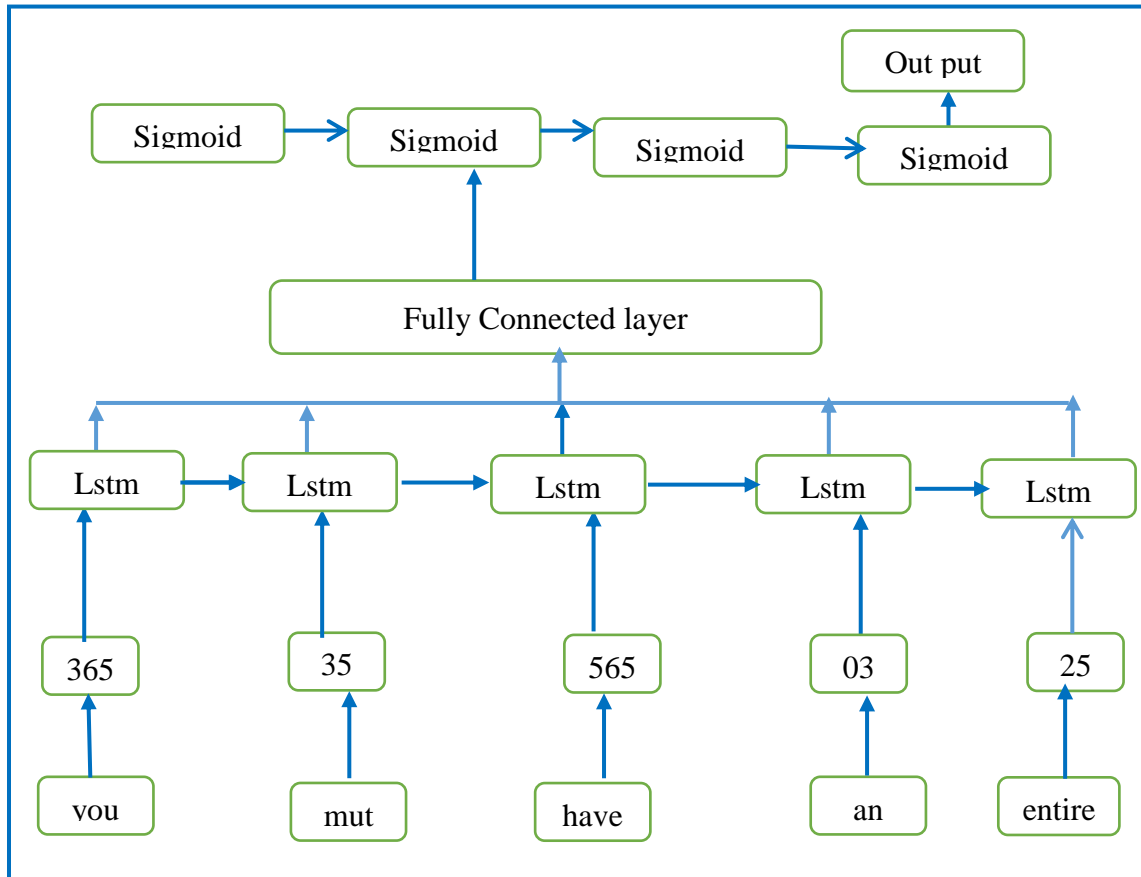


Figure 34 : exemple du LSTM avec sigmoïde activation

III.4.2.3.les fonctions d'activation

- **sigmoïde (logistique)** : utilisé en couche de sortie pour de la classification binaire. Intervalle de sortie $\{0,1\}$.
- **Tanh** : utilisé pour des LSTM pour des données en continue. Intervalle de sortie : $(-1,1)$.
- **Sotmax** : utilisé pour de la multi classification en couche de sortie. Intervalle de sortie $(-\infty;+\infty)$.
- **Relu** : ce sont les fonctions les plus populaires de nos jours, Très utilisé pour les CNN lambert[31] et les réseaux de multi perceptron[32]. Intervalle de sortie $(0;+\infty)$.

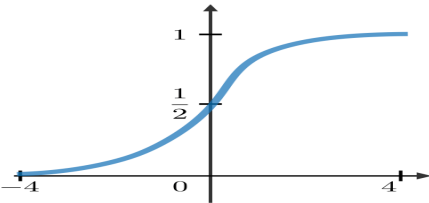
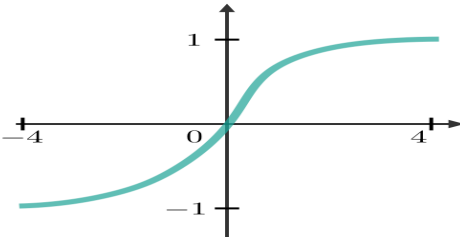
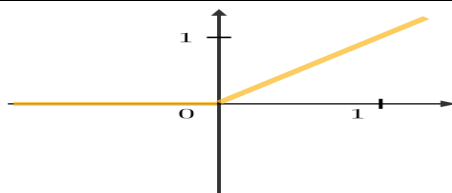
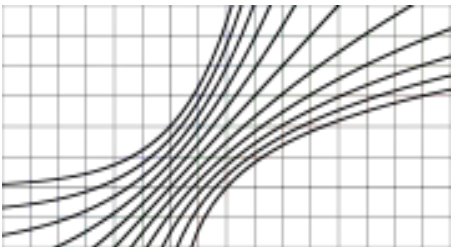
Nom	Graphe	Equation	Intervalle
Sigmoïde		$f(z) = \frac{1}{1+e^{-z}}$ (6.1)	[0, 1]
Tanh		$f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ (6.2)	[-1, 1]
Relu		$f(z) = \max(0, z)$ (6.3)	[0, +∞[
Softmax		$\delta(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}}$ (7)]-∞; +∞[

Table 11 : Les différentes fonctions d'activation

Fonction de « loss » :

Dans le contexte des réseaux de neurones récurrents, la fonction de « loss » (L) prend en compte le « loss » à chaque temps T de la manière suivante :

$$L(\hat{y}, y) = \sum_{t=1}^{T_y} L(\hat{y}^{<t>}, y^{<t>}) \quad (6.4)$$

« Backpropagation » temporelle :

L'étape de « backpropagation » est appliquée dans la dimension temporelle. À l'instant T , la dérivée du loss L par rapport à la matrice de coefficients W est donnée par :

$$\frac{\partial L^{(t)}}{\partial W} = \sum_{t=1}^T \frac{\partial L^{(t)}}{\partial W} |_{(t)} \quad (6.5)$$

III.5. les avantages et les inconvénients

Dans cette partie nous avons cité quelque points d'avantages et d'inconvénients pour les algorithmes d'apprentissage automatique et profond qui nous avons utilisées.

algorithme	L'avantage	l'inconvénient
Réseaux de neurones récurrents (Simple RNN)	<ul style="list-style-type: none"> • Possibilité de prendre en compte des entrées de toute taille • La taille du modèle n'augmente pas avec la taille de l'entrée • Les calculs prennent en compte les informations antérieures • Les coefficients sont indépendants du temps 	<ul style="list-style-type: none"> • Le temps de calcul est long • Difficulté d'accéder à des informations d'un passé lointain • Impossibilité de prendre en compte des informations futures un état donné
Long Short term memory (LSTM)	<ul style="list-style-type: none"> • LSTM est un excellent outil pour tout ce qui a une séquence utilisé pour génération du texte par exemple. • L'algorithme du LSTM fait la mise à jour par les poids et le time step est important. • la propagation constante des erreurs dans les cellules mémoire permet à lstm de remonter de très longs délais 	<ul style="list-style-type: none"> • Les LSTM prend plus de temps à se former • Les LSTM nécessitent plus de mémoire pour s'entraîner • Les LSTM sont faciles à overfitting Le droupout est beaucoup plus difficile à mettre en œuvre dans les LSTM • Les LSTM sont sensibles à différentes initialisations de poids aléatoires

	en cas de problème similaire.	
Arbre de décision (DT)	<ul style="list-style-type: none"> • Simple à comprendre et lisible Facilement interprétable • Robuste au bruit : peut prendre en compte tous les type d'attribut • Performent sur de grands jeux de données. 	<ul style="list-style-type: none"> • Risque de l'overfiting • La modification d'un seul nœud s'il est près du sommet modifie entièrement l'arbre
Support vecteur machine(Lsvc)	<ul style="list-style-type: none"> • N'a pas besoin de beaucoup d'entraînement. 	<ul style="list-style-type: none"> • Considéré comme lente
Logistique régression (RL)	<ul style="list-style-type: none"> • Robuste et n'a pas de paramètres configuration de règles • -largement connu et compréhensible 	<ul style="list-style-type: none"> • Nécessite des échantillons de grande taille pour atteindre un bon niveau de stabilité.
Naïve bayésien	<ul style="list-style-type: none"> • Très simple d'utilisation • Bon résultat lorsque les données est gros 	Très sensible à leur corrélation
Forêt aléatoire (RF)	<ul style="list-style-type: none"> • Plus rapide qu'arbre de décision • Meilleurs résultats qu'arbre de décision • Fonctionne efficacement sur de grandes bases de données • - Il dispose de méthodes pour 	<ul style="list-style-type: none"> • Risque d'overfiting

	<p>équilibrer les erreurs dans l'ensemble de données non équilibrés.</p>	
<p>Plus proche voisin (KNN)</p>	<ul style="list-style-type: none"> • Simple à concevoir • Flexible dans les cas de traitements non-linéaire séparables. 	<ul style="list-style-type: none"> • Sensible aux bruits • Très coûteux si le nombre de variables prédictives est très grand
<p>Gradient boosting(GB)</p>	<ul style="list-style-type: none"> • La méthode de gradient boosting généralement donne des bons résultats. 	<ul style="list-style-type: none"> • GBoost est plus difficile

Table 12 : les avantages et les inconvénients de différents algorithmes d'apprentissage

III.6. Conclusion

Dans ce chapitre nous avons présenté l'architecture de notre système avec les techniques et les méthodes de prétraitement ainsi que les représentations vectorielles par <mot, caractère, caractère N-gramme et tous> et la modélisation de chaque processus de notre système d'identification des textes offensants.

Dans la première section nous avons décrit l'architecture de notre système avec ses composants, ensuite nous avons abordé à la phase de prétraitement des données et les techniques de préparations des données avec les représentations vectorielles.

Après la préparation des données nous avons passé à l'étapes de modélisations à travers les méthodes et les algorithmes d'apprentissage automatique machine et deep Learning.

-Dans le chapitre suivant nous allons présenter l'implémentation et l'évaluation des résultats obtenus par notre classificateur choisi. les résultats de représentation vectorielles, les langages de programmation, les outils utilisés et l'application qui représente notre travail.

Chapitre 4

Les résultats obtenus et présentation de l'application

III.1. Introduction

Dans ce chapitre, nous allons présenter l'implémentation de notre système. Nous commençons tout d'abord par la présentation des langages de programmation et l'environnement de développement et les outils utilisés, ensuite nous allons présenter les résultats obtenus par les représentation vectorielles et ainsi les résultats et l'évaluation des algorithmes d'apprentissage automatique et profond avec la conclusion du meilleur modèle et à la fin passer à la représentation de notre application et le modèle que nous avons utilisé.

III.2. L'environnement du développement

Nous avons présenté dans cette section, le langage de programmation python utilisé et son environnement anaconda (spyder IDE, vs studio IDE) avec ces outils.

