République Algérienne Démocratique et Populaire
Ministère de l'Enseignement supérieur et de la Recherche Scientifique
Université de Blida 1
Faculté des Sciences
Département de l'informatique

Mémoire de Master

Spécialité Ingénierie des Logiciels

Thème :

Détection d'opinions dans des textes arabes basée sur l'apprentissage profond

Réalisé par :
Hamrouche Sana

Tekari Sarah

Promoteur :
M. Ferfera Soufiane

Devant le jury :

Mme. Mezzi Melyara Présidente

Mme. Boucetta Zouhel Examinatrice

Année universitaire : 2022/2023

Remerciements

Nous tenons tout d'abord à exprimer notre gratitude envers ALLAH, qui nous a accordé la santé, le courage, la persévérance et la sagesse nécessaires pour mener à bien ce travail.

Nous souhaitons aussi exprimer nos sincères remerciements à notre superviseur, monsieur Ferfera Soufiane, pour son soutien constant, ses conseils éclairés et son accompagnement tout au long de réalisation de ce mémoire.

Nous adressons nos remerciements aux membres du jury, pour avoir accepté d'évaluer ce mémoire.

Nous souhaitons ainsi exprimer notre reconnaissance envers nos proches, nos familles et nos amis, pour leur soutien inconditionnel, leurs encouragements constants et leur compréhension tout au long de ce parcours académique.

Enfin, nous tenons à remercier tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce mémoire. Leur apport, qu'il soit intellectuel, moral ou logistique, a joué un rôle important dans la concrétisation de ce travail.

Nous sommes profondément reconnaissantes envers toutes ces personnes qui ont joué un rôle essentiel dans la réussite de ce mémoire. Leur soutien indéfectible et leurs conseils précieux resteront gravés dans notre mémoire.

Merci du fond du cœur.

Hamrouche Sana

Tekari Sarah

Résumé

Plusieurs travaux et études se sont intéressés à la détection automatique d'opinions dans les textes en utilisant différentes techniques et méthodes. La majorité des travaux actuels dans la détection d'opinions utilisant l'apprentissage profond traitent des langues indo-européennes et en particulier la langue anglaise. Il serait donc intéressant de développer un modèle de détection d'opinions dans des textes en langue arabe.

De ce fait, nous avons proposé dans ce travail d'explorer quelques algorithmes d'apprentissage profond comme les réseaux de neurones convolutifs (CNNs) et les réseaux de neurones récurrents (RNNs) pour la détection d'opinions dans des textes en langue arabe. Nous avons examiné l'impact de l'utilisation de stemming sur les données d'entrer, et l'impact d'utiliser différentes techniques de représentation des mots tels que le Tf idf,Bag of words,Aravec et AraBERT sur la performance des modèles. On a également réalisé plusieurs expérimentations sur plusieurs autre dataset. Une analyse comparative a été effectuée sur les résultats obtenus. Finalement ces résultats indiquent que le modèle CNN nous donne une meilleure exactitude avec 95.32% sur le dataset HARD, De plus, nous avons constaté que la technique de representation des données AraBERT ne necissite pas une étape de stemming et qu'elle donne les meilleures performances comparant avec les autres techniques de representations sur tous les ensembles de données de test.

Mots clés :

Détection d'opinions, analyse des sentiments, langue arabe, apprentissage profond, représentation des documents textuel.

الملخص

ركزت العديد من الدراسات والأبحاث على الكشف التلقائي للآراء في النصوص باستخدام تقنيات وأساليب مختلفة. ويتناول غالبية الأعمال الحالية في الكشف عن الآراء باستخدام التعلم العميق بشكل رئيسي اللغات الاوروبية، وخاصة الإنجليزية. ومن ثم، سيكون من المثير للاهتمام تطوير نموذج للكشف عن الآراء في النصوص باللغة العربية.

لذلك، في هذه الدراسة، قمنا باقتراح عدة خوارزميات للتعلم العميق، مثل الشبكات العصبية التابعة للتصفية (CNNs)والشبكات العصبية التابعة للتكرار(RNNs) ، للكشف عن الآراء في النصوص العربية. كما فحصنا تأثير تمثيل النص على أداء هذه النماذج. كتأثير استخدام عمليات التجذير (Stemming) ، وتأثير استخدام تقنيات تمثيل الكلمات المختلفة مثل Tf-idf وBag of words وAraVec وAraBERT على أداء هذه النماذج. قمنا بتنفيذ تجارب متعددة على مجموعات بيانات مختلفة تم إجراء تحليل مقارن للنتائج المحصل عليها. في النهاية، أشارت هذه النتائج إلى أن نموذج CNN يعطينا دقة أفضل بنسبة 95.32% على مجموعة البيانات HARD. زيادة على ذلك، لاحظنا أن تقنية تمثيل البيانات AraBERT لا تتطلب خطوة للتجذير وتعطي أداءً متفوقًا مقارنة بالتقنيات الأخرى على جميع مجموعات بيانات الاختبار.

الكلمات الرئيسية :كشف الرأي، تحليل المشاعر، اللغة العربية، التعلم العميق، تمثيل البيانات النصية.

Abstract

Several studies and research have focused on the automatic detection of opinions in texts using different techniques and methods. The majority of current works in opinion detection using deep learning predominantly address Indo-European languages, particularly English. Thus, it would be interesting to develop an opinion detection model for texts in the Arabic language.

