People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of 8 May 1945-Guelma-
Faculty of Mathematics, Computer Science and Science of Matter
Department of Computer Science

Master Thesis

Specialty : Computer Science

Option : Science and technology of information and communication

Theme :

Une approche d'Analyse des Sentiments Appliquée au Dialecte Algérien

---

Presented by :

Menasri Ammar Chahir

supervised by :

Dr.Djalila Boughareb

Members Jury:

Dr.Abderrahmane Kefali

Dr. Lazhar Farek

June 2023

Remerciements

Tout d’abord, nous tenons à exprimer notre gratitude envers le bon Dieu de nous avoir accordé la détermination, la volonté et la force nécessaires pour mener à bien ce modeste travail. Nous souhaitons exprimer nos sincères remerciements et notre profonde reconnaissance à mon encadreur, Dr. DJAJILA BOUGHAREB, pour ses conseils, sa patience, sa disponibilité et son soutien inestimable tout au long de cette période. Nous tenons également à remercier chaleureusement les membres du jury d’avoir accepté d’évaluer notre travail et de l’avoir enrichi de leurs précieux commentaires. Nous sommes reconnaissants envers toutes les personnes qui nous ont apporté leur aide, de près ou de loin.

Nous voulons également adresser nos remerciements à tous nos enseignants qui ont veillé sur notre formation et nous ont transmis les connaissances nécessaires pour mener à bien ce projet. Leurs efforts et leur dévouement ont été essentiels dans notre parcours académique.

Enfin, nous tenons à exprimer notre gratitude envers nos proches, nos amis et notre famille pour leur soutien constant et leur encouragement tout au long de cette aventure. Leur présence et leurs encouragements ont été une source de motivation et de réconfort.

RÉSUMÉ

De nos jours, l'analyse de sentiment occupe une place prépondérante dans de nombreux domaines tels que la politique, la production et les services, pour n'en citer que quelques-uns. Les réseaux sociaux regorgent de textes dans lesquels les utilisateurs expriment librement leurs opinions sur une multitude de sujets, et ces opinions revêtent une importance considérable. Il est essentiel de comprendre le contenu véhiculé par ces textes, et un bon gestionnaire est celui qui sait écouter attentivement les opinions des citoyens. Dans cette optique, l'analyse des sentiments joue un rôle crucial dans la satisfaction des besoins des citoyens.
Dans le cadre de ce travail, notre objectif est de créer un modèle basé sur les réseaux de neurones récurrents LSTM (Long Short-Term Memory) qui sera en mesure d'analyser et de classifier un ensemble de publications issues des réseaux sociaux. Par la suite, nous développerons une application web permettant d'analyser les opinions des utilisateurs. Les deux classes que nous avons définies sont les sentiments positifs et négatifs. Cette étude se distingue parmi les rares travaux qui utilisent des modèles pour analyser les commentaires en utilisant le dialecte algérien. Les résultats obtenus sont très encourageants, avec une précision de 84,7 %, ce qui démontre la capacité du système à différencier un discours négatif d'un discours positif.

Mots Clés : fouille d'opinions, analyse des sentiments, fouille de texte, détection émotionnelle, web social, corpus annoté, Lexique de sentiments.

ABSTRACT

Today, sentiment analysis holds great importance in various fields such as politics, production, and services, among others. Currently, social media platforms are filled with texts in which internet users express their opinions on different subjects, and the value of their opinions is considerable. Understanding the content conveyed by these texts is essential. It can be said that a good manager is one who attentively listens to the opinions of citizens. In this regard, sentiment analysis is highly significant in meeting the needs of citizens.

In this work, we will create an LSTM model that will analyze and classify a set of posts from social media. Subsequently, we will develop a web application for analyzing user opinions. The defined classes are positive and negative. This work stands out among the few that utilize models to analyze comments using the Algerian dialect.

The obtained results are encouraging, with an accuracy of 84.7%, demonstrating the system's ability to distinguish between negative and positive discourse.

Keywords : opinion mining, Sentiment Analysis, text mining, emotional detection, social web, annotated corpus, Lexicon of Sentiment.

Table des matières

Remerciements.....i
RÉSUMÉ.....ii
ABSTRACT.....iii
Table des matières.....iv
Liste des Figures.....vi
Liste des Tableaux.....vii
INTRODUCTION GÉNÉRALE.....ix
1 CHAPITRE 1 :ANALYSE DES SENTIMENTS EN DIALECTE ALGÉRIEN.....1
1.1 Introduction.....1
1.2 Analyse des sentiments :.....1
1.2.1 Types de sentiments :.....2
1.3 Approches de classification des sentiments :.....2
1.3.1 Approche lexicale :.....2
1.3.2 Approche d'apprentissage automatique :.....3
1.3.2.1 Apprentissage supervisé :.....3
1.3.2.2 Apprentissage non supervisé :.....9
1.3.2.3 Apprentissage par renforcement :.....9
1.3.3 Approches hybrides :.....11
1.4 Dialecte Algérien et ses particularités :.....11
1.5 Spécificités orthographiques du dialecte Algérien :.....12
1.5.1 Conjugaison en dialecte Algérien :.....12
1.5.2 Négation en dialecte algérien :.....13
1.6 Travaux connexes.....14
1.6.1 Analyse de sentiments - cas de MSA (Arabe moderne standard :.....15
1.6.2 Analyse de sentiments - cas du dialecte algérien écrit en arabe :.....18

1.6.3 Analyse de sentiments - cas de traduction du dialecte :.....19
1.6.4 Analyse de sentiments - cas du dialecte algerien ecrit en latin :.....22
1.6.5 Analyse de sentiments - cas du dialecte Algérien écrit en arabe et en latin :.....24
1.7 Discussion.....25
1.8 Conclusion :.....27
2 CHAPITRE 2 :APPROCHE PROPOSEE ET METHODOLOGIE.....28
2.1 Introduction.....28
2.2 Approche Proposée :.....28
2.3 Description des Etapes :.....29
2.3.1 Collection de Données :.....29
2.3.2 Prétraitement.....29
2.3.3 Construction du modèle :.....31
2.4 Résultats et discussion :.....36
2.5 Conclusion :.....37
3 CHAPITRE 3 : IMPLÉMENTATION.....39
3.1 Introduction.....39
3.2 Ressources utilisées :.....39
3.3 Exemples de codes sources :.....41
3.4 Conclusion :.....46
CONCLUSION GÉNÉRALE.....47
Références.....48

Liste des Figures

Figure 1: Cadre d’un Rnn simple.....5
Figure 2: LSTM.....7
Figure 3: Mots vides.....30
Figure 4: Précision de l’entraînement et de validation.....32
Figure 5: Perte de l’entraînement et de validation.....33
Figure 6: courbe d’apprentissage.....35
Figure 7: Appel des Bibliothèques.....41
Figure 8: Lire la Dataset et le dictionnaire des mots vides.....42
Figure 9: fonction de nettoyage.....42
Figure 10: Architecture du modèle LSTM.....43
Figure 11: Importation des bibliothèques.....43
Figure 12: Chargement du modèle.....44
Figure 13: la fonction de nettoyage.....44
Figure 14: fonction de traduction.....44
Figure 15: Exemple applicatife.....45

Liste des Tableaux

Tableau 1: Tableau comparatif des méthodes NB et BN.....4
Tableau 2: Tableau comparatif des méthodes KNN, DTa et Rule based.....8
Tableau 3: Comparaison des méthodes de clustering 10
Tableau 4: Conjugaison du verbe ‘aimer’ au présent, passé et impératif.....12
Tableau 5: Les pronoms COD et COI du dialecte algérien.....14
Tableau 6: Les pronoms COD et COI dans le dialecte Algérien.....16
Tableau 7: Les résultats de classification par la méthode d’évaluation : validation croisée.....17
Tableau 8: Les résultats de classification par apprentissage.....18
Tableau 9: Nombre de commentaires par thème.....19
Tableau 10: Précision en utilisant ou non le module de calcul de similarité de phrases courantes.....19
Tableau 11: Statistiques relatives aux corpus de test.....21
Tableau 12: Résultats expérimentaux avec les deux lexiques utilisés.....22
Tableau 13: Données du dataset.....23
Tableau 14: Résultats des classificateurs sur les datasets originales et transcrits..24
Tableau 15: Exactitude et scores de précision, rappel et F1 en moyenne pour les ensembles de données Narabizi.....25
Tableau 16: Résultat du modèle.....34
Tableau 17: Résultats des modèles proposés dans [40].....35
Tableau 18: Exemple des résultats d’annotation obtenu par notre modèle.....37

INTRODUCTION GÉNÉRALE

À présent ,notre mode de vie repose largement sur l’accès à l’information numérique, qui est de plus en plus répandue grâce au développement du Web 2.0. Les internautes partagent de plus en plus de contenu, expriment leurs opinions et communiquent sur divers sujets, que ce soit dans des groupes de discussion, des blogs, des forums ou des sites d’évaluation de produits. Internet est devenu un outil essentiel pour les échanges personnels et professionnels, et les opinions publiées en ligne ont un impact significatif sur les utilisateurs. Des sondages ont révélé que la majorité des utilisateurs (80 %) recherchent déjà des avis en ligne sur un produit ou un service, et qu’ils sont prêts à payer deux fois plus cher pour un produit bénéficiant d’un avis positif par rapport à un autre [1].

Dans ce contexte, l’analyse de sentiments joue un rôle crucial dans la recherche, le marketing et l’industrie, car elle permet de comprendre les opinions des clients et des utilisateurs et d’ajuster les stratégies en conséquence. Avec des millions de commentaires publiés quotidiennement sur Internet et les réseaux sociaux, l’information disponible est devenue une mine d’or pour les entreprises.

Cependant, l’analyse de sentiments en arabe dialectal algérien présente des défis et des particularités en raison de la complexité de la langue et de la diversité des dialectes utilisés dans la région. Prenons l’exemple suivant : "cheft telephone jdid li khroj, raw3a khdemt bih 3jebni sah aw yestahel", qui se traduit en français par "Tu as vu le nouveau téléphone qui est sorti, il est génial, je l’ai utilisé et j’ai aimé, ça vaut la peine". Dans cet exemple, on peut constater la complexité du dialecte algérien, car une seule phrase peut contenir différents types d’écriture, que ce soit en arabe ou en latin.

Nous pouvons dire que des sentiments positifs ont été exprimés grâce aux mots "raw3a", "3jebni" et "yestahel", qui se traduisent par "génial", "aimé" et "vaut la peine". L'analyse de ce type de contenu pourrait aider une entreprise à comprendre les sentiments positifs des clients, ce qui leur permettrait de mettre en avant ce téléphone à travers une campagne publicitaire intensive en utilisant ces avis positifs.

Dans ce contexte, le but de ce travail est d'étudier les opinions et les sentiments exprimés sur les réseaux sociaux en arabe dialectal algérien.
Le modèle proposé dans ce mémoire pour l'analyse de sentiment est basé sur une approche d'apprentissage supervisé avec un réseau de neurones récurrents (RNN) utilisant des cellules LSTM (Long Short-Term Memory). L'objectif est de modéliser les dépendances à long terme et d'offrir une plus grande flexibilité et généralisation.

Organisation du mémoire :

Le mémoire est divisé en trois chapitres :

Le premier chapitre présente une introduction détaillée à l'analyse de sentiments, en exposant les différentes approches existantes, leur comparaison, ainsi que leurs avantages et limites.
Le deuxième chapitre se concentre sur la conception de notre système d'analyse de sentiments en arabe dialectal algérien, en expliquant comment nous avons conçu le système, les caractéristiques de notre corpus de données et le traitement des données brutes pour l'analyse de sentiments.
Le troisième chapitre présente en détail notre modèle d'analyse de sentiments, en expliquant comment nous avons utilisé les différentes approches pour développer notre modèle, les caractéristiques du modèle, ainsi que les tests et validations réalisés.

CHAPITRE 1 :ANALYSE DES SENTIMENTS EN DIALECTE ALGÉRIEN

1.1 Introduction

Le premier chapitre de ce travail se concentre sur l'analyse des sentiments sur les réseaux sociaux en dialecte algérien. Tout d'abord, nous définissons l'analyse de sentiment et expliquons son importance dans les domaines de la recherche, du marketing et de l'industrie. Nous abordons ensuite les spécificités du dialecte algérien et comment nous allons analyser les opinions publiques sur les réseaux sociaux dans cette langue. Différentes approches d'analyse de sentiment et quelques les algorithmes utilise sont également présentées, ainsi que les types de sentiment qui peuvent être identifiés, tels que positif, négatif ou neutre. Nous comparons ces approches et discutons des travaux récents en matière d'analyse de sentiment sur le dialecte algérien. L'objectif de ce chapitre est de fournir une vue d'ensemble complète de l'analyse de sentiment et de ses applications, tout en mettant en évidence les particularités du dialecte algérien dans ce domaine.