III.2.1. Software

➤ Python

Python a été conceptualisé à l'origine par Guido van Rossum à la fin des années 1980 en tant que membre de l'Institut national de recherche en mathématiques et en informatique. Initialement, il a été conçu comme une réponse au langage de programmation ABC qui a également été mis au premier plan aux Pays-Bas. Parmi les principales caractéristiques de Python par rapport au langage ABC, il y avait le fait que Python avait une gestion des exceptions et était ciblé pour le système d'exploitation Amoeba.

Fait amusant. Python n'est pas nommé d'après le serpent. Il porte le nom de l'émission de télévision britannique Monty Python.

Bien sûr, Python, comme d'autres langages, est passé par un certain nombre de versions. Python 0.9.0 a été publié pour la première fois en 1991. En plus de la gestion des exceptions, Python comprenait des classes, des listes et des chaînes. Plus important encore, il comprenait lambda, mapper, Ce qui l'alignait fortement par rapport à la programmation fonctionnelle.

En 2000, Python 2.0 est sorti. Cette version de était davantage un projet open-source de membres de l'Institut national de recherche en mathématiques et en informatique. Cette version de Python incluait des compréhensions de liste, un ramasse-miettes complet et supportait Unicode.

Python 3.0 était la prochaine version et a été publié en décembre 2008 (la dernière version de Python est la 3.6.4). Bien que Python 2 et 3 soient similaires, il existe des différences subtiles. Le plus remarquable est peut-être le fonctionnement de l'instruction print, car dans Python 3.0, l'instruction print a été remplacée par une fonction print (). [34]

- Python est le langage de programmation le plus utilisé dans le domaine du Machine Learning, du Big Data et de la Data Science.



Figure 35 : Logo du python

➤ **Anaconda**

Anaconda est une distribution gratuite et open-source des langages de programmation Python et R pour le calcul scientifique (science des données, applications d'apprentissage automatique, traitement de données à grande échelle, analyse prédictive, etc.), qui vise à simplifier la gestion des paquets et déploiement. La distribution comprend des packages de science des données adaptés à Windows, Linux et macOS. Il est développé et maintenu par Anaconda, Inc., qui a été fondée par Peter Wang et Travis Oliphant en 2012. En tant que produit Anaconda, Inc., il est également connu sous le nom d'Anaconda Distribution ou Anaconda Individual Edition, tandis que les autres produits de la société sont Anaconda Team Edition et Anaconda Enterprise Edition, qui ne sont pas gratuits.

Les versions de package dans Anaconda sont gérées par le système de gestion de package conda . Ce gestionnaire de paquets a été créé comme un paquet open-source séparé car il a fini par être utile seul et pour d'autres choses que Python. Il existe également une petite version bootstrap d'Anaconda appelée Miniconda , qui ne comprend que conda, Python, les packages dont ils dépendent et un petit nombre d'autres packages.[35]

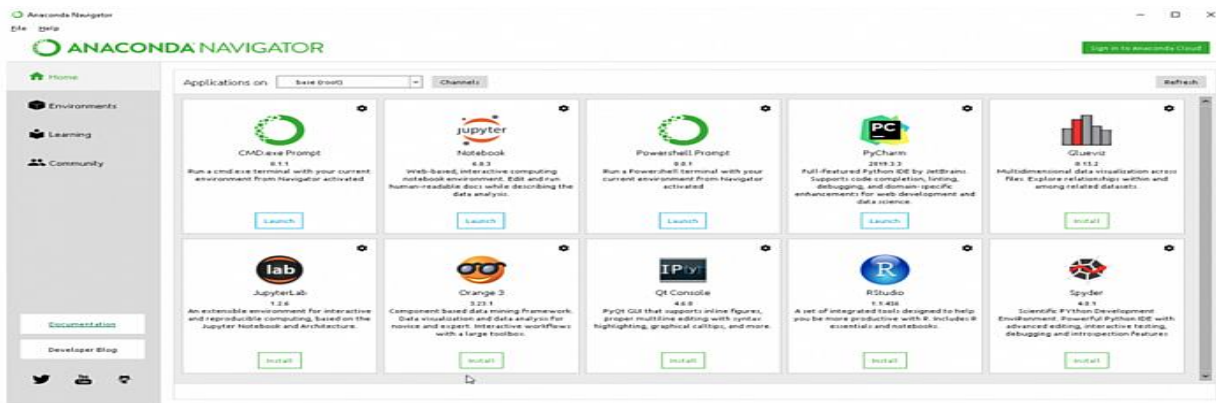


Figure 36 : l'environnement de anaconda

➤ Spyder

Spyder créé et développé par Pierre Raybaut en 2009, depuis 2012 Spyder a été maintenu et continuellement amélioré par une équipe de développeurs scientifiques Python et la communauté.

Spyder est un environnement de développement intégré (IDE) multiplateforme open source pour la programmation scientifique en langage Python. Spyder s'intègre à un certain nombre de packages importants de la pile scientifique Python, notamment NumPy , SciPy , Matplotlib , pandas, IPython , SymPy et Cython , ainsi que d'autres logiciels open source. Spyder est extensible avec des plugins propriétaires et tiers, inclut la prise en charge d'outils interactifs pour l'inspection des données et intègre des instruments d'assurance qualité et d'introspection de code spécifiques à Python, tels que Pyflakes, Pylint et Rope. Il est disponible multiplateforme via Anaconda, sur Windows, sur macOS via MacPorts et sur les principales distributions Linux telles que Arch Linux, Debian, Fedora , Gentoo Linux, openSUSE et Ubuntu.[36]

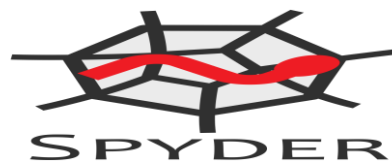



Figure 37 : Logo du spyder IDE

➤ Bash Scripting

Bash est un "shell Unix" : une interface de ligne de commande pour interagir avec le système d'exploitation. Il est largement disponible, étant le shell par défaut sur de nombreuses distributions GNU / Linux et sur Mac OSX, avec des ports existants pour de nombreux autres systèmes. Il a été créé à la fin des années 1980 par un programmeur nommé Brian Fox, travaillant pour la Free Software Foundation . Il était conçu comme une alternative logicielle libre au shell Bourne (en fait, son nom est un acronyme pour Bourne Again SHell), et il intègre toutes les fonctionnalités de ce shell, ainsi que de nouvelles fonctionnalités telles que l'arithmétique des entiers et le contrôle des tâches.[37]



```
Activities Terminal ven. 16:53 lenovo@gaziz: ~  
File Edit View Search Terminal Help  
(base) lenovo@gaziz:~$ python textClassification.py -tr trainSubTask_a.xlsx -trclm "tweet, subtask_a" -tst testSubTask_a.xlsx -tstclm "tweet, subtask_a" -vec all > outExp115.log
```

➤ HTML

HTML signifie «Hyper Text Markup Language» qu'on peut traduire par «langage de balises pour l'hypertexte». Il est utilisé afin de créer et de représenter le contenu d'une page web et sa structure. D'autres technologies sont utilisées avec HTML pour décrire la présentation d'une page Css et/ou ses fonctionnalités interactives JavaScript.

Figure 38 : Exemple du batshScript

HTML fonctionne grâce à des «balises» qui sont insérées au sein d'un texte normal. Chacune de ces balises indique la signification de telle ou telle portion de texte dans le site. On parle d'«hypertexte» en référence aux liens qui connectent les pages web entre elles. C'est la mécanique originelle du «World Wide Web» que nous connaissons aujourd'hui. En écrivant et publiant des pages web, vous devenez un acteur du Web dès que votre site est accessible en ligne.[38]



Figure 39 : Logo de langage de balise HTML

➤ CSS

CSS est l'un des langages principaux du Web ouvert et a été standardisé par le W3C. Ce standard évolue sous forme de niveaux (levels), CSS1 est désormais considéré comme obsolète, CSS2.1 correspond à la recommandation et CSS3, qui est découpé en modules plus petits, est en voie de standardisation [39], CSS est le langage que nous utilisons pour styliser un document HTML.



Figure 40 : Logo de langage CSS

➤ JavaScript

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les technologies HTML et CSS, JavaScript est parfois considéré comme l'une des technologies cœur du World Wide Web. Une grande majorité des sites web l'utilisent, et la majorité des navigateurs web disposent d'un moteur JavaScript dédié pour l'interpréter, indépendamment des considérations de sécurité qui peuvent se poser le cas échéant [40]

- JavaScript est le langage de programmation le plus populaire au monde.
- JavaScript est le langage de programmation du Web.
- JavaScript est facile à apprendre.



Figure 41 : Logo de JavaScript

➤ **Firestore**

Firestore est une plateforme développée par Google pour créer des applications mobiles et Web.

C'était à l'origine une société indépendante fondée en 2011. En 2014, Google a acquis la plateforme et c'est maintenant leur offre phare pour le développement d'applications.

Firestore est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++...). Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Facebook, Twitter et Github), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel. Lancé en 2011 sous le nom d'Envolve, par Andrew Lee et par James Templin, le service est racheté par Google en 2014. Il appartient aujourd'hui à la maison mère de Google : Alphabet. [41] , nous avons utilisé Firestore comme en-placement de stockage pour stocker et récupérer les coordonnées d'utilisateur.



Figure 42 : Logo Fire Base

➤ **Angular**

Conçu par Google et dont la version 1 est sortie en 2012, Angular s'inscrit définitivement dans une approche moderne de la création d'applications web «One page» et d'interfaces utilisateurs. Orienté objet, ce Framework MVC exploite le langage Typescript (langage compilé qui va générer du Javascript). Avec Angular, la logique applicative est insérée directement dans le HTML par l'intermédiaire d'éléments ou d'attributs (principe du data-binding). Il est également possible, Quelques avantages intéressants avec Angular JS [42]

- Applications web devenues radicalement plus légères côté serveur
- Synchronisation automatique des données pour une meilleure gestion dynamique des contenus
- Une plateforme modulaire
- Framework maintenu très régulièrement et en constante évolution
- Bénéficie d'une très importante communauté Services utilisant la technologie Angular

Parmi les nombreux sites et services en ligne développés avec Angular / Angular JS, nous retrouvons :

- [ABC News](#)
- [Project FI](#)
- [AT&T Community Forums](#)
- [Bellagio](#)
- [Express Google](#)
- [Microsoft Customer](#)

Angular est un cadre de conception d'applications et une plate-forme de développement pour créer des applications d'une seule page efficaces et sophistiquées.

Nous avons utilisé Angular pour la consommation de notre API.



Figure 43 : Logo du framework Angular

Scikit-learn:

Est une bibliothèque python destinée à l'apprentissage automatique. Elle est développée par de nombreux contributeurs notamment dans le monde académique par des instituts français d'enseignement supérieur et de recherche comme inria . Elle comprend notamment des fonctions pour estimer des forêts aléatoires, des régressions logistiques, des algorithmes de classification, et les machines à vecteurs de support. Elle est conçue pour s'harmoniser avec d'autres bibliothèques libres python notamment NumPy et Scipy. [43]

Keras :

La bibliothèque Keras permet d'interagir avec les algorithmes de réseaux de neurones profonds et d'apprentissage automatique , notamment Tensorflow, Theano, Microsoft Cognitive Toolkit ou PlaidML.