Hence, in this study, we proposed to explore various deep learning algorithms, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), for opinion detection in Arabic texts. Additionally, we examined the impact of text representation on the performance of these models. Such as the impact of using stemming on the input data and the impact of using different word representation techniques such as Tf-idf, Bag of words, AraVec, and AraBERT on the performance of these models. We also conducted several experiments on various other datasets. A comparative analysis was performed on the obtained results. Finally, these results indicate that the CNN model achieves a higher accuracy of 95.32% on the HARD dataset. Additionally, we observed that the AraBERT data representation technique does not require a stemming step and it provides the best performance compared to other representation techniques on all test datasets.

Keywords:

Opinion detection, sentiment analysis, Arabic language, deep learning, textual document representation.

Table des Matières

Introduction générale 1

Chapitre I État de l'art 3

I Introduction 3

II Analyse des sentiments 3

II. 1 Définition de l'analyse des sentiments 3

II. 2 Niveaux d'analyse des sentiments 4

II. 2. 1 Niveau du document 4

II. 2. 2 Niveau de la phrase 4

II. 2. 3 Niveau entité/aspect 5

II. 3 Types d'analyse des opinions 5

II. 4 Domaine d'application de l'analyse d'opinion 6

II. 5 Analyse des sentiments en langue arabe 6

II. 6 Défis de l'analyse des sentiments en langue arabe 7

II. 7 Approches d'analyse des sentiments 8

II. 7. 1 Approche basée sur les lexiques 9

II. 7. 2 Approche basée sur l'apprentissage automatique 9

II. 7. 3 Approche hybride 10

III Apprentissage profond 10

III. 1 Définition de l'apprentissage profond 10

III. 2 Définitions et concepts clés 10

III. 3 Avantages et limites de l'apprentissage profond 14

III. 4 Traitement des données d'entrée pour l'analyse des sentiments : 15

III. 4. 1 Collecte et préparation des données 15

III. 4. 2 Nettoyage et normalisation des données 15

III. 4. 3 Représentation des données 15

III. 5 Techniques d'apprentissage profond pour l'analyse des sentiments 19

III. 5. 1 Réseaux de neurones convolutifs (CNN) 19

III. 5. 2 Réseaux de neurones récurrents (RNN) 20

III. 5. 3 Réseaux de neurones à mémoire à court terme (LSTM) 21

III. 5. 4 Réseaux de neurones récurrents bidirectionnels à mémoire à court terme (BiLSTM) 22

III. 6 Évaluation des modèles d'analyse des sentiments 23

IV Travaux connexes 24
IV. 1 Revue de la littérature sur l'analyse des sentiments en langue arabe 24
IV. 2 Synthèse des travaux connexes 26
V Conclusion 27
Chapitre II Conception des modèles d'analyse des sentiments 28
I Introduction 28
II Description de l'architecture globale de notre système 28
II. 1 Chargement des données 29
II. 2 Prétraitement des données 29
II. 2. 1 Nettoyage et normalisation des données textuelles 29
II. 2. 2 Suppression des caractères inutiles 30
II. 2. 3 Suppression des stop words 31
II. 2. 4 Stemming 31
II. 2. 5 Tokenisation 32
II. 3 Représentation et pondération de données 32
II. 3. 1 Sac de mots (bag-of-words) 32
II. 3. 2 Tf-IDF 32
II. 3. 3 Word embedding 32
II. 4 Fractionnement des données 33
II. 5 Construction des modèles 34
II. 5. 1 Le modèle CNN 36
II. 5. 2 Le modèle LSTM 37
II. 5. 3 Le modèle BiLSTM 38
II. 5. 4 Le modèle CNN-BiLSTM 38
II. 5. 5 Choix des hyperparamètres : 39
III Conclusion 41
Chapitre III Tests et résultats 42
I Introduction 42
II Environnement de travail 42
II. 1 Matériel 42
II. 2 Logiciels et bibliothèques 42
III Description du dataset 43
IV Expérimentations et analyses des performances 45
IV. 1 Expérimentations des modèles proposés sur le dataset HARD : 45
IV. 1. 1 Impact du stemming 45

IV. 1. 2 Impact des techniques de représentation des mots 48
IV. 1. 3 Impact des modèles de Deep Learning 50
IV. 2 Expérimentation sur d'autres dataset 52
IV. 2. 1 Efficacité de notre modèle 52
IV. 2. 2 Validation des résultats expérimentaux 53
V Interprétation des résultats 55
VI Conclusion 57
Conclusion générale 58
Bibliographie 59