1.2 Analyse des sentiments :

L'analyse de sentiment, également appelé opinion mining ou sentiment analysis, est une branche de la recherche en traitement automatique du langage naturel qui vise à identifier les sentiments, les opinions ou les évaluations exprimées dans des unités d'information telles que les mots, les phrases, les paragraphes ou les documents. Son

but est de déterminer si un texte est positif, négatif ou neutre en utilisant des algorithmes informatiques pour analyser les opinions de l'auteur [2]. En d'autres termes, l'analyse des sentiments est le processus automatisé de détermination si une expression écrite est positive, négative ou neutre, en se basant sur l'opinion de son orateur.

1.2.1 Types de sentiments :

Il existe plusieurs types de classification des sentiments en analyse de sentiments [3]. Voici quelques exemples :

-Classification binaire : les sentiments sont classés en deux catégories, positives et négatives.

-Classification ternaire : les sentiments sont classés en trois catégories,positives, négatives et neutres.

-Classification par émotions : les sentiments sont classés en fonction des émotions qu'ils expriment, telles que la joie, la colère, la tristesse, etc.

Classification par intentions : les sentiments sont classés en fonction des intentions qu'ils expriment, telles que l'approbation, le rejet, la critique, etc.

Classification par intensité : les sentiments sont classés en fonction de leur intensité, allant de très positif à très négatif.

1.3 Approches de classification des sentiments :

Il y a pratiquement trois approches pour qualifier le sentiment dans les textes : l'approche lexicale et l'approche par apprentissage automatique. Une troisième, hybride consiste en une combinaison de composants linguistiques et de modules de classification [4] .

1.3.1 Approche lexicale :

Un lexique est un dictionnaire prédéfini qui contient des termes associés à une certaine polarité, tels que négatif, neutre ou positif. Il existe deux types principaux d'approches basées sur le lexique : l'approche basée sur le corpus et l'approche basée sur le dictionnaire [2] .

-Approche basée sur un dictionnaire : C'est la collection de mots d'opinion qui sont collectés manuellement. Les mots d'opinion aident à former la liste des graines. Lorsque ont trouvé des nouveaux mots pendant la classification des sentiments, puis il est ajouté à la graine liste.

-Approche basée sur le corpus : C'est la collection de grande quantité de mots basés sur un sujet spécifique. Il aide à élargir la liste des graines [5]. elle utilise deux bases approche dite statistique et sémantique [6].

1.3.2 Approche d'apprentissage automatique :

L'apprentissage automatique (en anglais Machine Learning) est un sous-domaine de l'intelligence artificielle qui donne à un système une capacité de compréhension grâce à ses algorithmes. L'idée est d'apprendre des algorithmes à partir des données et de faire des prédictions avec ces données et par cela les ordinateurs apprennent à résoudre des tâches spécifiques,sans avoir besoin de les programmer. Il existe 3 catégories de l'apprentissage automatique [7] :

1.3.2.1 Apprentissage supervisé :

Après avoir présenté les données et les résultats souhaités aux ordinateurs, ils sont capables de faire des prédictions pour de nouvelles données d'entrée. Cette catégorie a deux méthodes :

-**La classification probabiliste :**

-La classification probabiliste est une méthode d'apprentissage supervisé qui permet de prédire la classe d'appartenance d'un objet à partir de ses caractéristiques en se basant sur des probabilités conditionnelles. Les deux algorithmes les plus couramment utilisés pour cette méthode sont le Naive Bayes (NB) et le Bayesien Network (BN),et aussi les RNN simple et les LSTM.

Naive Bayes : Méthode qui peut être enseignée ou utilisée sur des données à petite échelle et qui peut fournir des résultats prédicatifs en temps réel. Elle peut également aider à classer une classe,dont les résultats peuvent être utilisés en parallèle pour augmenter l'échelle de l'ensemble de données, en particulier dans les études de cas de données à grande échelle [8].

Bayesian Network : méthode graphique probabiliste représentant la relation entre des variables aléatoires. Ce modèle se compose d'un graphique acyclique dirigé et d'un ensemble de distributions de probabilités conditionnelles pour chacune des variables du réseau [9]. Le réseau a une structure étendue et il est facile d'ajouter de nouvelles variables. Le tableau 1.1 montre la comparaison entre les deux méthodes [10]

Methode	Avantage	Désavantage	Évaluation	Travaux d'analyse de sentiments ayant utilisés la methode
Naive Bayes (NB)	Utile pour l'extraction de phrases subjectives Interprétation facile	Implémentationdifficile Caractéristiques indépendantes	Technique efficace malgré la simplicité	Élection présidentielle américaine (2014) [11]
Bayesian Network (BN)	-Facile à comprendre dans des domaines complexes -Peut faire face à un nombre limité de séquences de variables	Peu d'efforts pour la construction du modèle	Bonne précision	prédire les émotions humaines [12]

Tableau 1: Tableau comparatif des méthodes NB et BN

-**RNN :**

Un réseau neuronal récurrent (RNN) est un type de réseau neuronal profond pour la modélisation de séquences [13]. Dans les RNN, les connexions entre les neurones génèrent un graphe orienté. En raison de son état interne, les RNN peuvent traiter des séquences d'entrée . Chaque sortie du RNN est calculée en exécutant à plusieurs reprises la même fonction pour chaque instance. Cela permet de calculer les performances sur la base de tous les calculs précédents. La longueur du pas de temps est calculée à partir de la longueur de l'entrée dans l'architecture RNN[14].

-Le fonctionnement des RNN repose sur la réutilisation des états cachés, ces états caches sont des étapes de la séquence.Chaque neurone récurrent reçoit en entrée à la fois les données courantes et l'état caché précédent, puis génère un nouvel état caché. Cette récurrence permet aux RNN de capturer les dépendances à long terme, en prenant en compte l'historique des données séquentielles.

-Calcul de l'état cache : h(t)= f(w(x,y)*h(t-1)+w(w,h)*x(t)

h(t) :état cache a l'instant t f : fonction d'activation non

linéaire w(x,y) :matrice de poids des connexion récurrentes

h(t-1) :état cache précèdent h(w,h) :matrice de poids des

entre l'entrée et l'état cache x(t) : donne d'entrée a l'instant t

-Calcul de la sortie (prédiction) :

y(t)=w(h,y)*h(t) h(t) :

sortie a l'instant t

w(h,y) :matrice des connexion entre a l'état cache et la sortie La

figuer suivante montre cadre de travail des RNN.

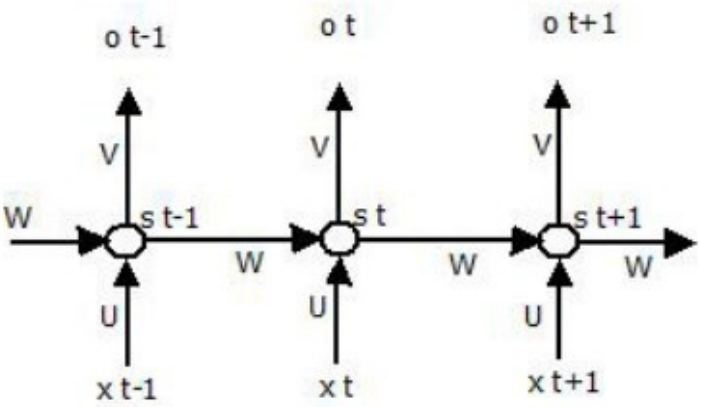

Figure 1: Cadre d'un Rnn simple

-**LSTM :**

Le réseau de mémoire à court terme (LSTM) est une architecture de réseau de neurones profond basée sur des RNN dans laquelle des portes oubliées sont utilisées pour éliminer le problème du gradient qui explose ou qui disparaît. Contrairement aux architectures de réseau de neurones récurrentes traditionnelles, LSTM permet l'erreur rétropropagation à travers un nombre fini de pas de temps [15]. Une unité LSTM typique

se compose d'une cellule et de trois types de portes : une porte d'entrée, une porte de sortie et une porte d'oubli. La cellule détermine quelles informations doivent être stockées et quand les unités doivent accéder aux informations en fonction de l'ouverture et de la fermeture de la porte opérations. La transition LSTM a été poursuivie sur la base des équations données ci-dessous [16]

$$f_t = \sigma_g\,(W_f \times x_t + U_f \times h_{t-1} + b_f)$$

$$i_t = \sigma_g\,(W_i \times x_t + U_i \times h_{t-1} + b_i)$$

$$o_t = \sigma_g\,(W_o \times x_t + U_o \times h_{t-1} + b_o)$$

$$c'_t = \sigma_c\,(W_c \times x_t + U_c \times h_{t-1} + b_c)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot c'_t$$

$$h_t = o_t \cdot \sigma_c(c_t)$$

xt : vecteur d'entrée de l'unité LSTM ft :
vecteur d'activation pour la porte oubliée il :
vecteur d'activation pour la porte d'entre ot :
vecteur d'activation de la porte sortie ht : :
vecteur d'état caché ct : l'état de la cellule
vecteur W : matrices de poids b :
paramètres du vecteur de biais
La figure suivante montre l'architecture de LSTM .

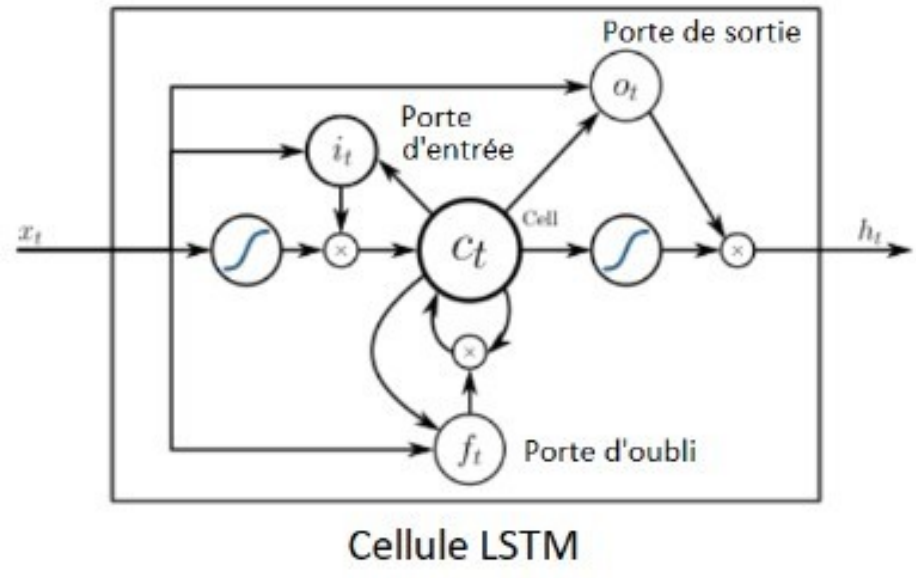

Figure 2: LSTM

La classification non-probabiliste : est souvent utilisée pour la détection de sentiment binaire en raison de la nature des SVM qui sont des classificateurs binaires. Pour pouvoir effectuer une classification multiclasses, le problème doit être transformé en un ensemble de problèmes de classification binaire [9] comme les algorithmes de KNN, Decison tree, Rule based.

-Nearest Neighbor (KNN) : KNN est l'un des plus populaires exemples de méthodes basées sur l'apprentissage [17]. Dans cette méthode, K est le nombre de voisins considérés qui est généralement impair, et la distance à ces voisins est déterminée sur la base des intervalles euclidiens standard [18].

Decision Tree : La classification par arbre de décision donne à l'utilisateur final une meilleure option pour catégoriser des phrases positives et négatives. Il est accompli en comparant les éléments les plus fréquents générées par les règles dans les données de formation avec les éléments les plus fréquents dans les données de test,permettant une classification simple [19].