Conçue pour permettre une expérimentation rapide avec les réseaux de neurones profonds, elle se concentre sur son ergonomie, sa modularité et ses capacités d'extension. Elle a été développée dans le cadre du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System)⁵. Elle a été initialement écrite par François Chollet [44]

Google Colab :



Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur. [45]

Jupyter Notebook:

Jupyter Notebook est une application Web Open Source permettant de créer et de partager des documents contenant du code (exécutable directement dans le document), des équations, des images et du texte. Avec cette application il est possible de faire du traitement de données, de la modélisation statistique, de la visualisation de données, du Machine Learning, etc. Elle est disponible par défaut dans la distribution Anaconda .[45]

flask

Est un Framework Web créé initialement par Armin Ronacher comme étant un poisson d'avril. Le souhait de Ronacher était de réaliser un Framework web contenu dans un seul fichier python mais pouvant maintenir des applications très demandées. Flask vous fournit des outils, des bibliothèques et des technologies qui vous permettent de créer une application Web. Cette application Web peut être des pages Web, un blog, un wiki ou devenir aussi grande qu'une application de calendrier Web ou un site Web commercial. En 1^{er} Avril 2010, la première version a été publiée et la dernière version 1.1.2 est publiée le 30 Avril 2020.[46]



III.2.2. HardWare

Nous avons utilisé une machine (Laptop) qui possède de ces composantes suivantes :

<i>Composante</i>	<i>Caractéristique</i>
Memory	7,6GiB
Processor	Intel® Core™ i5 CPU M 540 @ 2.53GHz × 4
Graphics	Intel® Ironlake Mobile
Gnome	3.28.2
OS Type	Ubuntu 18.04.1 LTS 64-bit
Disk	491,2 GB

Table 13 : Représente les Caractéristiques du hardware

III.3. Description du système

Notre système contient trois sous tâches principales qui sont : sous tâche_a , sous tâche_b et sous tâche_c et sous tâche_arabic pour la langue arabe. pour développer ou bien construire un modèle nous avons utilisé plusieurs outils et avant tous nous avons procédé par un prétraitement pour les deux langues.

Notre système est construit par des logiciels existant dans notre « lap top » ou bien en utilisant Cloud (Google colab) dans le cas d'apprentissage automatique profond (simple RNN et Lstm), le travail c'est de développer une application basé sur un modèle d'apprentissage qui permet de prédire la cible.

III.3.1. Prétraitement

Comme nous avons vu au chapitre 3 nous avons implémenté notre prétraitement en utilisant les techniques et des fonctions que nous avons créées du prétraitement (par exemple algorithme pour les « stopWords »)

```
def removeStopWord(text, lang):  
stop = stopMots.Words(lang)  
out = text.apply(lambda x: ''.join([Words for Word in x.split() if Word not in (stop)]))  
return out
```

Et nous avons utilisés différents bibliothèques pour charger et traité chaque sous tâches (NLTK RE ...etc.) Nous commençons tout d'abord par les prétraitements des sous tâches anglais ou la première étape est la normalisation (rendre le texte en minuscule tout d'abord et la suppression des blocs vides), en suite « remove ponctuation », les « stop Mot » , les lemmatisations, postag. ...etc., et ensuite nous avons répété le même traitement avec l'arabe

III.3.2. Les métriques d'évaluation :

III.3.2.1. Classification binaire <binary>

Dans cette étape nous allons montrer comment évaluer les performances d'un modèle via les métriques de précision, de rappel et de F1 score dans ML avec une brève explication des «métriques de confusion». Dans cette expérience, nous avons travaillé avec huit algorithmes boostés à deux classes ou bien 3 comme sous tâche _C et notre objectif est de prédire le type de la cible.

Une fois que nous avons construit notre modèle, la question la plus importante qui se pose est de savoir quelle est la qualité de notre modèle ? , Ainsi, l'évaluation de notre modèle est la tâche la plus importante du projet de l'identification des textes offensants, qui définit la qualité de nos prédictions.

➤ Accuracy

C'est une mesure de performance la plus intuitive et il s'agit simplement d'un rapport entre les observations correctement prédites et les observations totales. On peut penser que si nous avons une grande précision, notre modèle est le meilleur. Oui, la précision est une excellente mesure, mais uniquement lorsque vous avez des ensembles de données symétriques où les valeurs des faux positifs et des faux négatifs sont presque les mêmes. $\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$ (10.1)

➤ Précision

La précision est le rapport entre les observations positives correctement prédites et le total des observations positives prévues. $\text{précision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$ (10.2)

➤ Rappel

C'est le rapport entre les observations positives correctement prédites et toutes les observations de la classe réelle. $\text{Rappel} = \frac{\text{TP}}{\text{TP} + \text{FN}}$. (10.3)

➤ F1 Score

C'est la moyenne pondérée de la précision et du rappel. Par conséquent, ce score prend en compte à la fois les faux positifs et les faux négatifs. Intuitivement, ce n'est pas aussi facile à comprendre que la précision, mais F1 est généralement plus utile que la précision, surtout si vous avez une distribution de classe inégale. La précision fonctionne mieux si les faux positifs et les faux négatifs ont un coût similaire. Si le coût des faux positifs et des faux négatifs est très différent, il vaut

mieux regarder à la fois la précision et le rappel. $F1 \text{ Score} = 2 * (\text{Rappel} * \text{Précision}) / (\text{Rappel} + \text{Précision})$ (10.4)

- **True positives (TP):** Ce sont les valeurs positives correctement prédites, ce qui signifie que la valeur de la classe réelle est oui et la valeur de la classe prédite est également oui.
- **True negatives (TN):** Ce sont les valeurs négatives correctement prédites, ce qui signifie que la valeur de la classe réelle est non et que la valeur de la classe prédite est également non.
- **False positives (FP):** Lorsque la classe réelle est non et que la classe prédite est oui.
- **False negatives (FN):** Lorsque la classe réelle est oui mais que la classe prédite est non.

Nous allons faire un petit exemple pour sous tache A

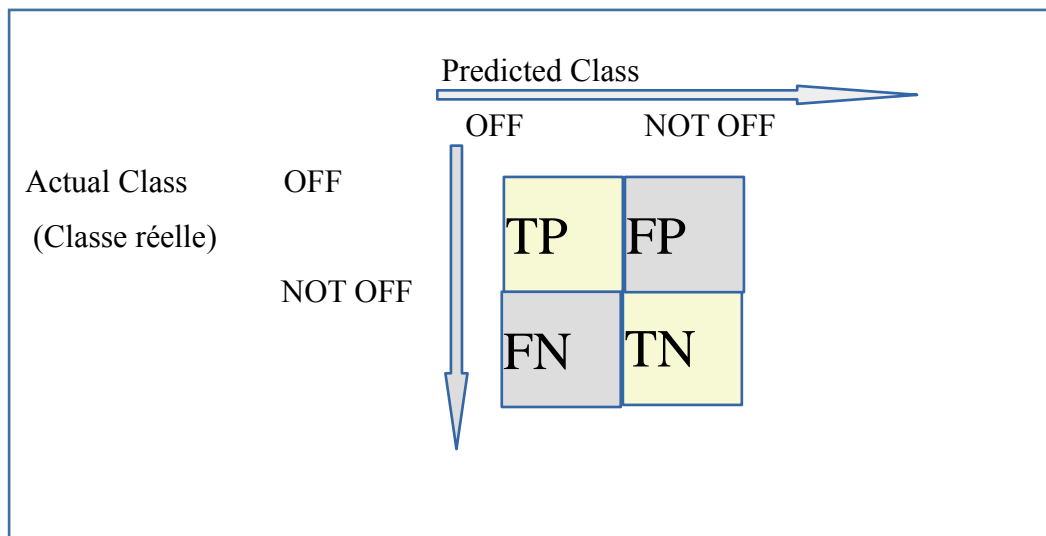


Figure 45 : Classe réelle et classe de prédiction

Dans un problème de classification déséquilibré avec plus de deux classes (dans nos cas sous tache B et C), le rappel est calculé comme la somme des true positives dans toutes les classes divisée par la somme des true positives et des fausses négatives dans toutes les classes.

III.3.2.2. Classification multi-classes

$\text{Rappel} = \frac{\text{Somme } c \text{ en } C (\text{TruePositives}_c)}{\text{Somme } c \text{ en } C (\text{TruePositives}_c + \text{FalseNegatives}_c)}$.

III.3.2.3 sous tache _A

Nous avons travaillé avec trois sous taches pour chaque sous tache on a fait le même travail et même étapes

Pour sous tache A on a séparé les étapes d'exécution par des stages (six stages) pour chaque stages il y'a des expériences d'autres manière (par exemple pour le stage un nous aurons 4 expériences <all, Mot, caractère et caractère n-gramme)

➤ Stage zéro

Comme nous avons vus en chapitre trois, le stage zéro nous aurons quatre expériences la première expérience montre les valeurs des modèles lorsque on travaille seulement sur les Mots et la deuxième expérience

Lorsqu'on prend le traite avec les caractèresetc.

Le tableau en dessous monte les scores (F mesure) des modèles avec chaque phase.

Mot		Caractère		Caractère n-gramme		Tous	
Modèle	Score	Modèle	Score	Modèle	Score	Modèle	Score
BNB	0.653	BNB	0.660	BNB	0.674	BNB	0.671
MNB	0.662	MNB	0.657	MNB	0.661	MNB	0.660
DT	0.666	DT	0.668	DT	0.681	DT	0.662
KNN	0.678	KNN	0.655	KNN	0.647	KNN	0.659
RF	0.705	RF	0.723	RF	0.724	RF	0.717
LR	0.708	LR	0.754	LR	0.757	LR	0.768
GB	0.727	GB	0.742	GB	0.746	GB	0.745
LSVC	0.747	LSVC	0.758	LSVC	0.768	LSVC	0.765

Table 14 : Les scores avant le prétraitement pour sous tache _A stage zéro

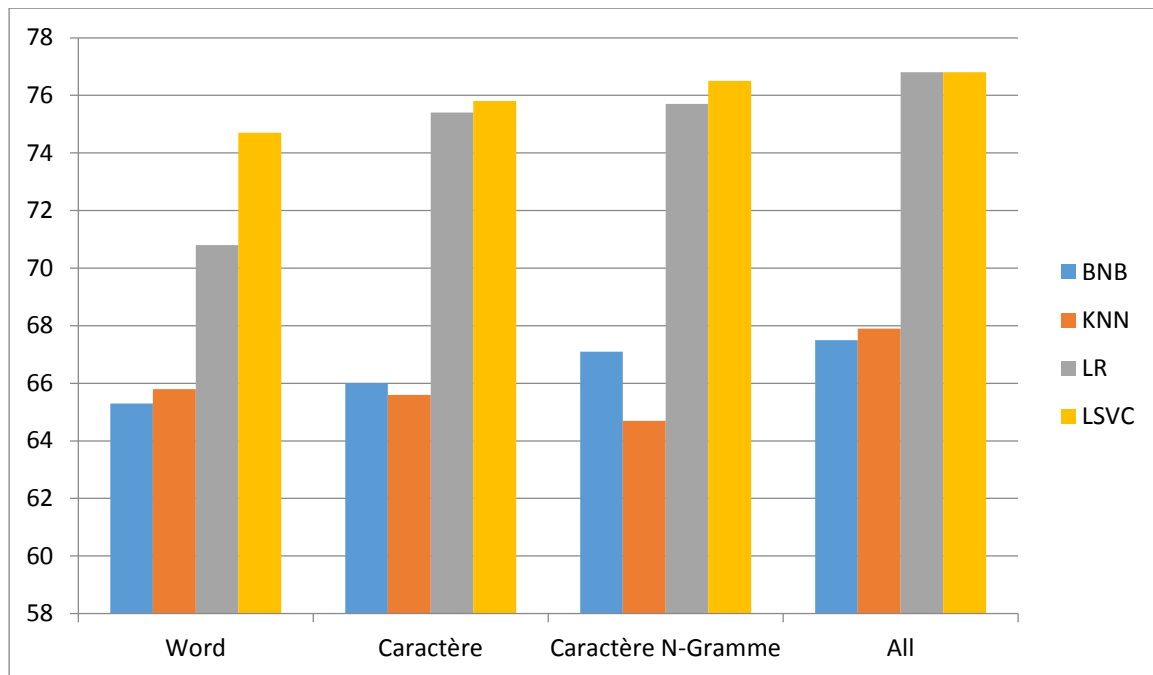


Figure 46 : Histogramme sous tache_A

D'après le tableau précédant nous avons remarqué que tous les résultats se situent entre un intervalle [0.64 ,0.76] et le meilleur classificateur est LinearSvc pour les trois phases