Liste des figures

Figure 1 Les trois niveaux d'analyse des sentiments. 4
Figure 2 Les types de phrases selon leur polarité et leur objectivité. 5
Figure 3 les approches d'analyse des sentiments....9
Figure 4 Les composants d'un neurone. 11
Figure 5 Les trois couches d'un réseau de neurone. 11
Figure 6 Anatomie d'un neurone formel. 12
Figure 7 Courbe de la fonction sigmoïde.[17] 13
Figure 8 Courbe de la fonction tanh. 13
Figure 9 Courbe de la fonction ReLu.[17] 13
Figure 10 Courbe de la fonction softmax....14
Figure 11 Une pyramide représente les trois étapes de traitement sur les données d'entrée. 15
Figure 12 Exemple comparatif de Skip-Gram et CBOW pour Word2Vec. 18
Figure 13 Les couches d'un réseau neurone convolutif. 20
Figure 14 Architecture globale d'un réseau de neurone récurrent. 20
Figure 15 Les différents types d'entrée d'un réseau neurone reçurent. 21
Figure 16 Architecture globale d'un réseau LSTM. 22
Figure 17 Architecture globale d'un réseau BiLSTM. 22
Figure 18 Architecture globale du système. 28
Figure 19 Architecture de CNN. 37
Figure 20 Architecture de LSTM....38
Figure 21 Architecture de BiLSTM. 38
Figure 22 Architecture de CNN-BiLSTM. 39
Figure 23 L'exactitude et la perte en utilisant la fonction d'optimisation Adam et SGD....40
Figure 24 Le pourcentage de critiques positives et négatives dans le dataset....44
Figure 25 Les cinq premières lignes du dataset original. 44
Figure 26 Les cinq premières lignes du dataset après le prétraitement et sans stemming. ..44
Figure 27 Les cinq premières lignes du dataset après le prétraitement et avec le stemmer ISRI. 45
Figure 28 Les cinq premières lignes du dataset après le prétraitement et avec le stemmer .45

Liste des tables

Tableau 1 Exemples d'expressions et de variations régionales en langue arabe pour l'analyse des sentiments. 7
Tableau 2 Exemples d'expressions culturelles en arabe et leurs défis pour l'analyse des sentiments. 8
Tableau 3 Les différentes significations des mots. 8
Tableau 4 Les avantages et les limites de l'apprentissage profond. 14
Tableau 5 Représentation de documents en sac de mots. 16
Tableau 6 Exemple de TF-IDF. 17
Tableau 7 Matrice de confusion. 23
Tableau 8 Résumé sur les travaux connexe à l'analyse des sentiments. 26
Tableau 9 La normalisation des caractères en langue arabe. 29
Tableau 10 les émojis avec leurs significations 30
Tableau 11 Comparaison entre des textes originaux et nettoyés. 30
Tableau 12 Les caractéristiques des modèles Twitter CBOW et Skip-gram. 33
Tableau 13 Tableau récapitulatif des hyperparamètres des modèles. 39
Tableau 14 Comparaison de la performance des modèles avec une et deux couches. 40
Tableau 15 Les caractéristiques des matérielles utilisées 42
Tableau 16 Évaluation de la performance des modèles de deep learning sans stemming. 46
Tableau 17 Évaluation de la performance des modèles de deep learning en utilisant ISRI stemmer. 46
Tableau 18 Évaluation de la performance des modèles de deep learning en utilisant Tashaphyne stemmer. 47
Tableau 19 Performances des modèles de Deep Learning en utilisant bag of words avec et sans stemming. 48
Tableau 20 Performances des modèles de Deep Learning en utilisant TF-idf avec et sans stemming. 48
Tableau 21 Performances des modèles de Deep Learning en utilisant AraVec(cbow) avec et sans stemming. 49
Tableau 22 Performances des modèles de Deep Learning en utilisant AraVec (skipgram) avec et sans stemming. 49
Tableau 23 Performances des modèles de Deep Learning en utilisant AraBert avec et sans stemming. 49
Tableau 24 Performances de modèle CNN en utilisant toutes les techniques de représentation des données. 50
Tableau 25 Performances de modèle LSTM en utilisant toutes les techniques de représentation des données. 50

Tableau 26 Performances de modèle BiLSTM en utilisant toutes les techniques de représentation des données. 50
Tableau 27 Performances de modèle CNN-BiLSTM en utilisant toutes les techniques de représentation des données. 51
Tableau 28 Le taux d'erreur dans les quatres modèles. 52
Tableau 29 Comparaison entre la performance de modèle BiLSTM[21] et notre modèle. ...52
Tableau 30 Un résumé sur les ensembles de données utilisées pour le test. 53
Tableau 31 Les résultats de test de différents modèles de réseaux de neurones sur les dataset Arabic Company Reviews, ASTD et Arabic 100k Reviews. 54

Introduction générale

La détection d'opinions est un domaine de recherche en plein essor qui vise à comprendre et à interpréter les opinions, les émotions et les attitudes exprimées dans le langage humain. Cependant, avec le développement des médias sociaux et des plateformes en ligne, on remarque une croissance exponentielle de la quantité de données textuelles produites par les utilisateurs, qui ont désormais la possibilité de partager leurs opinions et leurs émotions à grande échelle dans différentes langues, notamment l'anglais qui représente la première langue utilisée sur le web avec 58,8% selon[1]. Cette explosion de données textuelles a ouvert de nouvelles opportunités pour analyser et exploiter les opinons exprimées par les utilisateurs, se concentrant sur la classification des commentaires et leur polarité (positif, négatif, neutre).

Cependant, la détection d'opinion présente des défis particuliers, notamment en ce qui concerne les langues autres que l'anglais. Chaque langue a ses spécificités, ses nuances et ses expressions idiomatiques, ce qui rend l'analyse d'opinion assez complexe surtout pour le cas particulier de la langue arabe. En effet, en plus de sa richesse sémantique, de sa structure grammaticale complexe et de ses multiples niveaux d'expressions, la langue arabe se caractérise aussi par une diversité de dialectes et de variations régionales. Les sentiments et les émotions sont souvent exprimés de manière sensible et raffinée, nécessitant une compréhension fine des contextes culturels, sociaux et linguistiques.