Rule-based : Dans les approches de classification Rule-based, le modèle produit est un ensemble de règles. La règle est un structure de connaissance qui relie l'information

connue à d'autres informations dérivées de leur part. Une comparaissions entre les méthodes [10] :

Algorithmes	Avantages	Inconvénients	Travaux d'analyse de sentiments ayant utilises la methode
K-nearest neighbor (KNN)	Rapide à entraîner Très sensible au type de mesure	— Classification lente s'il y a beaucoup d'entraînement	campagne electorale [20]
Decision Tree	Facile à comprendre et à utiliser Meilleure option pour la classifica- tion POS/NEG Bonne performance pour les grandes bases de données	— Ne peut pas être uti- lisé dans les petites bases de données	Critique de restaurant[21]
Rule-Based	— Facile à comprendre	Devient lent s'il y a beaucoup de phrases Efficacité limitée dans la procédure du texte	explore efficacement les causes des emotions[22]

Tableau 2: Tableau comparatif des méthodes KNN, DTa et Rule based

1.3.2.2 Apprentissage non supervisé :

Cette méthode et basé sur le clustering , de nombreux algorithmes ont été proposés pour le regroupement de données dans l'analyse des sentiments ,ces algorithmes peuvent être classés en deux classes principales[23] :partition clustering(clustering de partions) et hierarchical clustering.

Partition clustering : L'objectif du clustering de partition est de diviser les données de manière à ce que les données d'un cluster ont le plus de similitudes et d'autre part ont le plus de distance avec les données des autres cluster [23] . Les algorithmes utilisés dans cette methode sont : K-means et fuzzy C-means.

Hierarchical clustering : Le clustering hiérarchique est en fait un ensemble de clusters imbriqués qui s'est organisé en arbre. Dans cette méthode,les clusters ou les groupes sont autorisés a avoir des sous-groupes. Les méthodes de clustering hiérarchique peuvent être divisées en deux catégories principales en fonction de leur structure qui sont :-Divisive methode et Bottom-up ou la methode aglomerative.

Le tableau 1.3 montre une comparaissions entre les méthodes du clustering [10] :

1.3.2.3 Apprentissage par renforcement :

Cette méthode permet à un système d'apprendre en interagissant avec son environnement. Il utilise des récompenses et des punitions pour renforcer ou décourager les comportements souhaités ou non souhaités comme les algorithme du RNN(Recurrent Neural Network) Les réseaux de neurones récurrents sont des algorithmes d'apprentissage automatique qui ont récemment gagné en popularité. Contrairement à d'autres algorithmes, les RNN ont la capacité de prendre en compte les données d'entrée sous forme de séquences, ce qui en fait un choix naturel pour les problèmes impliquant des données séquentielles telles que la reconnaissance de la parole, la traduction automatique, ou encore la prédiction de séries temporelles. La principale idée des RNN est d'extraire des caractéristiques en combinant linéairement les données d'entrée, puis de modéliser la sortie en utilisant une fonction non linéaire de ces caractéristiques. Cette approche permet aux RNN de prendre en compte les relations temporelles et contextuelles entre les données d'entrée, ce qui peut améliorer la précision des prédictions.

Critères	**Partitionnement de clustering**	**Clustering hiérarchique**
Méthode de re-groupement	Divise les données en un nombre de groupes dis- tincts	Construit une hiérarchie de clusters en utilisant une approche ascendante ou descendante
Nombre de clusters	Doit être spécifié à l'avance	Non spécifié à l'avance, mais peut être déterminé après l'analyse des don- nées
Taille des clus-ters	Peut varier considérable- ment en fonction des don- nées	Plus uniforme en raison de la hiérarchie de regroupe-ment
Coût de calcul	Plus efficace pour de grandes bases de données	Plus coûteux en termes de calcul, surtout pour de grandes bases de données
Interprétation des résultats	Plus facile à interpréter pour un petit nombre de clusters	Plus difficile à interpréter en raison de la hiérarchie de regroupement
Applications courantes	Analyse de sentiments, segmentation de marché	Classification d'images, identification de régions géographiques similaires

Tableau 3: Comparaison des méthodes de clustering .

1.3.3 Approches hybrides :

Cette approche est appelée aussi classification semi-supervisée,elle combine les points forts de deux approches précédentes, il y a trois façon de faire. La première est d'exploiter les outils linguistiques pour élaborer le corpus puis classer les textes par un outil d'apprentissage supervisé. La deuxième façon est d'utiliser l'apprentissage automatique pour établir le corpus d'opinion nécessaire à l'approche basée sur lexique. La troisième façon est le conjointement des deux approches précédentes et la combinaison de leurs résultats soit par un système de vote soit par un algorithme d'apprentissage [24]. Exemple d'algorithme : S3VM (Semi-Supervised Support Vector Machine).

1.4 Dialecte Algérien et ses particularités :

L'arabe algérien est la première langue en l'Algérie, langue maternelle de 75 à 85% de la population et maîtrisée par 95 à 100% de la population algérienne. Ses orateurs l'appellent darja « dialecte» ou darija, à l'opposé de l'arabe littéraire. Le dialecte algérien, est un groupe d'Arabe Nord-Africain, dialectes mélangés avec différentes langues parlées en Algérie, le frottement de plusieurs langues à travers l'histoire de la région a produit un langage complexe et riche comprenant des mots, des expressions et des structures linguistiques, ces langues tel que le Berbère, le Français, l'Italien, l'Espagnol et le Turc ainsi que des autres langues romanes méditerranéennes. Le dialecte algérien est fortement influencé particulièrement par le français où on peut trouver la commutation de codes et emprunter [25].

Le dialecte algérien est principalement utilisé à l'oral pour la communication quotidienne (par exemple, dans la vie quotidienne et dans les séries télévisées en Algérie), mais il n'est pas enseigné dans les écoles et n'est pas utilisé dans les communications écrites officielles. Cependant, ces dernières années, il gagne en importance dans les médias sociaux et commence à être utilisé plus fréquemment à l'écrit. Dans cette partie, nous nous concentrerons sur les caractéristiques spécifiques du dialecte , après avoir présenté les caractéristiques orthographiques et morphologiques communes avec l'arabe standard moderne (ASM).

1.5 Spécificités orthographiques du dialecte Algérien :

Le darija fait appel à toutes les voyelles et consonnes utilisées par l'ASM. En plus de ces dernières, elle est enrichi par les langues des groupes ayant colonisé ou géré la population algérienne au cours de l'histoire du pays. Parmi les langues de ces groupes, citons le turc, l'espagnol, l'italien et plus récemment le français [26].

1.5.1 Conjugaison en dialecte Algérien :

De la même manière que dans n'importe quelle langue, la conjugaison en darija implique l'ajout d'un ensemble de préfixes et de suffixes à un lemme donné, qui varient en fonction du pronom utilisé. Ces affixes sont généralement les mêmes pour tous les verbes.Dans le tableau 1, nous présentons la conjugaison du verbe 'aimer' en darija aux temps les plus couramment utilisés : le présent et le passé composé de l'indicatif, ainsi que l'impératif. Il convient de noter qu'en arabe, il y a deux formes pour la deuxième personne du singulier en fonction du sexe de la personne à qui l'on s'adresse.

Verbe	Présent	Traduction	Passé	Traduction	Impératif	Traduction
حب	نحب	J'aime	حبيت	J'ai aimé	-----	-----
	تحب	Tu aimes	حبيت	Tu as aimé	حب	Aime
	تحبي	Tu aimes	حبيتي	Tu as aimé	حبي	Aime
	يحب	Il aime	حب	Il a aimé	-----	-----
	تحب	Elle aime	حبت	Elle a aimé	-----	-----
	نحبو	Nous aimons	حبينا	Nous avons aimé	نحبو	Aimons
	تحبو	Vous aimez	حبيتو	Vous avez aimé	حبو	Aimez
	يحبو	Ils aiment	حبو	Ils ont aimé	-----	-----

Tableau 4: Conjugaison du verbe 'aimer' au présent, passé et impératif.

1.5.2 Négation en dialecte algérien :

Le darija offre deux façons principales d'exprimer la négation :

1) en utilisant les lettres (ma ...)

2) en utilisant le mot (machi, ...)

Pour illustrer cela, prenons l'exemple de la phrase "Je l'aime", qui devient en darija (nhabo) pour le masculin et (nhabha) pour le féminin. La négation de cette phrase en DARIJA est "Je ne l'aime pas ", ce qui se traduit par (manhabouch). Ainsi, pour faire une analogie avec le français, la lettre (man) joue le rôle du mot " ne ", et la lettre (ch) joue le rôle du mot " pas ". COD et COI du dialecte algérien (clitiques pronominaux) : Les COD et COI sont également agglutinés aux verbes conjugués en DALG, au même titre que les pronoms personnels et la négation. Les pronoms COD et COI représentent des suffixes du verbe conjugué. Ces pronoms ont déjà été étudiés [27]. Il existe cependant un nombre plus important de pronoms que ceux cités dans ces travaux car L'agglutination des pronoms de base entre eux donne naissance à de nouveaux suffixes. Nous récapitulons dans le tableau , l'ensemble des COD et COI de base.

COD	Exemple	Traduction	COI	Exemple	Traduction
ني	تحبيني	Tu m'aimes	لي	قولي	Tu me le dis
ك	كرهتك	Je t'ai détesté	لك	قولتلك	Je te l'ai dit
ه هو و	كرهتو حبيته	Je l'ai détesté Je l'ai aimé	لو	نقولو	Je lui dis
ها	كرهتها	Je l'ai détesté	لها	قولتولها	Je le lui ai dit
نا	كرهتونا	Vous nous avez détestés	لنا نا	قولتونا	Tu nous l'as dit
كم	حبيناكم	Nous vous avons aimés	لكم	قولتلكم	Je vous l'ai dit
هم	كرهتوهم	Vous les avez détestés	لهم	قوللهم	Dis-leur

Tableau 5: Les pronoms COD et COI du dialecte algérien

1.6 Travaux connexes

Dans le présent contexte de notre travail, il existe deux types de travaux. Le premier type, consiste à créer un outil d'analyse qui prend en entrée un tweet et fournit en sortie la polarité de ce tweet. Un exemple d'outil en ligne gratuit pour cela est l'application twitter Sentiment (Sentiment140) [28]. Les développeurs de cet outil ont testé trois algorithmes qui ont montré des taux de réussite compris entre 80% et 83% [29]. Le deuxième type de travaux consiste à créer un modèle d'analyse qui nous donne l'annotation d'un corpus (ou d'une partie de corpus) suivant l'une des approches citées auparavant, et là où notre travail s'inscrit.

1.6.1 Analyse de sentiments - cas de MSA (Arabe moderne standard :

Le travail cité dans le cas de MSA a été réalisé par Mohammed et al. (2014) dans le but de se concentrer sur l'aspect économique, plus précisément sur les revues de produits [30]. Ils ont commencé par la première étape, à savoir la collecte de données, puis ont procédé au prétraitement et à la classification. Leur corpus a été recueilli par eux-mêmes à partir de plusieurs sources en ligne telles que reviewzat.com et jawal123.com, etc. Il s'agit d'un ensemble de documents textuels, chaque document représentant un produit avec son type. Ils ont sélectionné cinq types de produits pour former ce corpus, à savoir les caméras, les PC portables, les téléphones portables, les tablettes et les télévisions. Le corpus compte 250 documents, 2812 phrases et 15466 mots. Deux évaluâtes ont travaillé sur l'étiquetage des opinions. Le premier était un expert en évaluation de produits, le deuxième était un spécialiste de la langue arabe. Un troisième évaluâtes non spécialiste a été utilisé uniquement pour valider les choix des deux autres annotateurs, afin d'obtenir un certain degré de fiabilité dans l'annotation. Ils ont rencontré quelques difficultés, notamment :

a) Les expressions émotionnelles (la joie, le malheur, la surprise...) pour résoudrecela, ils ont développé un petit convertisseur (symbole vers mot) qui fonctionne comme illustré dans le tableau .

Émoticône	Symbole	Commentaire	Commentaire après conversion de l'icône
	:)	ها تف روعة :)	هاتف روعة، سعيد
	^_^	جهاز ممتاز فاق كل التوقعات ^_^	جهاز ممتاز فاق كل التوقعات، مستمتع
	:(	الكاميرا سيئة للغاية):	الكاميرا سيئة للغاية، حزين
	:/	دقة الكاميرا دون المتوقع :/	دقة الكاميرا دون المتوقع، خاب أملي

Tableau 6: Les pronoms COD et COI dans le dialecte Algérien

b) L'élimination de la redondance des caractères tout en conservant la signification des mots.