< Mot, caractère, caractère N-gramme>a une valeur atteindre a $\approx 76,8\%$ et pour la phase <all> c'est le classificateur LR qui est atteindre une valeur maximale $\approx 76,8\%$ quand a la valeur la plus basse c'est pour le classificateur KNN dans les trois phases <all, caractère N-gramme et caractère > proche a

$\approx 64,7\%$ et la valeur $65,3\%$ pour BNB dans la phase <Mot>

➤ Stage 1

Remove stopMot							
Mot		Caractère		Caractère N-gramme		Tous	
Modèle	Score	Modèle	Score	Modèle	Score	Modèle	Score
BNB	0.653	BNB	0.683	BNB	0.686	BNB	0.678
MNB	0.667	MNB	0.658	MNB	0.663	MNB	0.662
DT	0.684	DT	0.674	DT	0.682	DT	0.691
KNN	0.686	KNN	0.679	KNN	0.680	KNN	0.677
LR	0.719	LR	0.756	LR	0.761	LR	0.770
LSVC	0.754	LSVC	0.768	LSVC	0.759	LSVC	0.766

Table 15 : Résultats des classificateurs après remove stopMot sous tache_A stage

Remove punctuation							
Mot		Caractère		Caractère N-gramme		Tous	
Modèle	Score	Modèle	Score	Modèle	Score	Modèle	Score
BNB	0.653	BNB	0.660	BNB	0.674	BNB	0.671
MNB	0.662	MNB	0.657	MNB	0.661	MNB	0.660
DT	0.667	DT	0.677	DT	0.692	DT	0.663
KNN	0.678	KNN	0.655	KNN	0.647	KNN	0.659
LR	0.708	LR	0.754	LR	0.757	LR	0.768
LSVC	0.747	LSVC	0.758	LSVC	0.768	LSVC	0.765

Table 16 : Résultats des classificateurs après remove punctuation sous tache _A stage 1

Pos Tag							
Mot		Caractère		Caractère N-gramme		Tous	
Modèle	Score	Modèle	Score	Modèle	Score	Modèle	Score
BNB	0.653	BNB	0.660	BNB	0.674	BNB	0.671
MNB	0.662	MNB	0.658	MNB	0.661	MNB	0.660
DT	0.671	DT	0.674	DT	0.691	DT	0.657
KNN	0.678	KNN	0.679	KNN	0.647	KNN	0.659
LR	0.708	LR	0.756	LR	0.757	LR	0.768
LSVC	0.747	LSVC	0.768	LSVC	0.768	LSVC	0.765

Table 17 : Résultats des classificateurs après remove stopMot sous tache _A stage 1

D'après les trois tableaux (pour la suppression des mots vides, les ponctuations et l'utilisation des « pos tag ») nous avons remarqué que les résultats qu'on a obtenus après la suppression des mots vides ont valeurs supérieur que les autres tableaux, donc nous concluons que la suppression des stops Mot a une grande importance lors du traitement textuelle.

➤ Stage 4

Dans un tableau suivant on va présenter deux classificateur <LSVC et LR> avec la suppression des mots vide et les ponctuations, et nous avons appliqué aussi <stem,pos Tag> Avec toutes les phases <Mot, caractère, caractère N-gramme, all>

Remove (pouctuation , stopWord) et appliquer (stem , pos tag)							
Mot		Caractère		Caractère N-gramme		Tous	
Modèle	Score	Modèle	Score	Modèle	Score	Modèle	Score
LSVC	0.754154	LSVC	0.768505	LSVC	0.759819	LSVC	0.766616
LR	0.719789	LR	0.756042	LR	0.761707	LR	0.770015
Moyenne	0.7369715	0.7622735		0.7607630000000001		0.7683154999999999	

Table 18 : Résultats des classificateurs pour sous tache _A stage 4

D'après tous ce qu'on a vu (les tableaux précédant) dans sous tache _A nous avons remarqué que les deux classificateurs (LR, LSVC) généralement obtiennent des meilleurs résultats dans ce tableau nous avons calculé la moyenne de score de chaque représentation pour savoir quelle sont les meilleures représentations vectorielle.

D'après les valeurs de moyenne nous avons conclu que meilleure représentation vectorielle c'est de combinaison entre (Mot, caractère et caractère N-gramme) avec (all≈76,83%)

III.3.2.4 sous tache _B

➤ Stage zéro

Nous rappelons que à sous tache précédent les classificateurs LR et LSVC qu'ils ont classé toujours en premier rang avec 76,8% à 76,8% , et donc nous avons remarqué dans cette sous tache que les deux classificateurs <LR et LSVC >ils ont gardé leurs places(≈Score) même si dans autre sous tache , par contre nous avons vu une recule des résultats pour les classificateurs BNB ,MNB,DT et pour les autres classificateurs ils ont toujours donnes presque le même score même

Mot		Caractère		Caractère N-gramme		Tous	
Modèle	Score	Modèle	Score	Modèle	Score	Modèle	Score
BNB	0.654	BNB	0.636	BNB	0.646	BNB	0.663
MNB	0.657	MNB	0.654	MNB	0.655	MNB	0.655
DT	0.627	DT	0.625	DT	0.635	DT	0.623
KNN	0.662	KNN	0.637	KNN	0.626	KNN	0.640
RF	0.685	RF	0.689	RF	0.690	RF	0.703
LR	0.688	LR	0.732	LR	0.729	LR	0.742
GB	0.712	GB	0.723	GB	0.727	GB	0.725
LSVC	0.726	LSVC	0.736	LSVC	0.746	LSVC	0.741

Table 19 : Résultats des classificateurs avant le prétraitement sous tache _B stage zéro

Chapitre III

➤ Stage 2

Dans ce section nous avons remarqué que les résultats qu'on a obtenu depuis la suppression des mots vides et la ponctuation presque les même par contre lors qu'on applique le pos Tag nous aurons des résultats un peu mieux par apport les autres technique du prétraitement.

Remove stopMot							
Mot		Caractère		Caractère N-gramme		Tous	
Modèle	Score	Modèle	Score	Modèle	Score	Modèle	Score
BNB	0.654	BNB	0.636	BNB	0.646	BNB	0.663
MNB	0.657	MNB	0.654	MNB	0.655	MNB	0.655
DT	0.623	DT	0.619	DT	0.647	DT	0.629
KNN	0.662	KNN	0.637	KNN	0.626	KNN	0.640
LR	0.686	LR	0.732	LR	0.729	LR	0.742
LSVC	0.726	LSVC	0.736	LSVC	0.746	LSVC	0.741

*Table 20 : Résultats des classificateurs après remove stopMot sous tache **_B** stage 1*

Remove ponctuation							
Mot		Caractère		Caractère N-gramme		Tous	
Modèle	Score	Modèle	Score	Modèle	Score	Modèle	Score
BNB	0.654	BNB	0.636	BNB	0.646	BNB	0.663
MNB	0.657	MNB	0.654	MNB	0.655	MNB	0.655
DT	0.636	DT	0.626	DT	0.645	DT	0.625
KNN	0.662	KNN	0.637	KNN	0.626	KNN	0.640
LR	0.688	LR	0.732	LR	0.729	LR	0.742
LSVC	0.726	LSVC	0.736	LSVC	0.746	LSVC	0.741

*Table 21 : Résultats des classificateurs après « remove ponctuation » sous tache **_B** stage 1*

Pos Tag							
Mot		Caractère		Caractère N-gramme		Tous	
Modèle	Score	Modèle	Score	Modèle	Score	Modèle	Score
BNB	0.653	BNB	0.660	BNB	0.674	BNB	0.671
MNB	0.662	MNB	0.658	MNB	0.661	MNB	0.660
DT	0.671	DT	0.674	DT	0.691	DT	0.657
KNN	0.678	KNN	0.679	KNN	0.647	KNN	0.659
LR	0.708	LR	0.756	LR	0.757	LR	0.768
LSVC	0.747	LSVC	0.768	LSVC	0.768	LSVC	0.765

Table 22 : Résultats obtenus après l'utilisation des Pos Tag sous tache _B stage 1

➤ Stage 4

Dans un ce stage on va présenter deux classificateur <LSVC et LR> avec la suppression des mots vide et les punctuations, et nous avons appliqué aussi <stem, pos Tag > Avec toutes les phases <Mot, caractère, caractère N-gramme, all>

Remove (ponctuation, stop Mot) et appliquer (stem, pos tag)							
Mot		Caractère		Caractère N-gramme		Tous	
Modèle	Score	Modèle	Score	Modèle	Score	Modèle	Score
LR	0.693731	LR	0.726964	LR	0.730363	LR	0.740181
LSVC	0.733006	LSVC	0.740937	LSVC	0.741314	LSVC	0.748112

Table 23 : Résultats obtenus pour sous tache _B stage 4

D'après tous ce qu'on a vu (les tableaux précédant) dans sous tache _B nous avons remarqué que les deux classificateurs (LR, LSVC) généralement obtiennent des meilleurs résultats dans ce tableau nous avons calculé la moyenne des scores de chaque représentation pour savoir quelle sont les meilleures représentations vectorielle.

D'après les valeurs de moyenne nous avons conclu que meilleure représentation vectorielle c'est de combinaison entre (Mot, caractère et caractère N-gramme) avec (all≈74,8%)

Chapitre III

III.3.2.5 Sous tache _C

➤ Stage zéro

Voici les résultats des classificateur dans sous tache C pour <Mot, caractère, caractère N-gramme, all>avant d'appliquer les technique prétraitement <PosTag ,stopMot ,tem,Lem>.