Dans ce contexte, différentes approches ont été développées pour analyser et extraire automatiquement les opinions à partir de données textuelles. Parmi ces approches, on trouve celles basées sur le lexique, qui reposent sur l'utilisation d'un dictionnaire de mots clés associés à des polarités spécifiques. Cette approche est rapide et simple à mettre en œuvre, car elle se fonde sur une correspondance directe entre les mots présents dans le texte et les catégories d'opinion prédéfinies. Cependant, elle peut être limitée en termes de couverture et de capacité à détecter des détails pour comprendre le contexte d'une phrase.

D'autres approches utilisent des algorithmes d'apprentissage pour analyser les données et extraire des informations. L'apprentissage automatique peut atteindre des niveaux élevés de précision dans des tâches telles que la détection d'opinions, car il utilise un ensemble de données d'entrainement pour identifier des caractéristiques et des relations dans les données textuelles.

Parmi les techniques basées sur l'apprentissage automatique, l'apprentissage profond (ou deep learning) émerge en tant qu'approche de plus en plus utilisée. L'apprentissage profond utilise des réseaux de neurones profonds pour capturer les relations complexes entre les caractéristiques des données et de réaliser des tâches d'analyse de manière automatique.

Ceci permet de développer des modèles permettant de classifier des opinions données en opinions positifs ou négatifs.

Néanmoins, pour détecter correctement l'opinion des textes il faut avoir certes un bon modèle de détection d'opinions mais il faut avoir aussi une bonne représentation des textes eux-mêmes. En effet, des textes mal représentés peuvent diminuer dans la précision de la détection d'opinions.

De ce fait, nous nous proposons dans ce travail d'explorer quelques algorithmes d'apprentissage profond pour la détection d'opinions dans des textes en langue arabe et de voir l'influence de la représentation des textes sur cette détection.

Notre mémoire est organisée de la manière suivante :

- Le premier chapitre contient un état de l'art dans le domaine de l'analyse des sentiments. Il introduit les notions fondamentales de cette discipline. Les travaux connexes dans le domaine de l'analyse des sentiments en langue arabe sont également revus. Il explore aussi le domaine de Deep Learning.
- Le deuxième chapitre présente une méthodologie détaillée pour la conception et l'évaluation de modèles d'analyse des sentiments en langue arabe. Nous aborderons les différentes étapes de prétraitement des données, de représentation des données textuelles, de construction des modèles basés sur le Deep Learning, et d'évaluation des performances.
- Le troisième chapitre est dédié aux tests et aux résultats. Il décrit le dataset utilisé pour les tests et met en évidence l'impact du stemming, des techniques de représentation des mots et des modèles de Deep Learning sur les résultats. Une discussion sur résultats est par ailleurs effectuée.

Nous terminons notre mémoire avec une conclusion générale où nous résumons les résultats obtenus dans le troisième chapitre et on donne aussi des perspectives pour des travaux futurs.

Chapitre I État de l’art

I Introduction

La détection d'opinion ou l'analyse des sentiments est une technique qui a émergé avec l'essor des plateformes des médias sociaux, où les utilisateurs peuvent exprimer leurs émotions à grande échelle. Son objectif est de classer un texte dans des catégories positives, négatives ou neutres, afin d'en déterminer la tonalité émotionnelle. L'analyse des sentiments est réalisée à l'aide de techniques d'apprentissage automatique, l'apprentissage profond étant la méthode la plus populaire en raison de sa capacité à traiter efficacement de grands volumes de données. Les modèles d'apprentissage profond peuvent comprendre des tâches complexes en abstrayant les représentations de données.

Dans ce chapitre, nous explorerons en détail les notions d’analyse des sentiments et de l’apprentissage profond. Nous aborderons également le processus à suivre pour effectuer une analyse de sentiments en utilisant les techniques de Deep Learning. Nous examinerons aussi plusieurs travaux connexes à ce domaine, afin de mieux comprendre l'évolution de cette technologie et les perspectives futures qu'elle offre.

II Analyse des sentiments

II. 1 Définition de l’analyse des sentiments

Définition 1 : L'analyse des sentiments, aussi connue sous le nom de classification des opinions, est la tâche de déterminer la polarité d'un document, c'est-à-dire s'il est positif, négatif ou neutre. Cette tâche est souvent accomplie en extrayant des caractéristiques du texte, telles que des mots-clés ou des expressions, puis en utilisant un algorithme de classification pour attribuer une polarité au document[2] .

Définition 2 : L'analyse des sentiments est une tâche du traitement du langage naturel qui vise à extraire et à identifier des informations subjectives à partir de données textuelles, telles que des opinions, des sentiments, des attitudes ou des émotions, et à les classifier en fonction de leur polarité, qui peut être positive, négative ou neutre [3].

D’après les deux définitions, on peut conclure que l’analyse des opinions ou analyse des sentiments est une technique de traitement automatique du langage naturel qui vise à comprendre et à préciser les sentiments, les opinions et les attitudes exprimées dans un texte subjectif donné soit un commentaire, un article de presse ou un message sur les réseaux sociaux.