Pour la classification, ils ont commencé par le stemming, qui est le processus de suppression de tous les préfixes et suffixes d'un mot pour produire le stem ou la racine. Ce processus est difficile en arabe, car par exemple, le stemming des mots (ra2i3) (merveilleux) et(morawi3) (terrible) donne le mot(raw3) (horreur), alors que ces deux mots ont des polarités inversées. Ils ont effectué leurs tests avec trois algorithmes de classification : les Machines à Vecteurs de Support (SVM), le Naïve Bayes (NB) et le K-plus proche voisin (KNN). Pour évaluer les performances de leur système, ils ont utilisé deux techniques : la validation croisée et la scission en pourcentage. La validation croisée : L'ensemble de données est divisé en K groupes aléatoires pour former les ensembles de test [31]. Dans ce cas, K est égal à 10, ce qui signifie que l'ensemble d'apprentissage est divisé en 10 groupes. Le modèle effectue l'apprentis- sage dix fois sur neuf parties d'entraînement et est évalué sur la dixième partie. Les dix évaluations sont ensuite combinées, et les résultats obtenus sont présentés dans le tableau suivant

.

	En termes de précision			En termes de rappel		
Corpus / Classificateur	KPPV	SVM	NB	KPPV	SVM	NB
Corpus à l'état brut	0,712	0,886	0,834	0,693	0,885	0,828
Corpus + light stemmer	**0,76**	0,904	0,861	**0,705**	**0,902**	0,857
Corpus + khoja stemmer	**0,76**	0,904	0,861	**0,705**	**0,902**	0,857
Corpus + normalisation	0,618	0,885	0,871	0,607	0,877	0,869
Corpus + normalisation + khoja stemmer	0,58	**0,912**	**0,876**	0,578	0,898	**0,873**
Corpus + normalisation + light stemmer	0,58	**0,912**	**0,876**	0,578	0,898	**0,873**

Tableau 7: Les résultats de classification par la méthode d'évaluation : validation croisée.

Ils ont conclu qui'il y avait un classificateur qui a obtenu de moins bonnes performances que les deux autres, à savoir le classificateur des k-plus proches voisins (KNN). Le SVM s'est révélé être le plus efficace avec différentes combinaisons de données. Scission en pourcentage : Dans cette méthode, le corpus est divisé de manière aléatoire en deux ensembles de données distincts. Le premier ensemble est l'ensemble d'apprentissage, tandis que le deuxième ensemble est l'ensemble d'évaluation. Dans leur cas, 80% des données sont utilisées pour l'ensemble d'apprentissage et le reste (20%) est destiné à l'ensemble de test. Les résultats obtenus sont présentés dans le tableau ci dessus .

	En termes de précision			En termes de rappel		
Corpus / Classificateur	KPPV	SVM	NB	KPPV	SVM	NB
Corpus à l'état brut	**0,803**	**0,946**	0,822	**0,776**	**0,939**	0,816
Corpus + light stemmer	0,799	**0,946**	0,881	0,735	**0,939**	0,878
Corpus + khoja stemmer	0,799	**0,946**	0,881	0,735	**0,939**	0,878
Corpus + normalisation	0,788	0,899	0,922	0,714	0,898	0,918
Corpus + normalisation + khoja stemmer	0,801	0,93	**0,946**	0,653	0,918	**0,939**
Corpus + normalisation + light stemmer	0,801	0,93	**0,946**	0,653	0,918	**0,939**

Tableau 8: Les résultats de classification par apprentissage.

1.6.2 Analyse de sentiments - cas du dialecte algérien écrit en arabe :

Le travail de M'hamed Mataoui et al. (2016) sur le dialecte algérien ont adopté une approche basée sur le lexique [32]. Pour construire leur modèle, ils ont créé trois lexiques : un lexique de mots clés, un lexique de mots de négation et un lexique de mots d'intensité. Ils ont également utilisé deux autres ressources : une liste d'émoticônes avec leurs polarités attribuées et un dictionnaire d'expressions courantes de dialecte algérien. Le lexique de mots clés contient 3093 mots, dont 713 mots positifs et 2380 mots négatifs. Les chercheurs ont collecté et annoté leur propre ensemble de données, qui comprend 7698 commentaires Facebook. Le tableau 1.9 présente la répartition des données collectées selon leurs thèmes.

Thèmes	Nombre de commentaires
économie	1705
politique	2422
société	1263
littérature et arts	1215
divers	1093

Tableau 9: Nombre de commentaires par thème.

Méthode	Précision
Sans utiliser le module de calcul de similarité de phrases courantes	76.68%
En utilisant le module de calcul de similarité de phrases courantes	79.13%

Tableau 10: Précision en utilisant ou non le module de calcul de similarité de phrases courantes

1.6.3 Analyse de sentiments - cas de traduction du dialecte :

Dans le travail de Imane Guellil et al 2017 , ils ont proposé une approche combinant l'utilisation de lexiques et un traitement spécifique de l'agglutination[33]. Ils ont utilisé deux lexiques annotés en sentiment et un corpus de test contenant 749 messages. Pour la construction du lexique, nous utilisons un lexique de sentiment en anglais en tant qu'entrée, car il existe plus de travaux sur l'analyse de sentiment dans cette langue . Chaque mot est ensuite traduit à l'aide d'une API de traduction, puis un lexique de sentiment est construit en extrayant chaque terme en dialecte algérien et en calculant son score. Nous expliquerons en détail ces étapes :

-Étape de traduction : ils ont fait appel a l'API GLOSBE qui prend un mot en anglais en entrée et renvoie un ensemble de mots en dialecte algérien. L'API utilise des utilisateurs natifs du dialecte pour effectuer la traduction. Ensuite, nous attribuons à tous les mots collectés le même score que le mot en anglais.

- Étape d'extraction du dialecte et calcul du score : Après la première phase, ils attribuons à chaque mot en dialecte algérien un score. Pour calculer ce score ensuit ils prenons la moyenne des scores de tous les mots anglais auxquels le mot en dialecte est associé.

- Étape de Calcul de la valence d'un message écrit en dialecte algérien : Cette étapeprend en entrée un message en dialecte algérien et renvoie sa valence et son intensité. Pour réaliser cela, ils appliquons un ensemble de traitements qui sont : - Suppression des espaces vides, des lettres longues (reconnues par tatweel).

-Suppression des exagérations.

-Suppression de certaines ponctuations telles que le "#" et espacement de certains points ("", "", "") attachés aux mots.

-Remplacement des caractères arabes par leurs codes Unicode pour traiter le phénomène lié à la présence de différentes lettres selon leur emplacement.

Ces traitements ont été inspirés des travaux de Cherif et al. (2015) Guellil et Azouaou (2017) Harrat et al(2016) et Saâdane et Habash (2015).

-Recherche des expressions du message dans le lexique de sentiments : Cette étape consiste à diviser la phrase en mots, puis à diviser chaque mot en préfixe + lemme + suffixe. Le préfixe et le suffixe sont ensuite supprimés, et seul le lemme est conservé. ils appuyons sur le travail de (Cherif et al) . Pour vérifier si le lemme existe dans le lexique qu'ils avens construit.

Expérimentation : Dans cette section, ils présenter les expérimentations réalisées par les auteurs

Lexique : Pour la construction du lexique, ils ont utilisé deux lexiques anglais, SOCAL [34] et SentiWordNet [35] ,Pour le lexique SOCAL, nous avons fusionné les lexiques, ce qui a donné 6 769 termes avec des scores étiquetés entre -1 et 5 pour les termes négatifs, et entre +1 et +5 pour les termes positifs. Après la traduction à l'aide de l'API, ils ont obtenu le lexique final appelé SOCALALG, qui compte 2 375 termes.

Parmi ces termes, il y a 1 363 termes négatifs, 948 termes positifs et 64 termes neutres (avec un sentiment égal à 0). Pour le lexique SentiWordNet, ils ont construit un lexique en calculant la moyenne du sentiment de chaque terme. Ensuite, ils multiplié les valeurs de sentiment par 5 pour aligner l'échelle avec le lexique SOCAL. Le lexique obtenu contient 39 885 termes avec des scores étiquetés entre +0,05 et +5 pour les termes positifs, et entre -0,05 et -5 pour les termes négatifs. Après la traduction, nous avons obtenu le lexique final appelé SentiALG, qui comprend 3 408 termes en dialecte algérien. Parmi ces termes, il y a 1 856 termes négatifs, 1 539 termes positifs et 13 termes neutres.

Tests : ils ont utilisé deux corpus pour les tests. Le premier corpus, appelé PADIC, contient 323 phrases. Ce corpus est le seul corpus parallèle multi dialectal qui inclut également le dialecte algérien . Le deuxième corpus contient 426 termes extraits de la plateforme Facebook. Les résultats des tests sont présentés dans les tableaux suivants.

Corpus	PADIC			Facebook		
	Pos.	Nég.	Tout	Pos.	Nég.	Tout
Nbre messages	157	166	323	220	206	426
Nbre mots	849	952	1 802	1 711	1 735	3 446
Nbre mots/message	5,41	5,73	5,57	7,78	8,42	8,1
Nbre caractères/message	21,9	24,0	23,0	33,3	35,9	34,6
Nbre messages avec émoticônse	0	0	0	38	19	57

Tableau 11: Statistiques relatives aux corpus de test.

	Lexique utilisé	PADIC			Facebook		
		P	R	F1	P	R	F1
n-gramme (1)	SOCALALG	0,71	0,45	0,55	0,68	0,36	0,47
	SentiALG	0,73	0,45	0,56	0,67	0,38	0,48
n-gramme + prétraitement (2)	SOCALALG	0,72	0,46	0,56	0,74	0 ,42	0,53
	SentiALG	0,74	0,47	0,57	0,71	0,42	0,53
N-gramme + prétraitement + lemme (3)	SOCALALG	0,70	0,69	0,70	0,70	0,63	0,67
	SentiALG	0,75	0,74	0,74	0,69	0,63	0,66
n-gramme + prétraitement + lemme + passé (4)	SOCALALG	0,70	0,70	0,70	0,72	0,64	0,67
	SentiALG	0,75	0,74	0,74	0,69	0,64	0,66
n-gramme + prétraitement + lemme + passé + négation (5)	SOCALALG	0,75	0,74	0,75	0,68	0,61	0,64
	SentiALG	0,78	0,78	0,78	0,67	0,61	0,64

Tableau 12: Résultats expérimentaux avec les deux lexiques utilisés.

1.6.4 Analyse de sentiments - cas du dialecte algerien ecrit en latin :

Le travaille de Ahmed Cherif Mazari et Abdelhamid Djeffal (2022) a travaille sur l'analyse de sentiment sur des commentaires dialectaux extraits des médias sociaux[36] . Ces commentaires concernent la langue parlée en Algérie, écrite en caractères arabes et/ou latins, pouvant être en arabe standard moderne, en français ou en dialecte local , leur travaille est deviser en 4 étapes : Premièrement, la constitution d'un ensemble de données de sentiments en dialecte algérien composé de 11 760 commentaires collectés sur diverses plateformes de médias sociaux, le tableau dessus :

Dataset	Commentaires	Polarity
Facebook	1300	Positive (51.96%)
YouTube	4561	Positive (70.66%)
Twitter	250	Positive (3.83%)
Total	6111	Positive (25.51%)
Facebook	1700	Negative (48.04%)
YouTube	3749	Negative (48.04%)
Twitter	200	Negative (48.04%)
Total	5649	Negative (48.04%)
Total	11760	-

Tableau 13: Données du dataset

Deuxièmement, la création des modèles Skip-Gram et CBOW par Word2vec à partir d'un corpus contenant 466 424 commentaires, ces derniers étant utilisés pour enrichir l'ensemble de données de sentiments en incluant des mots sémantiquement similaires. Troisièmement, la proposition d'un ensemble d'étapes de prétraitement adaptées au traitement des textes dialectaux comme la normalisation et la suppression des mot vide.

Enfin, la mise en œuvre et le test de différents classificateurs d'apprentissage automatique (SVM, Naive Bayes avec ses trois variantes (Bernoulli NB, Gaussian NB et Multinomial NB)) et de deux architectures d'apprentissage profond (CNN, RNN) pour évaluer et comparer l'ensemble de données dans sa version originale, dans une version transcrite en caractères latins, puis dans une version enrichie sémantiquement par les modèles Word2vec. Les expériences ont atteint des performances de classification des sentiments sur le "jeu de données transcrites en caractères latins" avec des précisions de (MNB : 84,21%, CNN : 64,11%) et sur "le jeu de données transcrites et enrichies par les modèles Word2vec" avec des précisions de (SVM : 83,70%,RNN : 65,21%). Dans le tableau suivant résume tout les résultat finale :

Classifier	Original dataset							Transcribed dataset						
	Without W2V	Enhanced by Skip-Gram			Enhanced by CBOW			Without W2V	Enhanced by Skip-Gram			Enhancedby CBOW		
		N=3	N=6	N=9	N=3	N=6	N=9		N=3	N=6	N=9	N=3	N=6	N=9
BNB	67.10	69.30	65.39	58.79	67.65	67.88	68.34	79.76	81.55	81.34	82.50	83.21	81.38	81.30
MNB	67.15	69.17	69.90	70.83	67.38	67.81	67.76	**84.21**	82.41	81.55	81.90	83.28	82.14	81.00
GNB	22.70	33.22	38.72	41.49	28.81	29.47	31.23	77.40	78.72	78.20	79.10	78.03	78.23	77.74
SVM	**68.07**	69.99	70.66	**72.04**	68.30	68.52	68.96	83.02	82.92	82.51	82.32	**83.70**	83.21	83.11
CNN	**62.96**	62.25	53.91	55.71	60.97	**62.64**	61.48	64.11	62.51	64.05	64.26	62.72	63.15	**64.56**
RNN	62.11	59.43	61.48	59.69	62.25	60.97	60.71	61.60	61.74	62.90	60.92	**65.21**	61.71	64.18

Tableau 14: Résultats des classificateurs sur les datasets originales et transcrits.