Mot		Caractère		Caractère N-gramme		Tous	
Modèle	Score	Modèle	Score	Modèle	Score	Modèle	Score
BNB	0.700	BNB	0.678	BNB	0.664	BNB	0.701
MNB	0.700	MNB	0.700	MNB	0.700	MNB	0.700
DT	0.634	DT	0.617	DT	0.616	DT	0.621
KNN	0.694	KNN	0.664	KNN	0.636	KNN	0.655
RF	0.711	RF	0.711	RF	0.713	RF	0.718
LR	0.708	LR	0.720	LR	0.723	LR	0.737
GB	0.724	GB	0.725	GB	0.721	GB	0.725
LSVC	0.731	LSVC	0.738	LSVC	0.742	LSVC	0.740

Table 24 : résultats des classificateurs avant le prétraitement sous tache _B stage zéro

III.3.2.6 Sous tache _Arabic

➤ Stage zéro

Dans ce section nous avons prend la comparaison entre les deux langues<arabe, anglaise >avant d'appliquer les techniques du prétraitement

Nous avons remarqué que les scores des classificateurs dans cette tache (arabic) plus élevés (ils se sont améliorés) que les taches anglaises et le classificateur GB prendre la place du LR avec une note $\approx 86.6\%$ par contre le classificateur LSVC il a gardé sa place toujours en premier avec une note 87%

Mot		Caractère		Caractère N-gramme		Tous	
Modèle	Score	Modèle	Score	Modèle	Score	Modèle	Score
BNB	0.821	BNB	0.816	BNB	0.822	BNB	0.816
MNB	0.821	MNB	0.821	MNB	0.821	MNB	0.821
DT	0.781	DT	0.791	DT	0.786	DT	0.776
KNN	0.832	KNN	0.821	KNN	0.830	KNN	0.832
RF	0.827	RF	0.836	RF	0.835	RF	0.845
LR	0.817	LR	0.828	LR	0.828	LR	0.845
GB	0.837	GB	0.858	GB	0.858	GB	0.866
LSVC	0.825	LSVC	0.867	LSVC	0.870	LSVC	0.863

Table 25 : Résultats des classificateurs avant le prétraitement pour sous tache *_Arabic* stage zéro

➤ Stage 4

Dans cette section nous avons appliqué quelque technique du prétraitement en arabe textuelle <la suppression des mots vide, les ponctuations, Pos Tag et lemmatisation>

Nous avons remarqué dans ce tableau en dessous que les classificateurs GB et LSVC ils partagent les première place pour la phase du mot le meilleur c'est le GB avec une note $\approx 83.5\%$ et pour les autres phases c'est le LSVC avec une note 87%

à la fin nous concluons que le meilleur représentation vectorielle c'est de prendre toute les phases <all> avec LSVC classificateur.

Mot		Caractère		Caractère N-gramme		Tous	
Score	Score	Modèle	Score	Modèle	Score	Modèle	Score
BNB	0.821	BNB	0.816	BNB	0.821	BNB	0.816
MNB	0.821	MNB	0.821	MNB	0.821	MNB	0.821
DT	0.802	DT	0.796	DT	0.830	DT	0.802
KNN	0.823	KNN	0.822	KNN	0.826	KNN	0.830
RF	0.827	RF	0.835	RF	0.836	RF	0.835
LR	0.820	LR	0.833	LR	0.835	LR	0.848
GB	0.835	GB	0.857	GB	0.861	GB	0.855
LSVC	0.826	LSVC	0.866	LSVC	0.872	LSVC	0.866

Table 26 : Résultats des classificateurs pour sous tache *_Arabic* stage 04

En résumé

D'après tous ce que nous avons vu dans les parties précédentes (utilisation de toutes les techniques et les fonctions de prétraitement parmi tous les classificateur) :

Il Ya LSVC et LR qui donnent des meilleur scores pour les textes anglais et GB, LSVC pour les textes arabes.

III.3.3. Les résultats obtenus

III.3.3.1. Corpus d'anglais

III.3.3.1.1. L'exécution de sous tache _A :

Après l'exécution de sous taches _A nous aurons 6 stages on va les citer comme suivante: et d'après l'utilisation des commandes d'exécution nous avons obtenu les résultats suivante:

- pour le stage 0 → 4 commandes
- pour le stage 1 → 20 commandes
- pour le stage 2 →40 commandes
- pour le stage 3→40 commandes
- pour le stage 4→20 commandes
- pour le stage 5→4 commandes

En totale nous aurons 128 commandes pour sous tache _A parmi les commandes qui nous avons obtenu

Exemple de commandes:

```
<python textClassification.py -tr trainSubTask_a.xlsx -trclm "tweet, subtask_a" -tst testSubTask_a.xlsx -tstclm "tweet, subtask_a" -vec all -s "english" -p True -st True -pos True -lem True > outExp125.log>
```

Et après l'exécution de 128 commandes on peut y avoir 128 expériences.

Chapitre III

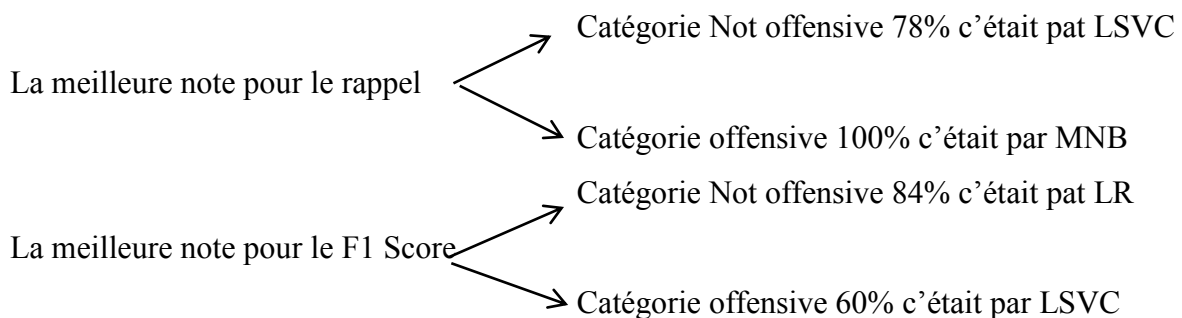
➤ Stage zéro

Le tableau en dessous montre les résultats obtenus avec les huit classificateurs d'apprentissage automatique (sous tâche _A →stage 0 →expérience 03).

Modèle	Tweet	Précision	Rappel	F1-score	Support	Score
KNN	Not	0.83	0.70	0.76	2060	0.655211
	Off	0.32	0.50	0.39	588	
LSVC	Not	0.88	0.78	0.83	1967	0.758308
	Off	0.52	0.70	0.60	681	
LR	Not	0.95	0.74	0.84	2214	0.754154
	Off	0.38	0.80	0.52	434	
DT	Not	0.76	0.74	0.75	1764	0.668051
	Off	0.50	0.52	0.51	884	
MNB	Not	1.00	0.66	0.79	2641	0.657100
	Off	0.01	1.00	0.02	7	
BNB	Not	0.80	0.71	0.76	1940	0.660498
	Off	0.40	0.51	0.45	708	
RF	Not	0.95	0.72	0.82	2281	0.723565
	Off	0.30	0.75	0.43	367	
GB	Not	0.95	0.73	0.83	2254	0.742825
	Off	0.34	0.80	0.48	394	

Table 27 : Les résultats pour sous tâche _A stage 0 expérience 3

Le tableau précédant (au-dessus) montre la valeur de <rappel, précision et F1 Score> pour chaque prédiction nous avons remarqué que la meilleure précision parmi tous les classificateurs c'était part MNB avec une note 100% pour la catégorie Not offensive et pour la catégorie offensive c'était par LSVC avec une note 52%.



Chapitre III

Pour le passage d'un stage un à un autre nous avons éliminé les deux classificateurs qui ont des Scores bas <KNN→0.655211 et MNB→0.657100>.

Et nous avons répété le même travail jusqu'à y arrive a dernier stage avec un meilleur classificateur qui a un bon score.

➤ Stage 5

Le tableau en dessous montre le résultat final avec le meilleur classificateur, nous avons éliminé les autres classificateurs dans une étape précédente.

Nous avons remarqué que le rappel l'était 78% pour les mots Not offensive et 73% pour les mots offensives, et concernant f1 Score 84% Not offensive et 61% pour les mots offensive.

Modèle	Tweet	Précision	Rappel	F1-score	Support	Score
LSVC	Not	0.90	0.78	0.84	1986	0.768505
	Off	0.53	0.73	0.61	662	

Table 28 : Les résultats de sous tache_A du stage 5 expérience 3

D'après les résultats précédents nous avons conclu que le meilleur classificateur dans cette tache (sous tache _A), c'était part LSVC (linear support vector machine).

III.3.3.1.2. L'exécution de sous tache _B :

Après l'exécution de sous taches _B nous aurons 6 stages on va les citer comme suivante, et d'après l'utilisation des commandes d'exécution nous avons obtenu les résultats suivantes:

- pour le stage 0 → 4 commandes
- pour le stage 1 → 20 commandes
- pour le stage 2 →40 commandes
- pour le stage 3→40 commandes
- pour le stage 4→20 commandes
- pour le stage5→4 commandes

En totale nous aurons 128 commandes pour la sous tache _B parmi les commandes que nous avons obtenu

```
<python textClassification.py -tr trainSubTask_b.xlsx -trclm "tweet, subtask_b" -tst
testSubTask_b.xlsx -tstclm "tweet, subtask_b" -vec all > outExp115.log>
```

➤ Stage zéro

Le tableau au-dessous montre les résultats obtenus avec le huit classificateur d'apprentissage automatique (sous tache _B →stage 0 →expérience 03).

Modèle	Tweet	Précision	Rappel	F1-score	Support	Score
KNN	TIN	0.31	0.44	0.36	563	0.637840
	UNT	0.03	0.16	0.05	25	
	Nan	0.83	0.70	0.76	2060	
LSVC	TIN	0.50	0.63	0.56	631	0.736027
	UNT	0.03	0.40	0.06	10	
	Nan	0.89	0.77	0.83	2007	
LR	TIN	0.34	0.74	0.46	363	0.732251
	UNT	0.01	0.50	0.02	2	
	Nan	0.96	0.73	0.83	2283	
DT	TIN	0.42	0.43	0.43	786	0.625378
	UNT	0.09	0.14	0.11	79	
	Nan	0.76	0.73	0.74	1783	
MNB	TIN	0.00	0.00	0.00	1	0.654456
	UNT	0.00	0.00	0.00	0	
	Nan	1.00	0.65	0.79	2647	
BNB	TIN	0.38	0.44	0.41	687	0.636329
	UNT	0.00	0.00	0.00	0	
	Nan	0.80	0.71	0.75	1961	
RF	TIN	0.36	0.55	0.44	521	0.689577
	UNT	0.02	0.40	0.03	5	
	Nan	0.89	0.72	0.80	2122	
GB	TIN	0.32	0.68	0.44	371	0.723187
	UNT	0.02	0.27	0.04	11	
	Nan	0.96	0.73	0.83	2266	

Table 29 : Les résultats de sous tache _B du stage 0 expérience 03

Chapitre III

Dans ce tableau précédant nous avons remarqué que la meilleur précision (50%) et F1 score (56%) pour la classe target (TIN) c'était part LSVC et pour le rappel (74% classe target <TIN> et 50%classe untarget <UNT>) c'était part LR.

Dans le passage d'un stage a un autre nous avons éliminé les deux classificateur qui ont des Scores bas dans cette sous tache nous avons éliminé DT →0.625378 et BNB →0.636329.

Pour l'étape suivante on va afficher les résultats concernant le stage 3 sous tache _B.

➤ Stage 3

Dans ce table en dessous nous avons affiché des résultats avec 4 classificateur après l'élimination les classificateur KNN, MNB, DT et BNB.

Modèle	Tweet	Précision	Rappel	F1-score	Support	Score
LSVC	TIN	0.41	0.64	0.50	506	0.726964
	UNT	0.03	0.40	0.06	10	
	Nan	0.92	0.75	0.83	2132	
LR	TIN	0.15	0.68	0.25	176	0.688444
	UNT	0.00	0.00	0.00	3	
	Nan	0.98	0.69	0.81	2469	
RF	TIN	0.25	0.57	0.35	347	0.683535
	UNT	0.03	0.15	0.05	26	
	Nan	0.93	0.71	0.80	2275	
GB	TIN	0.24	0.71	0.36	272	0.711480
	UNT	0.14	0.39	0.20	44	
	Nan	0.97	0.72	0.82	2332	

Table 30 : Les résultats de sous tache _B du stage 3 expériences 03

➤ Stage 5

Le tableau au-dessous montre le résultat final avec le meilleur classificateur, nous avons éliminé les trois classificateurs dans une étape précédente.