II. 2 Niveaux d'analyse des sentiments

Il existe trois niveaux de granularité dans l'analyse des sentiments, le niveau du document (Message level ou Document level), le niveau de la phrase (Sentence level) et le niveau des aspects (Entity and aspect level) [2]

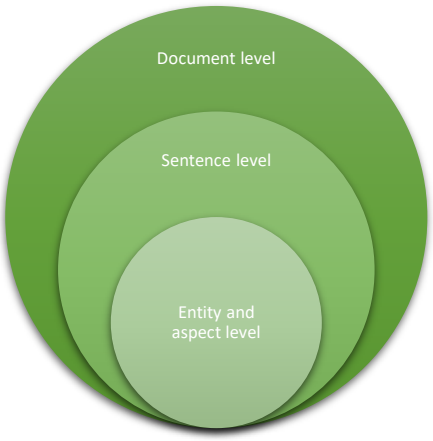

Figure 1 Les trois niveaux d'analyse des sentiments.

II. 2. 1 Niveau du document

Lors de l'analyse des opinions au niveau du document, il est nécessaire de déterminer si le sentiment général du document est positif, négatif ou neutre. C'est ce que l'on appelle la polarité[4]. À ce niveau d'analyse, les sentiments exprimés dans chaque document sont centrés sur un sujet particulier. Le document peut-être, par exemple, un article, une revue, un blog, etc. Cette forme de classification est la plus basique et simple[5].

II. 2. 2 Niveau de la phrase

L'analyse d'opinion au niveau de la phrase permet de détecter les opinions exprimées dans les phrases individuelles d'un document écrit. On distingue généralement deux types de phrases [6] :

- Phrase subjective : une phrase qui exprime une opinion, un sentiment ou une émotion personnelle de l'auteur. Elle peut varier d'une personne à l'autre et ne peut pas être prouvée ou réfutée de manière objective. Par exemple, "Le chat est beau" est une phrase subjective qui implique une appréciation esthétique personnelle et qui peut varier selon les opinions et les préférences de chacun.
- Phrase objective : une phrase objective est une phrase qui décrit des faits ou des informations de manière impartiale et sans émotion personnelle. Elle est vérifiable et peut être prouvée ou réfutée de manière objective. Par exemple, "Le chat est noir" est une phrase objective qui décrit simplement la couleur du chat sans impliquer de point de vue personnel.

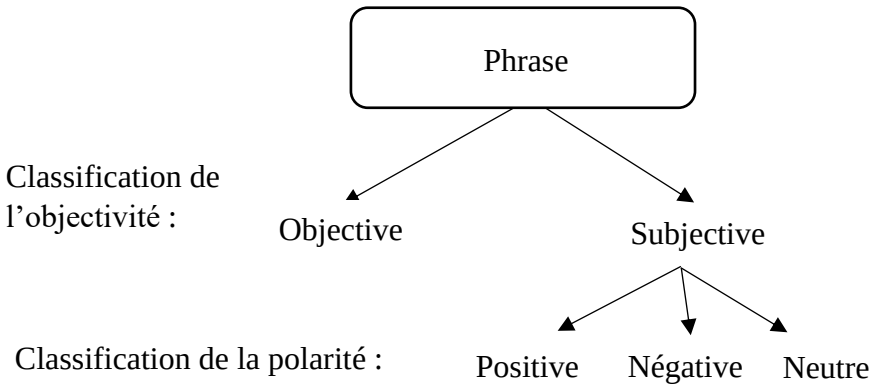

Figure 2 Les types de phrases selon leur polarité et leur objectivité.

Donc le principal avantage de l'analyse au niveau des phrases réside dans sa capacité à classer les phrases en termes de subjectivité ou d'objectivité[5].

II. 2. 3 Niveau entité/aspect

L'analyse au niveau de l'aspect est une approche plus fine que les analyses au niveau des documents et des phrases. L'objectif est de déterminer la polarite exprimée à propos d'un aspect spécifique d'une entité dans un document[7]. Par exemple, si l'on se concentre sur l'aspect "résolution de l'écran" dans la phrase "la résolution de l'écran est très bonne, mais la taille est très petite", l'opinion exprimée est favorable. À l'inverse, si l'on se concentre sur l'aspect "taille", l'opinion est défavorable.

II. 3 Types d'analyse des opinions

L'analyse des opinions est une technique qui permet de déterminer les opinions et les sentiments exprimés dans un texte donné. Il existe plusieurs types d'analyse des opinions, chacun se concentrant sur un aspect différent de l'opinion exprimée dans le texte. Voici une brève description de quelques-uns des types les plus courants :

- Classification de la polarité : le processus de classification par polarité consiste à classer les opinions présentées dans un texte comme positives, négatives ou neutres. Cette technique est utile pour déterminer le sentiment général véhiculé dans un grand nombre de textes.
- Analyse de l'émotion : cette stratégie est centrée sur la reconnaissance des émotions véhiculées dans les documents écrits, tels que la colère, la tristesse, la joie et l'horreur[8][9]. Elle est efficace pour comprendre les émotions présentes dans un point de vue particulier.
- Analyse des intentions : cette approche est centrée sur l'identification de l'objectif qui est derrière un point de vue, qu'il soit positif ou négatif. Elle s'avère bénéfique pour comprendre les raisons sous-jacentes des opinions exprimées.
- Analyse des aspects : cette méthode consiste à identifier les éléments particuliers d'un produit, d'un service ou d'un scénario qui sont mentionnés dans un document écrit, ainsi que les points de vue liés à ces facteurs.