1.6.5 Analyse de sentiments - cas du dialecte Algérien écrit en arabe et en latin :

Le dernier travail, celui d'Amine Abdaoui (2022) [37], où il a utilisé l'analyse de sentiment sur plus d'un million de commentaires récupérés grâce à l'API de Twitter , qui ont été postés par la grande majorité des Algériens. La collection des commentaires était soit écrite en arabe, soit en latin, soit les deux combinés. Après la collecte, un prétraitement a été effectué sur les données collectées en remplaçant toutes les mentions d'utilisateurs par @user, toutes les adresses e-mail par mail@email.com et tous les liens hypertexte par https ://anonymizedlink.com. Enfin, les données ont été séparées en deux catégories : données d'entraînement et données de test.

Le modèle qui a été développé utilise une architecture BERT (12 encodeurs,12 têtes d'attention et une dimension cachée de 768). Tout d'abord, il a entraîné un Tokenizer WordPiece [38] sur nos données d'entraînement avec un vocabulaire de 50 000

entrées. Ensuite, nous avons entraîné notre modèle de langue en utilisant la tâche de Modélisation du Langage Masquée . Comme les commentaires étaient courts, une probabilité de 25% a été utilisée au lieu de 15%, et la taille du lot a été fixée à 64 en raison de ses limites informatiques. L'entraînement de son modèle a pris plus de 9 jours pour effectuer 50 époques sur l'ensemble de nos données d'entraînement. Notre modèle a été créé à l'aide de PyTorch et téléchargé sur le HUB Transformers pour faciliter son utilisation. Nous l'avons nommé DZIRIBERT. Les résultats de notre modèle sur différentes bases de données sont présentés dans le tableau ci-dessus.

Modèle	mBERT	XLM-R	AraBERT	QARiB
Exactitude	74.2	79.9	73.8	79.5
Précision	75.2	80.9	73.0	79.1
Rappel	62.0	64.9	26.0	26.1
Score F1	33.3	26.5	27.0	28.1
Modèle	Camel-BERT-da	Camel-BERT-mix	MARBERT	DziriBERT
Exactitude	75.2	74.6	76.0	74.0
Précision	66.0	34.6	38.7	34.6
Rappel	69.1	38.2	43.8	37.5
Score F1	70.2	39.1	41.7	39.4

Tableau 15: Exactitude et scores de précision, rappel et F1 en moyenne pour les ensembles de données Narabizi

1.7 Discussion

Dans le contexte de l'analyse de sentiments, il existe deux types de travaux. Le premier type consiste à créer un outil d'analyse qui prend en entrée un tweet et fournit en sortie la polarité de ce tweet. Un exemple d'outil en ligne gratuit pour cela est l'application Twitter Sentiment (Sentiment140). Les auteurs de cet outil ont testé trois algorithmes qui ont montré des taux de réussite compris entre 80% et 83%.

Le deuxième type de travail consiste à créer un modèle d'analyse qui donne l'annotation d'un corpus en utilisant l'une des approches mentionnées précédemment. Dans le cas de l'analyse de sentiments pour le dialecte algérien écrit en arabe, Mohammed [26] ont réalisé une étude axée sur les revues de produits. Ils ont collecté

leur propre corpus de données à partir de différentes sources en ligne telles que reviewzat.com et jawal123.com. Le corpus comprend 250 documents, 2812 phrases et 15466 mots, représentant cinq types de produits tels que les caméras, les PC portables, les téléphones portables, les tablettes et les télévisions. Deux évaluateurs ont travaillé sur l'étiquetage des opinions, avec un troisième évaluateur non spécialiste pour valider les choix des deux autres annotateurs.

Pour la classification, ils ont utilisé le stemming pour supprimer les préfixes et suffixes des mots et produire la racine. Ils ont testé trois algorithmes de classification : les Machines à Vecteurs de Support (SVM), le Naïve Bayes (NB) et le K-plus proche voisin (KNN). Ils ont évalué les performances de leur système en utilisant la validation croisée et la scission en pourcentage du corpus. Le SVM s'est révélé être le plus efficace avec différentes combinaisons de données.

Dans le cas de l'analyse de sentiments pour le dialecte algérien écrit en arabe, M'hamed Mataoui [28] ont adopté une approche basée sur le lexique. Ils ont construit trois lexiques : un lexique de mots clés, un lexique de mots de négation et un lexique de mots d'intensité. Ils ont collecté et annoté leur propre ensemble de données à partir de commentaires Facebook. Ils ont obtenu de meilleurs résultats en utilisant un module de calcul de similarité de phrases courantes.

Dans le cas de la traduction du dialecte algérien, Imane Guellil [29] ont proposé une approche combinant l'utilisation de lexiques et un traitement spécifique de l'agglutination. Ils ont utilisé deux lexiques annotés en sentiment et un corpus de test contenant 749 messages. Ils ont utilisé l'API GLOSBE pour la traduction des mots de l'anglais vers le dialecte algérien. Ensuite, ils ont calculé le score de chaque mot en dialecte algérien en utilisant les scores des mots anglais associés. Ils ont appliqué un ensemble de traitements spécifiques pour calculer la valence d'un message en dialecte algérien et rechercher les expressions dans le lexique de sentiments.

Ces travaux présentent différentes approches et techniques utilisées pour l'analyse de sentiments dans le contexte du dialecte algérien écrit en arabe. D'après notre analyse de ces travaux, nous constatons que l'utilisation des SVM est présente dans la plupart des études, mais cela ne garantit pas toujours les meilleurs résultats, notamment sur

de grandes bases de données. En revanche, les RNN ont montré de bons résultats dans ce type de problèmes.

1.8 Conclusion :

Dans ce chapitre nous avons abordé l'analyse des sentiments sur les réseaux sociaux en dialecte algérien. Nous avons commencé par définir l'analyse de sentiment et expliquer son importance dans la recherche, le marketing et l'industrie. Ensuite, nous avons discuté des spécificités du dialecte algérien et comment nous allons analyser les opinions publiques dans cette langue sur les réseaux sociaux. Différentes approches d'analyse de sentiment ont été présentées, ainsi que les algorithmes utilisés et les types de sentiments qui peuvent être identifiés (positif, négatif ou neutre). Nous avons comparé ces approches et discuté des travaux récents en matière d'analyse de sentiment sur le dialecte algérien.

L'objectif de ce chapitre était de fournir une vue d'ensemble complète de l'analyse de sentiment et de ses applications, tout en mettant en évidence les particularités du dialecte algérien dans ce domaine. L'analyse de sentiment, également appelée opinion mining ou sentiment analysis, est une branche de la recherche en traitement automatique du langage naturel qui vise à identifier les sentiments, les opinions ou les évaluations exprimées dans des unités d'information telles que les mots, les phrases, les paragraphes ou les documents. Son but est de déterminer si un texte est positif, négatif ou neutre en utilisant des algorithmes informatiques pour analyser les opinions de l'auteur.

Ce chapitre a presenté une brève synthese du domaine étudié. La suite du travail se concentrera sur l'analyse des sentiments en dialecte algérien. Il nous a permis de comprendre les spécificités linguistiques du dialecte algérien et les défis associés à l'analyse des opinions dans ce contexte.

CHAPITRE 2 :APPROCHE PROPOSEE ET METHODOLOGIE

2.1 Introduction

Dans ce chapitre, nous allons présenter l'approche proposée, et les étapes nécessaires à la réalisation d'un système d'analyse de sentiments des commentaires issues du dialecte Algerien. Nous présentons ci-dessous les étapes de réalisation allant de la collection de données jusqu'à la discussion des résultats obtenus.

2.2 Approche Proposée :

L'approche utilisée pour l'analyse de sentiment dans ce système est l'apprentissage supervisé avec un réseau de neurones récurrents (RNN) utilisant des cellules LSTM.

Le système prend en entrée un corpus de texte étiqueté et entraîne un modèle de classification binaire (positif, négatif) qui peut être utilisé pour prédire les classes d'étiquettes de nouveaux textes. Le modèle utilise une couche d'embedding pour représenter les mots sous forme de vecteurs denses. Les vecteurs d'embedding sont de taille 256 dimensions. Ensuite, une couche LSTM est ajoutée avec 16 unités LSTM. Cette couche LSTM permet au modèle de capturer les dépendances séquentielles dans les données textuelles.Enfin, une couche dense avec une fonction d'activation softmax est utilisée pour la classification binaire des sentiments. Cette couche a 2 neurones de sortie, correspondant aux deux classes de sentiments possibles : positif et négatif. Le modèle attribue une probabilité à chaque classe et prédit la classe avec la probabilité la plus élevée.

Comme on dit précédemment concernent les RNN simple qui peuvent traiter les donnes textuelles séquentielle ce qui les rend adaptés aux tâches de traitement du langage naturel (NLP) et conserver une mémoire des états précèdent qui lui donne l'avantage de capture une dépendance a long terme des donnes textuelles Cela est essentiel pour l'analyse de sentiments, où la signification d'un mot peut dépendre du contexte et des mots précédents. Mais le RNN simple a des limitation comme l'oublie des information a la rencontre de nouvelle information ou les donnes textuelle sont grande il rencontre une difficulté a capture la dépendance des donnes textuelles et le problème de l'explosion ou la disparition du gradient qui perturbent l'apprentissage [39], pour cella on ajouter un couche LSTM qui aide le RNN a surmonter ses difficulté grâce a ses trois porte qui aide le déterminer quelle information doit être stockes dans la mémoire , la cellule LSTM régler le flux d'information a travers la mémoire qui rend l'apprentissage et la dépendance a long terme plus efficace . Donc cette combinaison améliore la capacité du modèle à analyser les sentiments dans le texte de manière précise et cohérente.

2.3 Description des Etapes :

2.3.1 Collection de Données :

Dans nos expérimentations, nous avons utilisé DzSentiA qu'il s'agit d'une base de données contenant environ 50 000 Commentaires, comprenant 24 932 sentiments positifs et 24 932 négatifs de manière équilibrée, tous les textes sont en dialecte algérien. Cette base de données a été faite par Abdelli et al (2017) [40] et elle est disponible publiquement sur kaggle et github.

2.3.2 Prétraitement

Le prétraitement des données est l'ensemble des opérations utilisées pour préparer les données brutes avant leur utilisation dans un modèle d'apprentissage automatique ou dans une analyse statistique. La premier sous étape de prétraitent s'agit de la tokenization qui consiste à diviser le texte en séquences de mots ou de sous-mots appelés "tokens". Cela permet de convertir le texte en une séquence de symboles

compréhensibles pour les algorithmes de traitement de texte. A la fin de cette étape nous avons obtenu (176898) mots différent.

-La deuxième sous étape était le nettoyage de données, c'est-à-dire les mots inutiles du vocabulaire qui ne fournissent aucune information sur la polarité des textes sont éliminés. Alors, nous avons éliminé les balises HTML, les noms d'utilisateurs qui sont détectés par la balise <>, les hashtags détectés par la balise <> et les URLs. Puisque cette étude est basée sur des formes textuelles, les émoticônes sont également supprimés. Aussi, les lettres répétées plus de deux fois dans le même mot sont supprimées en les réduisant à une seule lettre, et toute la ponctuation (. * , ; : ! ?) et les caractères spéciaux sont également supprimés.

-Les mots vides sont également supprimés. Tout d'abord, nous avons opté à la suppression de toutes les lettres uniques car elles sont considérées comme des mots vides de sens. Ensuite, la suppression des mots vides pour le français et l'arabe (arabe standard moderne -MSA) en utilisant des listes prédéfinies. Pour le dialecte algérien, une liste spéciale de mots vides a été créé en relation avec ce travail, elle contient plus de 700 mots vide, la figure suivante montre quelques mots de cette liste.