Model	Tweet	Précision	Rappel	F1-score	Support	Score
LSVC	TIN	0.50	0.65	0.56	612	0.741314
	UNT	0.03	0.50	0.06	8	
	Nan	0.90	0.77	0.83	2028	

Table 31 : les résultats de sous tache _B du stage 5 expériences 04

D'après les résultats précédant nous avons conclu que le meilleur classificateur dans cette tache (sous tache _B), c'était part LSVC (linear support vector machine).

III.3.3.1.3. L'exécution de sous tache _C :

Dans cette sous tache nous avons exécuté la sous taches _C et nous aurons 6 stages on va les citer comme suivante: et d'après l'utilisation des commandes d'exécution nous avons obtenu les résultats suivantes:

- pour le stage 0 → 4 commandes
- pour le stage 1 → 20 commandes
- pour le stage 2 →40 commandes
- pour le stage 3→40 commandes
- pour le stage 4→20 commandes
- pour le satage5→4 commandes

En totale nous aurons 128 commandes pour sous tache _C parmi les commandes que nous avons obtenu.

```
<python textClassification.py -tr trainSubTask_c.xlsx -trclm "tweet, subtask_c" -tst testSubTask_c.xlsx -tstclm "tweet, subtask_c" -vec all -p True > outExp117.log>
```

le tableau au-dessous montre les résultats obtenus avec l'huis classificateurs d'apprentissage automatique (sous tache _C->stage 1 ->expérience 03

Chapitre III

Modèle	Tweet	Précision	Rappel	F1-score	Support	Score
KNN	GRP	0.11	0.15	0.12	151	0.664275
	IND	0.21	0.40	0.27	262	
	OTH	0.01	0.07	0.02	15	
	Nan	0.88	0.73	0.80	2220	
LSVC	GRP	0.13	0.41	0.20	66	0.738671
	IND	0.34	0.58	0.43	290	
	OTH	0.01	0.50	0.02	2	
	Nan	0.95	0.77	0.85	2290	
LR	GRP	0.05	0.59	0.09	17	0.720166
	IND	0.13	0.69	0.22	97	
	OTH	0.00	0.00	0.00	0	
	Nan	0.99	0.72	0.83	2534	
DT	GRP	0.17	0.17	0.17	207	0.620091
	IND	0.32	0.34	0.33	476	
	OTH	0.04	0.05	0.04	65	
	Nan	0.78	0.76	0.77	1900	
MNB	GRP	0.00	0.00	0.00	0	0.700906
	IND	0.00	0.00	0.00	0	
	OTH	0.00	0.00	0.00	0	
	Nan	1.00	0.70	0.82	2648	
BNB	GRP	0.18	0.20	0.19	178	0.678248
	IND	0.12	0.47	0.19	131	
	OTH	0.00	0.00	0.00	0	
	Nan	0.92	0.73	0.81	2339	
RF	GRP	0.07	0.45	0.13	33	0.720921
	IND	0.22	0.56	0.32	202	
	OTH	0.00	0.00	0.00	2	
	Nan	0.96	0.74	0.83	2411	
GB	GRP	0.07	0.52	0.12	27	0.725076
	IND	0.18	0.62	0.28	151	
	OTH	0.01	0.20	0.02	5	

	Nan	0.98	0.74	0.84	2465	
--	-----	------	------	------	------	--

Table 32 : Les résultats de sous tache _C du stage 1 expérience 03

D'après le tableau au-dessus nous avons remarqué que le meilleur F1 Score 20% pour la classe groupe (GRP) et 43% pour la classe individuelle (IND) c'était part LSVC, concernant le rappel 59% pour la classe Groupe et 69% pour la classe individuelle c'était part LR et 50% pour la classe other (OTH).

Pour la précision il y a un écart entre les classificateurs 34% pour la classe individuelle (IND) C'était part LSVC et 18% → groupe (GRP) c'était par BNB.

Pour le passage d'un stage à un autre nous avons éliminé les deux classificateurs qui ont des scores bas dans cette sous tache nous avons éliminé DT → 0.620091 et KNN → 0.664275.

Pour l'étape suivante on va afficher les résultats concernant le stage 5.

➤ Stage 5

Le tableau en dessous montre le résultat final avec le meilleur classificateur

Model	Tweet	Précision	Rappel	F1-score	Support	Score
LSVC	GRP	0.13	0.39	0.19	66	0.737915
	IND	0.35	0.59	0.44	297	
	OTH	0.01	0.50	0.02	2	
	Nan	0.94	0.77	0.85	2283	

Table 33 : Les résultats de sous tache _C du stage 5 expériences 03

D'après les résultats précédant nous avons conclu que le meilleur classificateur dans cette tache (sous tache _C), c'était part LSVC (linear support vector machine).

III.3.3.2. Corpus arabe

Dans cette tache nous avons fait le même travail précédant comme nous avons vu en tache anglais après l'exécution de cette tache nous aurons six stage et après l'exécution des commandes il va nous rendre 128 expériences.

Parmi tous les résultats que nous avons vu nous allons afficher l'expérience 11 du stage 4

Modèle	Tweet	Précision	Rappel	F1-score	Support	Score
KNN	Non-Offensive	0.99	0.84	0.91	774	0.83125
	Offensive	0.12	0.65	0.20	26	
LSVC	Non-Offensive	0.98	0.87	0.92	743	0.86500
	Offensive	0.32	0.81	0.46	57	
LR	Non-Offensive	1.00	0.84	0.91	775	0.84500
	Offensive	0.15	0.88	0.26	25	
DT	Non-Offensive	0.87	0.86	0.87	667	0.77750
	Offensive	0.34	0.37	0.36	133	
MNB	Non-Offensive	1.00	0.82	0.90	800	0.82125
	Offensive	0.00	0.00	0.00	0	
BNB	Non-Offensive	0.99	0.82	0.90	796	0.81625
	Offensive	0.00	0.00	0.00	4	
RF	Non-Offensive	0.99	0.85	0.91	768	0.84125
	Offensive	0.17	0.75	0.27	32	
GB	Non-Offensive	0.99	0.86	0.92	754	0.85625
	Offensive	0.26	0.80	0.39	46	

Table 34 : Les résultats de tâche arabe stage 4 expérience 11

Le meilleur F1 Score (F mesure) pour la catégorie « Not offensive » c'était par LSVC et GB (92%)

Concernant le rappel :

La classe Not offensive 87%→LSVC

La classe offensive 88%→LR.

Pour la précision le LSVC et MNB partagent la même valeur 100%(Not offensive)

Et 34% pour la catégorie offensive c'était part DT.

Le tableau en dessous montre les résultats du dernier stage (5)

Modèle	Tweet	Précision	Rappel	F1-score	Support	Score
LSVC	Non-Offensive	0.98	0.87	0.93	771	0.87
	Offensive	0.34	0.83	0.49	59	

Table 35 : les résultats de tache arabe stage 5 expérience 4

D'après les résultats précédant nous avons conclu que le meilleur classificateur pour cette tache (sous tache _C), est LSVC aussi (linear support vector machine).

Une petite conclusion :

- ✓ dans ce chapitre nous avons découvert que la meilleure représentation vectorielles c'est la combinaison entre tous les cas possible <tout>.
- ✓ le modèle le plus performant de notre data est le modèle LSVC.

III.3.4. L'apprentissage profond

Dans cette partie nous avons abordé deux algorithmes d'apprentissage automatique profond parmi tout ce que nous savons (nous avons traité notre data <seulement les sous taches de la langue anglaises>avec simple RNN et LSTM).

III.3.4.1. Modèle LSTM

III.3.4.1.1. Sous tache_A (binary classification)

Dans cette partie nous présentons au Tableau suivant qui est composé d'une couche d'entrée (embedding layer <150>) et de couche LSTM (64 unités) et nous avons utilisé deux couches en plus (l'une contient 256 unités <FC 1>et l'autre une seule unité <pour out put> avec deux fonctions d'activations (tanh, sigmoïde).

Layer (type)	Output Shape	Param #
inputs (InputLayer)	[(None, 150)]	0
embedding_1 (Embedding)	(None, 150, 50)	50000
simple_rnn_1 (SimpleRNN)	(None, 64)	7360
FC1 (Dense)	(None, 256)	16640
activation_2 (Activation)	(None, 256)	0
dropout_1 (Dropout)	(None, 256)	0
out_layer (Dense)	(None, 1)	257
activation_3 (Activation)	(None, 1)	0
Total params: 74,257		
Trainable params: 74,257		
Non-trainable params: 0		

Figure 47 : Les paramètres de configuration LSTM

Les Tweets en entrée est de nombre maximum des mots 50, et la dimension de la représentation vectoriel 150

Param = vocabulaire* dimension+ biais.

Totale Param (embedding)= 50*1000+0=50000

Trainable parameter =96,337.

Non_Trainable paramètre = 96,337

Non_Trainable =0

Input_length = 50 : la taille maximale de notre tweets et de 150 mots.

Trainable = False : ne pas mettre à jour les poids, et le contraire pour (True)

Dropout : C'est une technique de régularisation (pour combattre l'overfitting), [nous avons travaillé avec optimiser <Adam> avec une valeur 50%]

Recurrent_dropout : C'est une technique de régularisation pour combattre l'overfitting dans l'état récurrent.

W_regularizer : permettent d'appliquer des pénalités sur les paramètres de couche ou l'activité de couche pendant l'optimisation. Ces pénalités sont intégrées dans la fonction de perte optimisée par le réseau.

Batch_normalization : Cette technique peut améliorer fortement la convergence lors de l'entraînement. Elle consiste à normaliser en moyenne et en variance les sorties des couches du réseau.

Epoche : c'est le nombre maximum d'itérations d'entraînement (pour notre cas nous l'avons effectué une valeur 100).

Batch_Size : définit le nombre d'échantillons qui vont être propagés à travers le réseau. Nous l'avons effectué une valeur égale 1.

Embedding layer : la première couche de notre modèle, avec une taille de 150 qui représente chaque mots.

Nous avons effectué 15 % pour le test et 85 % pour le train et 30 % pour la validation et nous avons donné 64 unités pour LSTM.

III.3.4.1.2. Sous tache_B (multi-classes) :

En sous tache A et B Pour les paramètres du LSTM et nous l'avons changé complètement

Tableau suivant qui est composé d'une couche d'entrée (embedding layer <32>) et de couche LSTM (100 unités) et nous avons utilisé une couche de sorti <softmax>qui contient (3 unités).

➤ Model: "functional_11"

Layer (type)	Output Shape	Param #
inputs (InputLayer)	[(None, 150)]	0
embedding_6 (Embedding)	(None, 150, 32)	32000
dropout_6 (Dropout)	(None, 150, 32)	0
lstm_6 (LSTM)	(None, 100)	53200
dense_5 (Dense)	(None, 3)	303

Total params: 85,503
Trainable params: 85,503
Non-trainable params: 0

Figure 48: Montre les paramètres du lstm pour multi-classes

➤ Evaluations

Le tableau en dessous représente les résultats d'évaluation du modèle LSTM pour les trois sous taches (A, B et C).

Chapitre III

Accuracy: $(tp + tn) / (p + n)$

```
accuracy = accuracy_score(testy, classes)
```

```
print('Accuracy: %f % accuracy)
```

precision $tp / (tp + fp)$

```
precision = precision_score(testy, classes)
```

```
print('Precision: %f % precision)
```

recall: $tp / (tp + fn)$

```
recall = recall_score(testy, classes)
```

```
print('Recall: %f % recall)
```

f1: $2 tp / (2 tp + fp + fn)$

```
f1 = f1_score (testy, classes)
```

```
print('F1 score: %f % f1)
```

Tache	Précision	Rappel	F1 Score	Accuracy	Fonction loss
Sous tache _A	0.6878	0.4435	0.5199	0.7437	0.5703
Sous tache _B	0.8905	1.002	0.98	0.7150	0.6799
Sous tache _C	0,7680	1,0001	0,878	0,6982	0,5670

Table 36: les résultats d'évaluation du modèle LSTM.