II. 4 Domaine d'application de l'analyse d'opinion

La détection d'opinions peut s'appliquer à plusieurs domaines, notamment :

a) Marketing et publicité :

L'analyse d'opinion est une technique utilisée pour mieux comprendre les attitudes et les opinions des consommateurs à l'égard de divers produits et services. C'est un outil efficace pour développer des campagnes publicitaires qui trouvent un écho auprès du public ciblé.

b) Service clientèle :

Les entreprises utilisent l'analyse d'opinion pour suivre les réactions des clients et répondre rapidement à toute plainte ou problème.

c) Politique :

L'analyse d'opinions est utilisée pour surveiller l'opinion publique et mesurer la popularité des partis politiques et des candidats.

d) Éducation :

Les établissements d'enseignement utilisent l'analyse d'opinion pour évaluer la satisfaction des étudiants et améliorer la qualité de l'enseignement.

e) Médias sociaux :

L'analyse d'opinions est largement utilisée sur les plateformes de médias sociaux pour suivre les tendances et les opinions des utilisateurs.

f) Santé :

L'analyse d'opinions est utilisée pour surveiller les opinions et les sentiments des patients et pour améliorer les soins de santé.

g) Tourisme :

Les entreprises du tourisme utilisent l'analyse d'opinions pour surveiller les commentaires des voyageurs et pour améliorer l'expérience client.

h) Finance :

L'analyse d'opinions est utilisée pour surveiller l'opinion des investisseurs et pour prédire les tendances du marché.

II. 5 Analyse des sentiments en langue arabe

L'analyse des sentiments en arabe respecte les principes fondamentaux appliqués dans d'autres langues. Cependant, il existe certains aspects uniques liés à la langue arabe. Les ressources lexicales arabes telles que les dictionnaires de synonymes et les corpus annotés servent de base aux techniques utilisées dans l'analyse des sentiments en arabe.

L'analyse des sentiments en arabe est une tâche complexe en raison des caractéristiques uniques de la langue, telles que les racines et les affixes, ainsi que de son vocabulaire étendu. L'existence de différents dialectes dans la langue arabe complique encore la tâche de création de modèles capables de couvrir tous les dialectes. Néanmoins, avec les progrès des techniques d'apprentissage automatique et de traitement du langage naturel, plusieurs méthodes sont apparues pour l'analyse des sentiments en arabe. Ces méthodes vont des approches traditionnelles basées sur le lexique aux méthodes avancées basées sur l'apprentissage automatique telles que les réseaux neuronaux et les arbres de décision.

II. 6 Défis de l'analyse des sentiments en langue arabe

L'analyse des sentiments en général et en langue arabe en particulier est un domaine complexe et exigeant, qui implique de nombreux défis pour les chercheurs et les praticiens, à l'opposé de la simple classification de texte, en raison des nombreux défis du domaine :

- La langue arabe est parlée dans de nombreux pays et régions, et il existe des différences significatives entre les dialectes. Cela rend l'analyse des sentiments en arabe encore plus difficile, car les expressions et les mots peuvent varier considérablement selon les régions.

Pays/Région	Dialecte	Exemple de phrases avec variations
Algérie	Algérien	"Saha" (salut) ou " el-muškila " (problème)
Maroc	Darija	"Salam" (salut) ou "Moochkil" (problème)
Tunisie	Tunisien	" Aslema " (salut) ou " Mochkla " (problème)
Égypte	Égyptien	"Salam" (salut) ou " mushkila " (problème)
Golfe	Persique Arabe du Golfe	"Salam" (salut) ou " waqi'a " (problème)
Levant	Arabe du Levant	"Marhaba" (salut) ou "Mish mabsout" (problème)

Tableau 1 Exemples d'expressions et de variations régionales en langue arabe pour l'analyse des sentiments.

- Le traitement de la négation est l'un des défis majeurs de l'analyse des sentiments en langue arabe, tout comme pour d'autres langues. La négation peut inverser le sens d'un mot ou d'une phrase, ce qui peut affecter la polarité globale de l'avis ou de la critique.
  Par exemple, en arabe, la phrase "الطعام لم يكن جيدا" (al-ta'am lam yakun jayyidan) signifie "la nourriture n'était pas bonne". La négation "لم" (lam) est utilisée pour exprimer le "non" ou le "pas". Ainsi, sans tenir compte de la négation, on pourrait considérer cette phrase comme ayant une polarité positive. Cependant, en tenant compte de la négation, la phrase doit être considérée comme ayant une polarité négative.
- La langue arabe est fortement associée à la culture arabe, qui possède des normes et des idéaux spécifiques pour transmettre des émotions et des points de vue. Par exemple, les expressions de politesse, de respect et de modestie peuvent être courantes dans les textes arabes, ce qui peut rendre difficile la détection des véritables sentiments exprimés.

Exemples d'expression en arabe	Type de défi pour l'analyse des sentiments
"أرجوكم أن تفعلوا ذلك"	Politesse et modestie
"حضرتكم تعلمون أفضل مني"	Respect et modestie
"لا أريد الإساءة لأحد"	Préférence pour éviter les critiques directes
"الحمد لله على كل حال"	Attitude positive et optimisme

Tableau 2 Exemples d'expressions culturelles en arabe et leurs défis pour l'analyse des sentiments.