Figure 3: Mots vides

-La troisième sous étape de pré-traitement était la division de données, où les données ont été divisées en ensembles d'entraînement et de test, dans la partie entraînement on a pris 80% de la base de données et pour la partie du test on a pris 20%

.

2.3.3 Construction du modèle :

Le modèle est construit en utilisant la bibliothèque Keras. Nous commençons par initialiser un modèle séquentiel. Ensuite, nous ajoutons une couche d'embedding qui représente les mots sous forme de vecteurs denses.Les mots sont représentés dans un espace d'embedding de dimension 256 car plus la taille est grande l'apprentissage du modèle est efficace . La longueur d'entrée est déterminée par la taille de notre ensemble de données.

Ensuite, nous ajoutons une couche LSTM à notre modèle. Dans notre cas, une seule couche LSTM est utilisée avec 16 unités LSTM. Cette couche permet au modèle de capturer les dépendances séquentielles dans les données textuelles. Le taux de dropout pour la couche LSTM est défini à 0.8, ce qui contribue à la régularisation du modèle. . Après la couche LSTM, une couche Dropout est ajoutée avec un taux de dropout de 0.5. Cela permet de désactiver aléatoirement certains neurones pendant l'entraînement, ce qui aide à prévenir le surajustement. Enfin, nous ajoutons une couche dense avec une activation softmax. Cette couche génère des probabilités normalisées pour chaque classe de sentiment (positif, négatif). Le modèle est compilé avec une fonction de perte de catégoriel cross-entropy, un optimiser Adam et la métrique d'accuracy pour évaluer les performances du modèle. Pour qui va suivre on va donner quelques justification de nos choix .

-Fonction de perte : on a utilisé la cross-entropie pour l'ajustement du poids du modèle pendant l'entraînement. L'objectif est de minimiser la perte, c'est-à-dire que plus la perte est petite, meilleur est le modèle. Un modèle parfait a une perte d'entropie croisée de 0.

-Optimiseur Adam : L'optimiseur Adam est un algorithme populaire utilisé dans l'apprentissage en profondeur qui aide à ajuster les paramètres d'un réseau de neurones en temps réel pour améliorer sa précision et sa vitesse. Adam signifie

Adaptive Moment Estimation, ce qui signifie qu'il adapte le taux d'apprentissage de chaque paramètre en fonction de ses gradients historiques et de son élan. Et Il a été démontré qu'il fonctionne bien sur un large éventail d'ensembles de données et peut aider les réseaux de neurones à converger plus rapidement et plus précisément pendant la formation [41].

-Activation softmax : comme dans notre base de donnes comporte juste 2 classe de sentiment alors la fonction softmax est décrite comme une combinaison de plusieurs fonctions sigmoïdes. Comme les fonctions sigmoïdes renvoient des valeurs comprises entre 0 et 1, qui peuvent être traitées comme des probabilités d'un point de données appartenant à une classe particulière. C'est pourquoi les fonctions sigmoïdes sont principalement utilisées pour les problèmes de classification binaire.

- Métrique d'accuracy : L'accuracy (précision) est une métrique couramment utilisée pour évaluer la performance d'un modèle de classification [42].

Ensuite, le modèle est construit à partir des données d'entraînement en utilisant la validation croisée et un ensemble d'options pour optimiser l'entraînement et éviter le désajustement. Résultats et discussion :

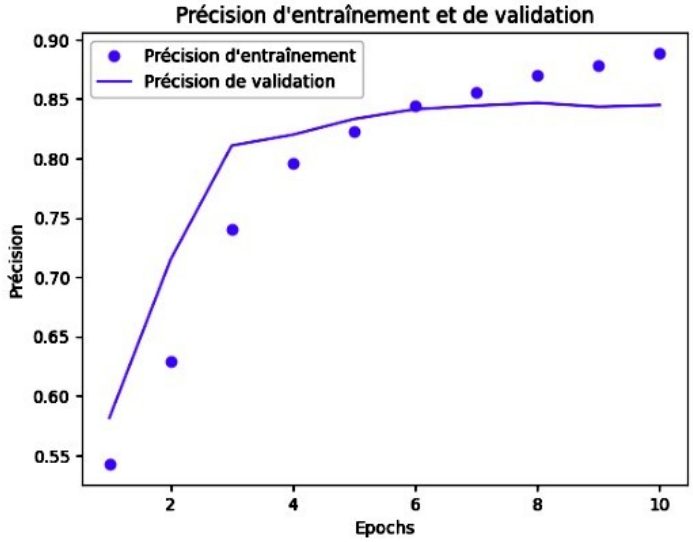

Figure 4: Précision de l'entraînement et de validation

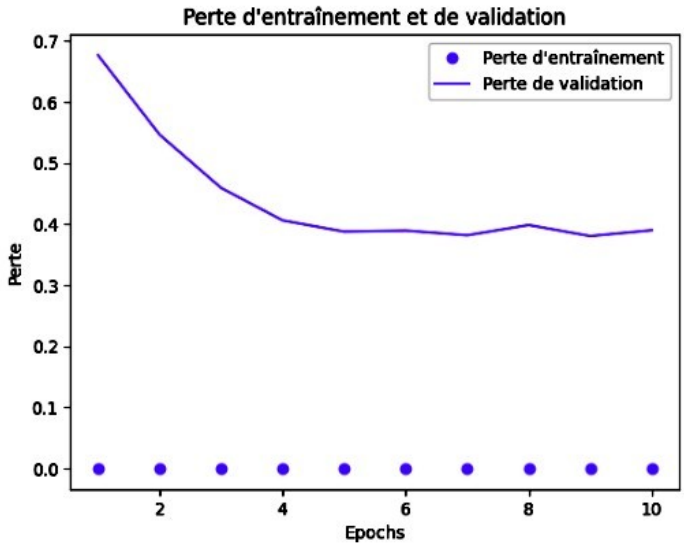

Figure 5: Perte de l'entraînement et de validation

Nous avons entraîné notre modèle sur un ensemble de données et évalué ses performances à l'aide de différentes mesures. Les résultats obtenus sont présentés en matière de précision, rappel et f-score.

Précisons : Les mesures de précision indiquent la capacité du modèle à prédire correctement les classes dans les différents ensembles de données. Une précision élevée dénote de bonnes performances de prédiction.

-Rappel : Le rappel (recall) est une mesure de performance qui évalue la capacité d'un modèle à identifier tous les exemples positifs dans un ensemble de données . -Formule de calcul : tp / (tp + fn) où tp est le nombre de vrais positifs et fn le nombre de faux négatifs. Le rappel est intuitivement la capacité du classificateur à retrouver tous les échantillons positifs. [42]

F-score :Le F-score (ou F-mesure) est une mesure de performance qui combine à la fois la précision et le rappel d'un modèle .

F-score = 2 * (Précision * Rappel) /(Précision + Rappel) Où : Précision = tp/ (tp+ fn)

Rappel = tp/ (tp+ fn)

Le tableau suivant montre les résultats générés par notre modèle en matière de précision, rappel , et F-score.

Métrique	Valeur
Rappel	82.75%
F-score	84.45%
Précision d'entraînement	92.9%
Précision de test	84.7%
Précision de validation	84.7%

Tableau 16: Résultat du modèle

En examinant les résultats par époque, nous observons une amélioration progressive de la précision tout au long de l'entraînement. Cela suggère que le modèle a appris à généraliser les schémas et à améliorer ses performances au fil du temps.Il convient de noter que la perte de validation diminue au fur et à mesure des époques, indiquant une convergence du modèle.

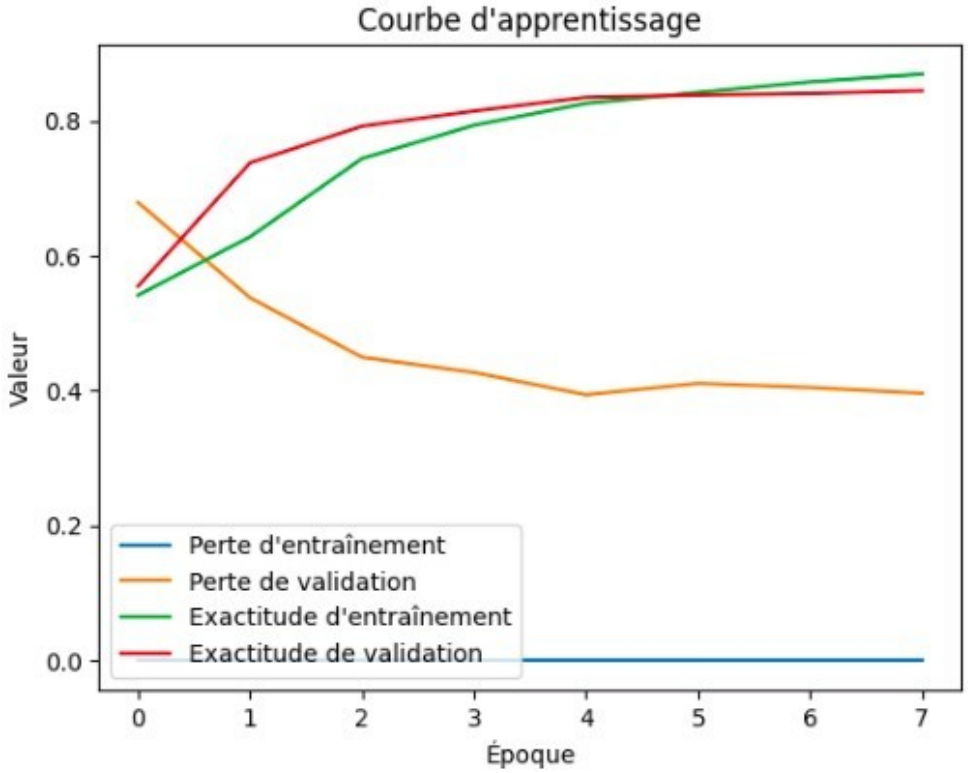

Figure 6: courbe d'apprentissage

Le tableau suivant montre les résultats de Abdelli [40] sur le même dataset "DZSentiA" :

Classifier	Accuracy	Precision	Recall	F1 score
SVM	0.86	0.89	0.82	0.85
LSTM	0.81	0.79	0.75	0.77

Tableau 17: Résultats des modèles proposés dans [40]

2.4 Résultats et discussion :

Deux modèles, à savoir le SVM et le LSTM, ont été utilisés dans l'étude précédente de Abdelli et al. [40]. Cependant, notre étude se concentre exclusivement sur le LSTM. Ils ont utilisé le modèle CBOW (Continuous Bag of Words) pour fusionner cinq principes en un seul fichier texte, puis ont appliqué l'embedding sur les données à l'aide de ce modèle CBOW. En revanche, nous avons utilisé un tokenizer comprenant 50 000 mots et appliqué directement un embedding de taille 256 sur nos données. 64 unités LSTM ont été utilisées comme hyperparamètre, tandis que nous n'en avons utilisé que 16. De plus, ils ont utilisé une taille de batch de 20 et effectué 100 000 itérations, avec une longueur maximale de séquence de 250. En ce qui nous concerne, nous avons utilisé une taille de batch de 100, effectué 10 itérations et maintenu la même longueur maximale de séquence. Nous avons obtenu une amélioration en matière de rappel avec un taux de 82.75 pour notre modèle contre 82% en utilisant SVM et 85% en utilisant LSTM, et un F-score de 84.45% contre 85% et 77% en utilisant SVM et LSTM respectivement.

Malgré le taux de précision élevé que nous avons obtenu, notre application a donné quelques erreurs de classification. Le tableau suivant présente un exemple des résultats de prédiction fournis par le système.

Exemple	Publication	Notre annotation	Résultat système
1	عيد سعيد و مبارك و كل عام و أنت بألف خير إن شاء هللا	1	1
2	سفيان فيغولي يتطوع إلحدى المائدات اإلفطارية للمسلمين في فرنسا ، برافو سفيان	1	1
3	هذا فعل ال اخالقي اين تعليم سيدنا محمد(ص)عندما كان جاره يهودي وعندما مرت جنازة يهودي باهلل عليكم ماذا فعل؟؟؟!!!!!!!	-1	-1
4	خدم خدما عيانا خلاه معجبدنيش	-1	-1
5	انا حاب كاس رايب و ربع كسرة خير من لفريت	0	-1
6	إذا عرفتو في اقل من 5 ثواني دير جام	0	-1
7	الجميع يشهد أن # الدوون هو االفضل ولكن عبودي الكتلوني لديه رأي آخر	1	1
8	مهابل ونالقاو	0	-1
9	يعمري هديك اتبسيمة	1	1
10	هذا االنسان ماهوش رجل وطني	-1	-1

Tableau 18: Exemple des résultats d'annotation obtenu par notre modèle.