➤ Les graphes

Tache	accuracy	Fonction loss
Sous tache _A		

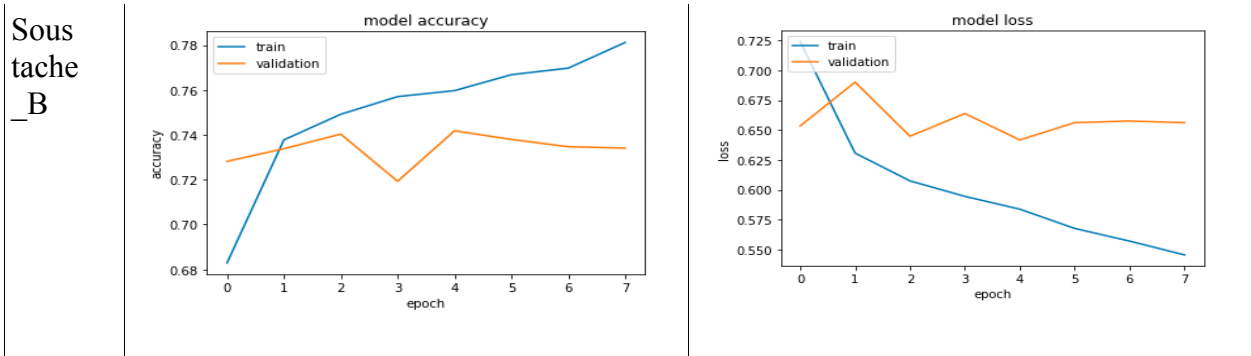


Table 37 : Montre les graphes du LSTM en différents taches

Nous avons remarqué pour le graphe précédent (ce que lié au accuracy) que les valeurs de train (en Bleu) et de validation (en Rouge) dans une augmentation continue tout à fait normale selon le nombre des epochs.

Et pour la fonction de perte les valeurs de train et de validation Dans une nette diminution selon le nombre des epochs.

III.3.3.2. Modèle Simple RNN

Dans cette partie nous avons travaillé avec le modèle simpleRNN en utilisant les même paramètres du modèle LSTM pour chaque sous tache et nous avons obtenus les résultats comme suivant :

➤ Évaluation

Tache	Précision	Rappel	F1 Score	Accuracy	Fonction loss
Sous tache _A	0.7500	0.3109	0.4259	0.7427	0.6116
Sous tache _B	0.7982	1.0001	0.9216	0.6989	0.7490
Sous tache _C	0,701	0,665	0,876	0,6848	0,5650

Table 38 : Les résultats d'évaluation du modèle Simple RNN

➤ les graphes

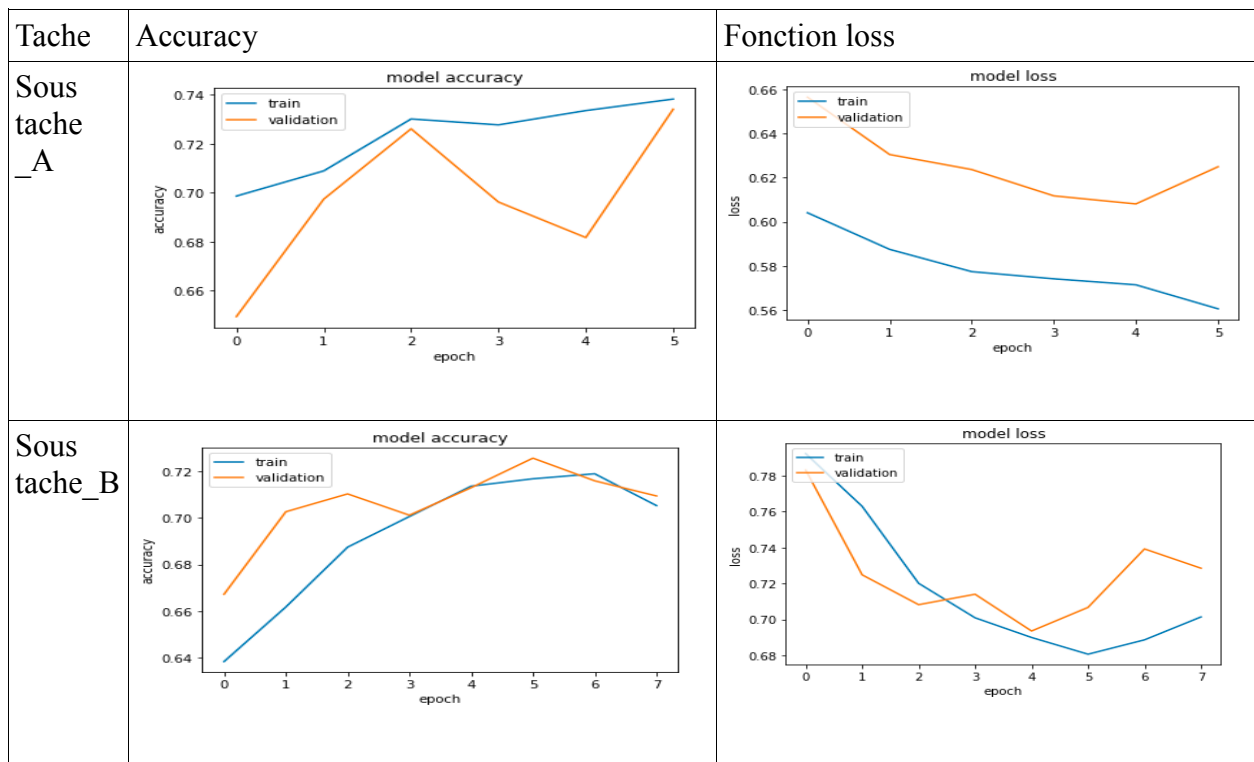


Table 39 : Montre les graphes du SimpleRNN en différents taches

III.3.3.3.Comparision avec les travaux

Dans cette partie nous avons fait une petite comparaison entre les travaux précédents et notre travail concernant le résultat du LSTM.

Algorithme	Auteur	Corpus	La langue	Les taches	F1 Score
LSTM	Marcos Zampieri 1, Preslav Nakov 2, Sara Rosenthal 3, Pepa Atanasova 4, Georgi Karadzhov...	9093037 tweets	Anglaise	Sous tache_A	0.681
				Sous tache_B	0.657
				Sous tache_C	0.585
	Houari Benchabekh	39720 tweets	Anglaise	Sous tache_A	0.5199
				Sous tache_B	0.98
				Sous tache_C	0,878

Table 40 : Comparaison avec les travaux précédant

III.4. Application :

Concernant l'application nous avons travaillé avec « framework » <flask> et pour le modèle nous avons testé nos tweets avec le modèle LSVC la figure(51) suivante montre la première fenêtre ou nous pouvons tester les tweet en langue arabes ou bien les tweet en langue anglaise.



Figure 49 : Interface d'accueil

Nous allons commencer tout d'abord par traiter les tweets anglais, on va prendre ce tweet par exemple < **you have a black face** > donc notre système va détecter est ce qu'il est du type offensant ou pas. la figure suivantes monte le résultat de détection avec quelques exemples de sous tache _A.

TEXT OFF OR Not

Tweet

Predict
Next
Home

This Tweet is Offensive

Label	Tweet
Not	@USER @USER Why is John Kerry running his mouth again as if we cared about what he has to say? I can't think of a single damn thing Kerry has accomplished besides run his mouth. Trump's accomplishments exceed anything Kerry could ever dream of. Even before becoming president!
Off	@USER @USER @USER How do you come up with all these lies. You have not done a thing as far as gun control. Open your eyes Obama your hero had 8 years in office. All your doing is sucking around for the last minute vote.
Not	@USER What manga you reading?
Off	@USER Yea Jeff should go. Antifa is a terrorist organizationL

Figure 50 : Détection du tweet dans les textes anglais(offensive)

S'il est de type « not offensive » (non offensant) on s'arrête ici sinon (offensive) on va passer à l'étape suivante pour la catégorisation et spécification de quel genre de l'attaque. Pour la détection du type not offensive nous allons prendre un autre exemple .

Pour l'exemple précédant nous allons le catégoriser soit de type target (ciblé) ou untarget (non ciblé) la figure suivante montre la catégorie du tweet avec quelques exemples de catégorisation.

TARGET OR UNTARGET Classifier

Tweet

Next
previous

Tweet Offensive and Untarget

Label	Tweet
TIN	Go home you're drunk!!! @USER #MAGA #Trump2020 🇺🇸🇺🇸 URL
UNT	@USER @USER #Westminster @USER #Tories @USER @USER @USER @USER Absolutely pathetic #appeasement 🤢🤢🤢
TIN	Thanks for your subscription to Ringtone UK your mobile will be charged \$5/month Please confirm by replying YES or NO. If you reply NO you will not be charged
UNT	ANTI-ANTIFA IS BALLS

Figure 51 : Catégorisation des tweet (sous tache_B)

S'il appartient à la catégorie (untarget) on s'arrête ici sinon (target) on va passer à l'étape suivante pour spécification du genre de l'attaque.

Pour la dernière étape définir quel genre d'attaque il peut être de type individuel groupe, autre, donc la figure suivante montre le genre de ce tweet <**How do you come up with all these lies**>

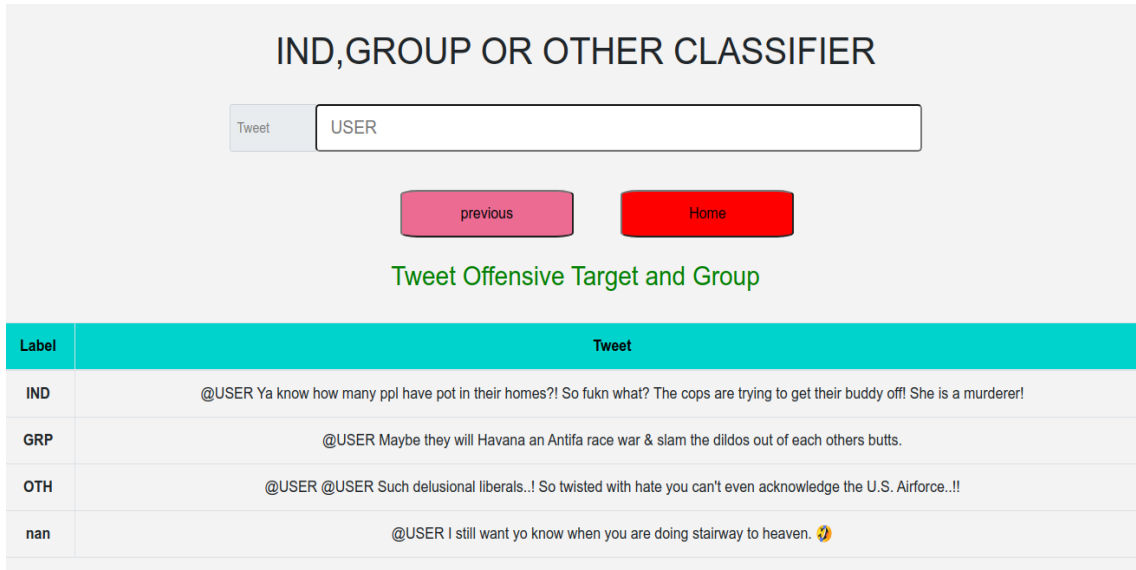


Figure 53 : Montre le genre du tweet (sous tache _C)