- Il y a relativement un nombre insuffisant d'ensemble de données disponibles pour l'analyse des sentiments en arabe[10], ce qui limite les possibilités d'entraîner et de tester des modèles d'apprentissage automatique pour l'analyse des sentiments.
- Les diacritiques : sont des signes orthographiques utilisés en arabe pour indiquer la prononciation correcte des mots. Cependant, l'absence de diacritique peut affecter la reconnaissance des mots lors de l'analyse ce qui rend nécessaire l'utilisation de règles morphologiques complexes pour identifier les unités lexicales et analyser le texte de manière précise[11].
- Signification des mots : il est possible d'identifier différentes significations associées à un mot, lorsqu'on effectue une analyse de sentiment, il est important de prendre en compte le contexte pour pouvoir déterminer la signification correcte du mot et classer correctement le texte selon la polarité[10]. Cela peut être un défi pour les modèles d'analyse de sentiment, car cela nécessite une compréhension approfondie de la langue arabe.

Le mot	La signification
تشغيل	Activer / Faire fonctionner / Diriger / Mettre en marche
مجرد	Simplement / Seulement / Même pas / Purement
مثل	Comme / Tel que / Pareil à / Par exemple
جريمة	Crime / Délit / Infraction / Violation
حرية	Liberté / Autonomie / Indépendance / Démocratie

Tableau 3 Les différentes significations des mots.

C'est pour cela les chercheurs doivent prendre en compte ces défis pour développer des approches efficaces pour l'analyse des sentiments en langue arabe. Différentes approches ont été proposées pour l'analyse des sentiments, comme les approches basées sur les lexiques, approches d'apprentissage automatique et approches hybrides.

II. 7 Approches d'analyse des sentiments

Ce sont des méthodes utilisées pour détecter les sentiments exprimés dans les textes, elles peuvent être basées sur des lexiques ou des techniques d'apprentissage automatique pour analyser les émotions (Machine Learning) ou par une combinaison des deux approches (approche hybride)[12] .

La figure suivante montre ces approches :

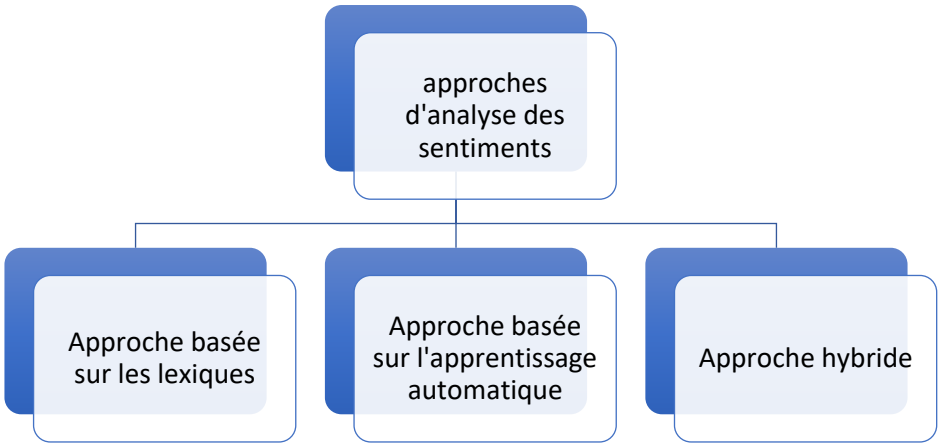

Figure 3 les approches d'analyse des sentiments

II. 7. 1 Approche basée sur les lexiques

L'approche basée sur les lexiques : cette approche consiste à utiliser des lexiques de sentiments, c'est-à-dire des dictionnaires de mots associés à des polarités positives ou négatives. Les textes sont ensuite analysés à l'aide de ces lexiques, qui permettent d'attribuer une polarité globale au texte en fonction des polarités des mots qu'il contient. Cette approche est relativement simple à mettre en œuvre, mais elle est limitée par la qualité des lexiques utilisés et la difficulté de comprendre la relation entre les mots de phrase.

II. 7. 2 Approche basée sur l'apprentissage automatique

Les approches d'apprentissage automatique utilisent des algorithmes d'apprentissage pour prédire la polarité des textes. Cela implique généralement de collecter un grand volume de données d'entraînement constituées de textes préétiquetés avec leur polarité (positif, négatif ou neutre). Le modèle apprend ensuite à partir de ces données pour prédire la polarité des nouveaux textes qui n'ont pas encore été étiquetés. Les algorithmes d'apprentissage automatique couramment utilisés pour l'analyse des sentiments comprennent les arbres de décision, les réseaux de neurones, les SVM (machines à vecteurs de support) et les méthodes basées sur l'apprentissage en profondeur, comme les réseaux de neurones profonds. Il est possible d'adopter une approche automatique en analyse des sentiments en utilisant soit des techniques d'apprentissage supervisé, soit des techniques d'apprentissage non supervisé ou semi-supervisé :

- **Apprentissage supervisé :** est un type d'apprentissage où un modèle est entraîné à partir d'un ensemble de données étiquetées. Les données étiquetées sont celles ou chaque échantillon de données à une étiquette ou une catégorie connue. Par exemple, si vous entraînez un modèle pour reconnaître la polarite des sentiments, vous aurez besoin d'un ensemble de données étiquetées où chaque entrée est étiquetée comme positive ou négative. Dans l'apprentissage supervisé, le modèle est entraîné à trouver une fonction mathématique qui peut prendre les données d'entrée et produire la bonne sortie, c'est-à-dire la bonne étiquette.