Ces résultats témoignent de l'efficacité de notre modèle dans la tâche d'évaluation des performances. Ils sont prometteurs pour son application dans des scénarios réels nécessitant une classification précise.

Dans les exemples (5,6, 8), on constate que notre annotation diffère de celle du modèle, car ce sont des phrases neutres, tandis que notre modèle reconnaît uniquement deux types de sentiments (positif, négatif). Ainsi, si le modèle rencontre une difficulté dans l'analyse du sentiment d'une phrase, il tend à le classer comme négatif par défaut.

2.5 Conclusion :

En conclusion, le système d'analyse de sentiments proposé repose sur une approche d'apprentissage supervisé utilisant un réseau de neurones récurrents (RNN) avec des cellules LSTM. Les différentes étapes de sa réalisation ont été décrites en détail. L'ensemble de données utilisé pour l'entraînement du modèle est équilibré et comprend environ 50 000 entrées en dialecte algérien. Les avantages de cette architecture sont sa précision dans la prédiction de la polarité des textes et sa capacité à être étendue à d'autres tâches de traitement de texte. Cependant, les limites de cette architecture résident dans sa dépendance à la qualité et à la quantité des données d'entraînement.

CHAPITRE 3 : IMPLÉMENTATION

3.1 Introduction

Introduction Dans ce chapitre, nous allons presenter les outils utilisés pour la mise en œuvre du système d'analyse de sentiment que nous avons développé. Quelques codes sources sont également discuté ainsi que les interfaces développées

3.2 Ressources utilisées :

Dans notre expérimentation, nous avons utilisé un ordinateur de marque Dell équipé d'un processeur multi-cœur I5, avec une fréquence d'horloge de 2,20 GHz et une mémoire RAM de 4 Go. Le modèle que nous allons utiliser dans l'application a été développé sur Google Colab et l'application sera exécutée sur notre ordinateur personnel .

Pour la programmation de l'application, nous avons utilisé l'environnement Python. Python est un langage de programmation portable, dynamique, extensible, gratuit, syntaxe très simple, code plus court que C ou Java, multi thread, orienté objet, évolutif . . . [43]. Egalement, nous avons utilisé les packages suivants :

-NumPy (Numerical Python) : NumPy est une bibliothèque très populaire en Python utilisée pour effectuer des calculs numériques et des opérations mathématiques. Elle fournit des structures de données performantes telles que des tableaux multidimensionnels et des fonctions pour effectuer des opérations mathématiques avancées. NumPy est largement utilisé dans les domaines de la science des données, du calcul scientifique et de l'apprentissage automatique.

-Pandas : Pandas est une bibliothèque open-source basée sur NumPy qui offre des structures de données et des outils d'analyse de données faciles à utiliser. Elle permet de manipuler et d'analyser facilement des données tabulaires, comme des tableaux ou des fichiers CSV. Pandas fournit des fonctionnalités pour le nettoyage des données, la manipulation des données manquantes, le regroupement, le tri et la fusion des données. Elle est largement utilisée dans le domaine de l'analyse de données et de la préparation des données.

-Matplotlib : Matplotlib est une bibliothèque de visualisation de données en 2D pour Python. Elle permet de créer des graphiques, des diagrammes, des histogrammes, des diagrammes en barres, des nuages de points, etc. Matplotlib offre une grande flexibilité pour personnaliser les graphiques et les rendre esthétiquement agréables. Cette bibliothèque est souvent utilisée dans le domaine de la science des données, de la visualisation de données et de la génération de graphiques pour l'analyse de données.

-Scikit-learn : Scikit-learn est une bibliothèque d'apprentissage automatique opensource qui fournit des outils pour effectuer des tâches d'apprentissage automatique telles que la classification, la régression, le regroupement et la sélection de caractéristiques. Elle offre une implémentation simple et efficace d'une grande variété d'algorithmes d'apprentissage automatique, ainsi que des fonctions pour l'évaluation des modèles, la validation croisée et le prétraitement des données. Scikit-learn est largement utilisé dans le domaine de l'apprentissage automatique et de l'analyse prédictive.

-TensorFlow : TensorFlow est une bibliothèque d'apprentissage automatique développée par Google. Elle est principalement utilisée pour construire, entraîner et déployer des modèles d'apprentissage automatique, en particulier des réseaux neuronaux profonds. TensorFlow offre une grande flexibilité et une bonne performance, et elle prend en charge les calculs parallèles sur des unités de traitement graphique (GPU). Cette bibliothèque est très utilisée dans le domaine de l'apprentissage automatique, de l'intelligence artificielle et de la recherche en sciences des données.

-Re : Le module "re" est une bibliothèque standard de Python utilisée pour les expressions régulières. Il permet de rechercher et de manipuler des motifs de texte.

-Keras : Keras est une bibliothèque populaire pour le développement de modèles d'apprentissage automatique en Python et fournit des utilisateur pour prétraiter le texte, y compris la tokenisation des mots.

-Pickle : Pickle est un module de la bibliothèque standard de Python utilisé pour la sérialisation des objets. Il permet de sauvegarder et de charger des objets Python dans des fichiers.

-Arransia :Aaransia est une bibliothèque logicielle open source développée pour la translittération des langues arabes. Elle permet de convertir les textes arabes écrits avec une variante de l'alphabet arabe et perment de transformer l'arab ecrit en latin en une forme standardisée de l'arabe moderne standard (MSA).

-Streamlit : Streamlit est une bibliothèque utilisée pour créer des applications web interactives en Python. Elle permet de développer rapidement des interfaces utilisateur pour afficher des données, interagir avec des modèles et créer des tableaux de bord.

3.3 Exemples de codes sources :

Dans cette section, nous allons présenter quelques exemples de codes sources. La Figure 3.1 présente un morceau de code qui permet d'appeler les bibliothèques nécessaires pour compiler notre application.

```
import numpy as np
import pandas as pd
import re
import tensorflow as tf
from keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Embedding, Dropout
from sklearn.model_selection import train_test_split
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.utils.class_weight import compute_class_weight
```

Figure 7: Appel des Bibliothèques

La Figure 3.2 présente les instructions nécessaires pour lire le Dataset et le dictionnaire des mots vides.

```
data = pd.read_csv('/content/dataset.csv')


with open('/content/algerian_arabic_stopwords.txt', 'r') as f:
    stopwords = set(f.read().split())
```

Figure 8: Lire la Dataset et le dictionnaire des mots vides

La Figure 3.3 présente les instructions nécessaires de la fonction pour faire la suppression des caractère spéciaux et des mots vides de la dataset ansi que son application sur la dataset .

```
def clean_text(text: str, algerian_arabic_stopwords: List[str]) -> str:
    # Convert to string
    text = str(text)
    # Remove urls
    text = re.sub(r'http\S+', '', text)
    # Remove user mentions
    text = re.sub(r'@\w+', '', text)
    # Remove punctuation
    text = re.sub(r'[^\w\s]', '', text)
    # Convert to lowercase
    text = text.lower()
    # Split text into words
    words = text.split()
    # Remove stop words
    words = [w for w in words if not w in stopwords]
    # Join words back into sentence
    text = " ".join(words)
    return text

# Apply text cleaning to data
data['text'] = data['text'].apply(lambda x: clean_text(x, stopwords))
```

Figure 9: fonction de nettoyage

La Figure 3.4 présente l’architecture de notre modèle LSTM ainsi que ses paramètres et sa configuration.

```
# Model architecture
nb_lstm = 1
lstm_units = 16
lstm_dropout = 0.8
epochs = 10
dropout_rate = 0.5
batch_size = 100

model = Sequential()
model.add(Embedding(50000, 256, input_length=X.shape[1]))
for i in range(nb_lstm):
    model.add(LSTM(lstm_units, dropout=lstm_dropout, recurrent_dropout=lstm_dropout, return_sequences=True if i < nb_lstm-1 else False))
model.add(Dropout(dropout_rate))
model.add(Dense(2, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

# Define early stopping callback
early_stop = EarlyStopping(monitor='val_loss', patience=3)

# Define class weights
class_weights = {0: 1 / len(y_train[y_train[:,0] == 1]), 1: 1 / len(y_train[y_train[:,1] == 1])}

# Train the model
history = model.fit(X_train, y_train, validation_data=(X_test, y_test),
                    epochs=epochs, batch_size=batch_size, callbacks=[early_stop],
                    class_weight=class_weights)
```

Figure 10: Architecture du modèle LSTM

Dans les figures suivantes, nous vous montrerons quelques exemples du code de notre application ainsi que de l'application elle-même.

La Figure 3.5 présente l'importation des bibliothèques utilisées pour l'application.

```
import numpy as np
import pandas as pd
import streamlit as st
import re
from keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import load_model
import pickle
from typing import List
from aaransia import SourceLanguageError, transliterate
import re
from os import listdir
from os.path import isfile, join
from pyarabic import araby
```

Figure 11: Importation des bibliothèques

La Figure 3.6 présente le chargement du modèle ainsi que le tokenizer et les mots vides.

La Figure 3.7 présente la fonction de nettoyage des phrase

La Figure 3.9 présente la fonction de traduction du latin au Arab (MSA)

La figure 3.10 presente notre application web avec un example

```
model = load_model('FINALsentiment_modelss.h5')


with open('FINALsentiment_tokenizer.pkl', 'rb') as f:
    tokenizer = pickle.load(f)


with open('algerian_arabic_stopwords.txt', 'r') as f:
    stopwords = set(f.read().split())
```

Figure 12: Chargement du modèle

```
def clean_text(text: str, algerian_arabic_stopwords: List[str]) -> str:
    text = str(text)
    text = re.sub(r'http\S+', '', text)
    text = re.sub(r'@\w+', '', text)
    text = re.sub(r'[^\w\s]', '', text)
    text = text.lower()
    text = araby.strip_tashkeel(text)
    text = re.sub(r'\_+', ' ', text)
    text = re.sub(r'\ر+', 'ر', text)
    text = re.sub(r'\ا+', 'ا', text)
    text = re.sub(r'\وو+', 'و', text)
    text = re.sub(r'\هههه+', 'هههه', text)
    text = re.sub(r'\ةة+', 'ة', text)
    text = re.sub(r'\يي+', 'ي', text)
    text = re.sub(r'أ', 'ا', text)
    text = re.sub(r'آ', 'ا', text)
    text = re.sub(r'إ', 'ا', text)
    text = re.sub(r'ة', 'ه', text)
    text = re.sub(r'ى', 'ي', text)
    text = " ".join(text.split())
    words = text.split()
    words = [w for w in words if not w in stopwords]
    text = " ".join(words)
    return text
```

Figure 13: la fonction de nettoyage

```
def transliterate_arabic(text: str) -> str:
    try:
        transliterated_text = transliterate(text, source='al', target='ar')
    except SourceLanguageError:
        transliterated_text = ""
    return transliterated_text
```

Figure 14: fonction de traduction

FIGURE 3.8 – fonction de traduction.