Figure 52 : Détection de type du tweet arabe (offensive)

<انت وسخ وقناتك وسخة> l'exemple dans la figure au dessus

Exemple 2: pour le tweet <الرئيس عبد المجيد تيون مريض با الكرونا>

Offensive Or Not Arabic Classifier

Tweet: الرئيس عبد المجيد تيون مريض با الكرونا

Predict

Home

هذه التغريدة ليست مسيئة

Label	Tweet
Offensive	@User.IDX أه يا فتاة وسخة ما فيهاش رجل 🤔🤔🤔🤔
Non-offensive	يعني بالله وين الموضوع المهم اللي اتفكرك دول الخليج علاقتها جيدة مع اسرائيل ولها مكتب في قطر يعني بنضحكك على وعلى مشكلة ايران ولا دولة خليجية او حتى عربية اعلنت الحرب عليها كله قطعنا علاقات وماعرف ايش وقاعدة العبيدي في قطر تحمى دول الخليج زيها زي كوريا الجنوبية واليابان من ايران وضحكوا عليكم الحكومات بكم صفقة شراء صاروخ وماعرف ايش . (اصحوا يا عرب اصحوا)
Offensive	هاذي لبنان اقسى من فلوبيهم ما فيش بلد عديمة الإنسانية جيرانكم الزفت ومعاملة عنصرية لي هالدرجة سبحان الله بكرة لما يجيكم الدور من اسرائيل مش حلقو جار اكرم من سوريا
Non-offensive	الحلقة دي ممكن تغير منظورنا في حاجات كتير حولينا

Figure 54 : Détection de type du tweet arabe (not offensive)

III.5.conclusion

Dans ce chapitre nous avons découvrons l'environnement sur lequel nous avons travaillé avec ces outils et les bibliothèques que nous avons utilisées, en suite les résultats des représentations vectorielles pour les deux parties arabe et anglaise et nous avons eu comme conclusion que la représentation optimale c'était par (tous) la combinaison entre les trois. Après nous avons abordé à la comparaison entre les huit classificateurs en utilisant la Métrique d'évaluations (F mesure), et nous avons constaté que le meilleur classificateur pour Notre data est le LSVC pour les deux partie de notre corpus (anglaise, arabe).d' un autre coté nous avons touché quelque algorithmes d'apprentissage automatique profond et nous avons représenté les hyper paramètres que nous avons utilisés, et les résultats que nous avons obtenu.

En fin, nous avons représenté notre système par une application performante et Efficace

Conclusion Générale

Conclusion générale et perspectives

Nous avons essayé par notre projet de fin d'étude d'accompagner la lutte contre la propagation du langage offensant sur les réseaux sociaux. En outre, notre système (une application Web) permet de classificateur selon trois niveaux et critères différents (le type, la cible, le type de la cible) des textes extraient des commentaires en langue arabe et en langue Anglaise en utilisant notre corpus collecté depuis Tweeter.

Dans ce travaille, nous avons rencontré beaucoup de problèmes et de difficultés concernant les langues et leur caractéristiques. Notamment, la langue arabe et sa complexité particulière durant la phase de la classification.

Nous avons aussi exploité grâce à notre approche les différents algorithmes d'apprentissage automatique et profond, entraînés sur les données de notre corpus. Les algorithmes ont été testés et évalués tout en établissant une comparaison entre chacun de ces algorithmes, cependant, nous avons obtenu des bons résultats dans l'apprentissage automatique avec l'algorithme (LSVC) et d'excellents résultats dans l'apprentissage profond avec l'algorithme LSTM, concernant les représentations vectorielles nous avons conclu (après test) que la meilleure parmi les quatre représentations c'est de prendre la combinaison entre eux (le tout), Les différentes expériences vécu lors de ce projet nous amènes à envisager beaucoup de perspectives, parmi ces perspectives :

- enrichir le corpus pour avoir plus de Data.
- étudier d'autres approches et algorithmes.
- étudier et travailler avec d'autres langues
- étudier et travailler avec les dialectes notamment ceux de notre pays l'Algérie.
- pouvoir détecter celui qui utilise le langage offensant en enregistrant tous ces informations personnelles, afin d'aider la police électronique et les poursuivre judiciaires.
- traiter d'autres type à par le texte comme les images, les vidéos et les sons.
- Rendre notre application plus efficace et plus performante .

Références

Références

[1]. **Shashank H. Yadav & Pratik M. Manwatkar An approach for offensive text detection and prevention in Social Networks**

[2]. Jocelyn Ziegler et Ibrahim Shalabi, Elèves-Avocats 3 décembre 2019 LES DÉLITS D'OPINION SUR LES RÉSEAUX SOCIAUX À L'ÉPREUVE DE LA LIBERTÉ D'EXPRESSION ?

[3]. David Barrière Publié le 4 novembre 2014 Où s'arrête la liberté d'expression sur les médias sociaux ?

[5]. T. Johnson, R. Shapiro et R. Tourangeau, «Enquête nationale sur les attitudes des Américains à l'égard de la toxicomanie XVI: les adolescents et les parents.», Dans le National Center on Addiction and Substance Abuse.vol.2011, 2011

[6]. Segun Taofeek Aroyehun & Alexander Gelbukh.2018. Détection d'agression dans les médias sociaux: utilisation de réseaux de neurones profonds, augmentation des données et pseudo-étiquetage.Dans les actes de TRAC-2018

[7]. Hao-Ren Yao,Eugene Yang,Katina Russell,Nazli Goharian,Ophir Frieder
Published: August 20, 2019

[8]. Aytuğ Onan &Serdar Korukoğlu & Hasan Bulut
Expert Systems with Applications: An International Journal September 2016

[9]. (Mubarak et al., 2020) WOLI at SemEval-2020 Task 12: Arabic Offensive Language Identification on Different Twitter Datasets

[10]. (Pitenis et al., 2020) SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020)

[11]. Marcos Zampieri 1, Preslav Nakov 2 , Sara Rosenthal 3 , Pepa Atanasova 4 , Georgi Karadzhov 5 Hamdy Mubarak 2 , Leon Derczynski 6 , Zeses Pitenis 7 , C, agrı C, ~ oltekin ..
8 SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020)

[12].T.DERDRA Amel, F.BENSFIA, « La Représentation Conceptuelle pour la Catégorisation des Textes Multilingue », Mémoire de Master, Université Abou Bakr Belkaid–Tlemcen, 2011-2012.

[14]. John Morkes et Jakob Nielsen Les contraintes du nouveau media

[15]. S.ABDELOUAHAB, «Processus de classification supervisée de textes arabes par la méthode K PPV Application aux articles de presse», Mémoire de Master, Université de M'sila, 2011-2012.

Références

- [16]. M. F. Porter, «An algorithm for suffix stripping », Program, pp 130–137, Morgan Kaufmann Publishers Inc, 1980.
- [17]. J.Clech, D.A.Zighed. « Une technique de réétiquetage dans un contexte de catégorisation de textes » 2004.
- [18]. Leila Baccour publié en 2005 un Système de Segmentation de Textes Arabes basé sur l'analyse contextuelle des signes de ponctuations et de certaines particules, Tunisie : Laboratoire LARIS-Faculté des Sciences Economiques et de Gestion BP 1088, 3018-Sfax–, 2005.
- [19]. arabes Lamia Hadrich Belguith, Chafik Aloulou & Abdelmajid Ben Hamadou MASPAR : De la segmentation à l'analyse syntaxique de textes
- [20]. M. CHERAGUI, une Analyse Morphologique de la langue arabe basée sur l'Aide Muticritere à la Décision, Algérie: Université d'Adrar, 2012.
- [21]. Sara Rosenthal¹, Pepa Atanasova², Georgi Karadzhov³, Marcos Zampieri⁴, Preslav Nakov A Large-Scale Semi-Supervised Dataset for Offensive Language Identification
- [22]. Noureddine Doumi 2017. Extraction d'information à partir d'un texte arabe: extraction des entités nommées ET leurs relations sémantiques (page 35)
- [23]. Dhruvil Karani 1 sept 2018 Introduction à Mot Embedding ET Mot2Vec
- [24]. Ria Kulshrestha Nov 24, 2019 NLP 101: Word2Vec — Skip-gram and CBOW
A crash course in word embedding.
- [25]. Brett Grossfeld,directeur adjoint du marketing de contenu
- [26]. By Younes Benzaki 2 octobre 2018 Introduction à l'algorithme K Nearst Neighbors (K-NN).
- [27]. Raphaël Richard: Lazy Learning
- [28]. Dan Nelson Gradient Boosting Classifiers in Python with Scikit-Learn
- [29]. MARIE-JEANNE VIE 25 JUIN 2020 Gradient Boosting
- [30]. Aditi Mittal Publié le 23 janvier 2020 Deep Learning vs Machine Learning
- [31]. lambert .R 11 janvier 2019 Focus : Le Réseau de Neurones Convolutifs
- [32]. lambert .R 11 février 2018 Le Réseau de Neurones Artificiels ou Perceptron Multicouche.
- [33] By Younes Benzaki | 2 octobre 2018 Introduction à l'algorithme K Nearst Neighbors (K-NN)
- [34].JohnWolfé <A Brief History of Python> 4 mars 2018
- [35].Anaconda (distribution Python)
- [36]. Spyder (software) [https://fr.qaz.wiki/wiki/Spyder_\(software\)](https://fr.qaz.wiki/wiki/Spyder_(software))

Références

- [37]. UNIX Shell
- [38]. 9 mai 202, par des contributeurs MDN HTML (HyperText Markup Language)
- [39]. 12 août 2020 par contributeurs MDN
- [40]. JavaScript
- [41]. Fire base
- [42]. Angular, pour des applications SPA
- [43] Scikit-learn
- [44] keras
- [45]. Henri Michel google colab
- [46] Flask Documentation

Les fonctions mathématique

- (1). JOSEPH AZAR PUBLIÉ: 29-10-2019 INTRODUCTION À LA RÉGRESSION LOGISTIQUE AVEC PYTHON
- (2). Younes Benzaki | 2 octobre 2018 Introduction à l'algorithme K Nearest Neighbors (K-NN)
- (3). Younes Benzaki | 26 juillet 2017 Naive Bayes Classificateur pour Machine Learning
- (4). Aditi Mittal 12 oct 2019 understanding Rnn and Lstm
- (5). august 27, 2015 understanding LSTM networks
- (6). Afshine Amidi et Shervine Amidi Pense-bête de réseaux de neurones récurrents
- (7). Wikipédia
- (8). Stacey Ronaghan may 11, 2018 the mathematics of Decision tree , random Forest and feature importance Scikit-learn and Spark.
- (9). MARIUS BORCAN 6 may 2020 TF-IDF Explained And Python Sklearn Implementation
- (10). AMAN KHARWAL 8 November 2020 Introduction to Accuracy, F1 Score, Confusion Matrix, Precision and Recall in Machine Learning.

Figure 23: Posted on avril 16, 2018 by Admin M2 IESC Artificial Intelligence, Machine Learning, and Deep Learning: Same context, Different concepts

Figure 33 : Dan Feb 2020 Predicting weather using LSTM18

Figure 02, Figure 03: mars 2013 Extraction des informations et des connaissances

Figure 04 : Balkiss.hamad 26 février 2018 Différence entre les deux types d'apprentissage