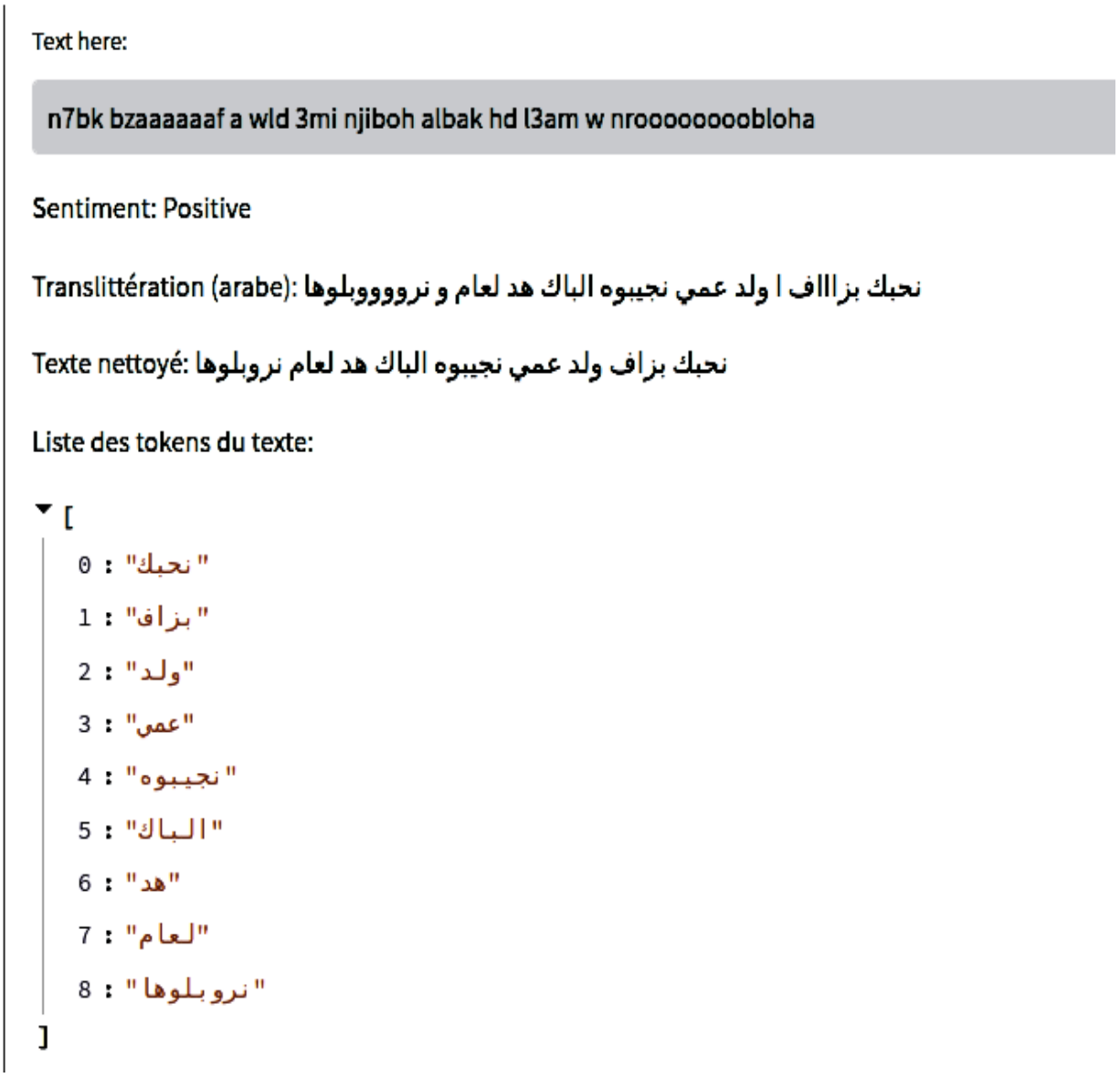

Figure 15: Exemple applicatife

3.4 Conclusion :

Dans ce chapitre nous avons présenté les outils utilisés pour développer notre système d'analyse de sentiment. Nous avons également discuté de quelques codes sources et des interfaces développées.Nous avons présenté quelques exemples de codes sources,tels que l'appel des bibliothèques nécessaires,la lecture du Dataset et du dictionnaire des mots vides, la fonction de nettoyage des données, ainsi que des exemples de mots vides et de l'architecture de notre modèle LSTM ,ce chapitre d'implémentation nous a permis de présenter les outils et les ressources utilisés pour développer notre système d'analyse de sentiment. Les codes sources et les interfaces développées sont essentiels pour comprendre et mettre en œuvre notre application.

CONCLUSION GÉNÉRALE

En conclusion, ce travail fournit une contribution importante pour les entreprises et les chercheurs qui s'intéressent à l'analyse de sentiments en arabe dialectal algérien. Les résultats obtenus et les implications discutées peuvent aider à mieux comprendre les opinions et les attitudes des gens sur différents sujets, ainsi qu'à prédire leur comportement.

L'approche développée utilise des techniques d'apprentissage automatique, notamment l'utilisation d'un modèle LSTM (Long Short-Term Memory) pour la classification des sentiments, les résultats obtenus ont démontré une bonne performance de l'approche proposée, avec une précision de 84.7, qui est assez satisfaisante pour la prédiction des sentiments des utilisateurs. Cela permet de mieux comprendre les opinions et les attitudes des individus qui utilise le dialecte algérien et peut être utilisé pour des applications telles que l'analyse des réseaux sociaux, le suivi de la réputation en ligne, et la recommandation de produits ou de services.

En ce qui concerne les perspectives des travaux futurs, il serait intéressant d'étendre l'approche proposée pour prendre en compte d'autres dialectes arabes, afin d'obtenir une couverture plus large et une meilleure compréhension des sentiments dans les différents contextes culturels, et L'enrichissement de Dataset par d'autres commentaires en dialecte algérien afin d'obtenir des résultats bien précis . De plus, l'amélioration de la performance du modèle en explorant d'autres architectures et en optimisant les hyperparamètres

Références

[1] Sébastien Gillot, Fouille d'opinions, Colloque du Master Recherche en Informatique,2010, 1-35.

[2] Liu B (2012) Sentiment analysis and opinion mining. Synthesis lectures on human language technologies.Morgan & Calypool Publishers, pp 1–167.

[3] Article écrit par Anthony DEMOGUE, Consultant, membre de la BCOM Data chez HeadMind Partners Digital,\\https://www.headmind.com/fr/text-mining-analyse-de-sentiments/

[4] Cynthia Van Hee, L'analyse des sentiments appliquée sur des tweets politiques : une étude de corpus, Faculté associée de linguistique appliquée Université Bruxelles Belgique,2013.)

[5] Keshtkar Fazel, Inkpen Diana., "A bootstraping method for extracting paraphrases of emotion expressions from texts" Comput Intell;vol. 0, 2012.
[6]Medhat, Walaa, Ahmed Hassan, and Hoda Korashy. "Sentiment analysis algorithms and applications: A survey" Ain Shams Engineering Journal 5.4 :1093-1113, 2014.

[7] Larbes A BDELKRIM, Amrani O KBA et Khantoul B ILEL. « Une Approche deep learning pour l'analyse des sentiments ». In : (2021).

[8] Wongkar M, Angdresey A (2019) Sentiment analysis using Naive Bayes algorithm of the data crawler: Twitter. InProceedings 2019 4th International Conference Informatics Computing ICIC 2019. \\https://doi.org/10.1109/ICIC47613.2019.8985884

[9] Sierra B, Lazkano E, Jauregi E, Irigoien I (2009) Histogram distance-based bayesian network structure learning: a supervised classification specific approach. Decis Support Syst 48(1):180–190

[10] Hemmatian, F., Sohrabi, M.K. A survey on classification techniques for opinion mining and sentiment analysis. Artif Intell Rev 52, 1495–1545 (2019).\\ https://doi.org/10.1007/s10462-017-9599-6

[11] Anjaria M, Guddeti RMR (2014) A novel sentiment analysis of social networks using supervised learning. Soc Netw Anal Min 4(1):1–15

[12] Ren F, Kang X (2013) Employing hierarchical Bayesian networks in simple and complex emotion topic analysis. Comput Speech Lang 27(4):943–968

[13] Li X, Wu X. Constructing long short-term memory based deep recurrent neural network for large vocabulary speech recognition. arXiv preprint. 2014.

[14] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9(8), 1735–1780.

[15] Lu, C., Huang, H., Jian, P., Wang, D., & Guo, Y. D. (2017, May). A P-LSTM neural network for sentiment classification. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 524-533). Springer, Cham

[16] Rojas-Barahona, L.M., 2016. Deep learning for sentiment analysis. Language and Linguistics Compass 10 (12), 701–719.

[17] Tsagkalidou K, Koutsonikola V, Vakali A, Kafetsios K (2011) Emotional aware clustering on micro-blogging sources. In: D'Mello S, Graesser A, Schuller B, Martin JC (eds) Affective computing and intelligent interaction. ACII 2011. Lecture notes in computer science, vol 6974. Springer, Berlin, pp 387–396

[18] Duwairi RM, Qarqaz I (2014) Arabic sentiment analysis using supervised classification. In: International IEEEconference on future internet of things and cloud (FiCloud), pp 579–583

[19] Kasthuri S, Jebaseeli AN (2020) An efficient decision tree algorithm for analyzing the twitter sentiment analysis. J Crit Rev 7(4):1010–1018

[20] Alfaro C, Cano-Montero J, Gómez J, Moguerza JM, Ortega F (2016) A multi-stage method for content classification and opinion mining on weblog comments. Ann Oper Res 236(1):197–213

[21] Liu B, Zhang L (2012) A Survey of Opinion Mining and Sentiment Analysis. In: Aggarwal C., Zhai C. (eds) Mining text data. Springer, Boston, MA, pp 415–463

[22] Gao K, Xu H, Wang J (2015) A rule-based approach to emotion cause detection for Chinese micro-blogs. Expert Syst Appl 42(9):4517–4528

[23] Li G, Liu F (2014) Sentiment analysis based on clustering: a framework in improving accuracy and recognizing neutral opinions. Appl Intell 40(3):441–452

[24] Damien Poirier et Françoise Fessant et Cécile Bothorel et Emilie Guimier de Neef et Marc Boullé, Approches Statistique et Linguistique Pour la Classification de Textes d'Opinion Portant sur les Films, Revue des Nouvelles Technologies de l'Information RNTI-E-17, 2010, 147-169.

[25] Harrat S., Meftouh K., Smaïli K., « Machine translation for Arabic dialects (survey) »,Information Processing & Management, 2017.

[26] Saâdane H., Habash N., « A conventional orthography for Algerian Arabic », Proceedings of the Second Workshop on Arabic Natural Language Processing, p. 69-79, 2015.

[27] Saâdane H., Habash N., « A conventional orthography for Algerian Arabic », Proceedings of the Second Workshop on Arabic Natural Language Processing, p. 69-79, 2015.

[28] Soumia Elyakoute Herma et Khadidja Saifia, Analyse des sentiments cas Twitter, Mémoire de master, Université de Ghardaia, 2016.

[29] Alec Go et Richa Bhayani et Lei Huang, http://twittersentiment.appspot.com/ .

[30] Mohamed Ali Sghaier et Houssem Abdellaoui et Rami Ayadi et Mounir Zrigui, Analyse de sentiments et extraction des opinions pour les sites e-commerce : application sur la langue arabe, CITALA, 2014, 57-61.

[31] J. Chiquet, Validation croisée pour le choix de paramère de méhodes, Module MPR -option modélisation, 2009.

[32] M'hamed Mataoui et Omar Zelmati et Madiha Boumechache, A Proposed Lexicon-Based Sentiment Analysis Approach for the Vernacular Algerian Arabic, Research in Computing Science, 2016, 55-68.

[33] Guellil, Imane & Azouaou, Faical & Saadane, Houda & Semmar, Nasredine. (2018). Une approche fondée sur les lexiques d'analyse de sentiments du dialecte algérien. TAL Traitement Automatique des Langues. 58. 41 à 65.

[34] Taboada M., Brooke J., Tofiloski M., Voll K., Stede M., « Lexicon-based methods for sentiment analysis », Computational linguistics, vol. 37, no 2, p. 267-307, 2011.

[35] Baccianella S., Esuli A., Sebastiani F., « Sentiwordnet 3.0 : an enhanced lexical resource for sentiment analysis and opinion mining. », LREC, vol. 10, p. 2200-2204, 2010.

[36] Ahmed Cherif Mazari , Abdelhamid Djeffal 'Sentiment Analysis of Algerian Dialect Using Machine Learning and Deep Learning with Word2vec' , 2022 , Informatica 67–78

[37] Abdaoui, Amine and Berrimi, Mohamed and Oussalah, Mourad and Moussaoui, Abdelouahab,DziriBERT: a Pre-trained Language Model for the Algerian Dialect ,2022, arXiv preprint arXiv:2109.12346

[38] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “Albert: A lite bert for self-supervised learning of language representations,” arXiv preprint arXiv:1909.11942, 2019.

[39] Alexander Rehmer,Andreas Kroll ,'On the vanishing and exploding gradient problem in Gated Recurrent Units', 2020 ,IFAC-PapersOnLine

[40] A. Abdelli, F. Guerrouf, O. Tibermacine and B. Abdelli, "Sentiment Analysis of Arabic Algerian Dialect Using a Supervised Method," 2019 International Conference on Intelligent Systems and Advanced Computing Sciences (ISACS), Taza, Morocco, 2019, pp. 1-6.

[41] Diederik P. Kingma, Jimmy Lei Ba, ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION, conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015

[42] Renuka Joshi, Interpretation of Performance Measures, \\https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures .

[43] Presentation du language 9812/python.html, Python, \\http://www.linux-center.org/articles/