- **Apprentissage non supervisé :** d'autre part, l'apprentissage non supervisé est un type d'apprentissage où un modèle est entraîné à partir d'un ensemble de données non étiquetées. Dans ce cas, le modèle doit trouver les relations entre les données d'entrée sans aucune information préalable sur les catégories ou les étiquettes. L'apprentissage non supervisé utilise des techniques telles que les réseaux de neurones auto-encodeurs, le clustering et la réduction de dimensionnalité.
- **Apprentissage semi-supervisé :** il existe également une troisième catégorie d'apprentissage, appelée l'apprentissage semi-supervisé, qui est un mélange des deux approches ci-dessus. Dans l'apprentissage semi-supervisé, une partie des données est étiquetée et une partie est non étiquetée.

II. 7. 3 Approche hybride

Cette approche combine les deux approches précédentes, c'est-à-dire qu'elle utilise à la fois des lexiques de sentiments et des algorithmes d'apprentissage automatique pour analyser les textes et identifier les sentiments qui y sont exprimés. Cette approche permet de bénéficier des avantages des deux approches, mais elle nécessite ainsi une quantité importante de données, d'entraînement et de connaissances en matière de traitement automatique du langage naturel.

III Apprentissage profond

III. 1 Définition de l'apprentissage profond

L'apprentissage profond (Deep Learning) est une branche de l'apprentissage automatique qui relève de l'intelligence artificielle (IA)[13]. Qui utilise des réseaux de neurones artificiels pour apprendre à partir de données passées en entrée, afin d'effectuer des tâches telles que la reconnaissance d'images, la reconnaissance vocale, et la classification. Le terme "profond" fait référence à la profondeur de ces réseaux de neurones, qui contiennent plusieurs couches cachées et sont capables d'apprendre des représentations complexes des données en utilisant des algorithmes d'optimisation tels que la descente de gradient. Le Deep Learning est devenu particulièrement populaire ces dernières années en raison des avancées dans le matériel informatique, de l'abondance de données disponibles et de l'amélioration des algorithmes.

III. 2 Définitions et concepts clés

a) Un neurone :

Cellule de base du tissu nerveux, capable de recevoir, d'analyser et de produire des informations. (La partie principale, ou corps cellulaire du neurone, est munie de prolongements, les dendrites et l'axone.)[14]

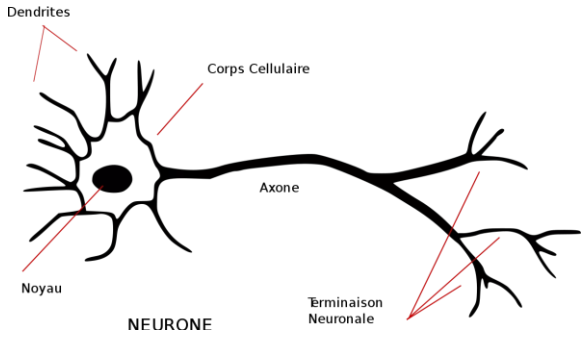

Figure 4 Les composants d'un neurone.

b) Réseaux de neurones :

Un réseau de neurones est un modèle de traitement de l'information inspiré du fonctionnement du cerveau humain. Il est composé de plusieurs neurones interconnectés, organisés en couches, qui sont capables de traiter des informations et de reconnaître des modèles dans des données d'entrée complexes. Les réseaux de neurones sont utilisés dans de nombreux domaines tels que la reconnaissance d'image, la reconnaissance vocale, la prédiction de séries temporelles et la classification de données.

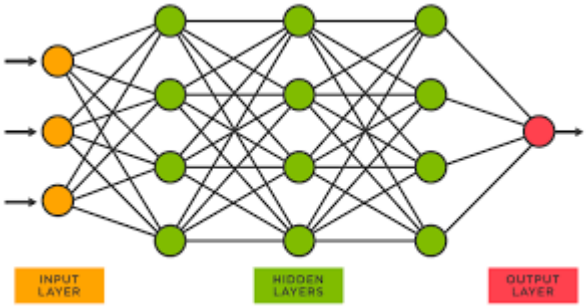

Figure 5 Les trois couches d'un réseau de neurone.

c) Couches :

Un réseau neuronal se compose de neurones interconnectés qui opèrent sur des données d'entrée pour générer une sortie. Ces neurones sont regroupés en couches, et chaque couche peut avoir un nombre différent de neurones. Les neurones de chaque couche sont connectés à tous les neurones de la couche précédente afin d'assurer un fonctionnement compatible.

Un réseau neuronal comporte différents types de couches qui remplissent des fonctions distinctes dans le traitement des informations. La première couche, la couche d'entrée, est chargée de recevoir les données d'entrée et de les transmettre à la première couche cachée. Les couches cachées sont chargées d'extraire les caractéristiques des données d'entrée et de les transformer à l'aide de fonctions non linéaires. Enfin, les couches de sortie sont chargées de fournir la réponse finale du réseau neuronal, qui peut prendre la forme d'une prédiction, d'une classification ou d'un autre type de sortie.

Les réseaux neuronaux peuvent comprendre plusieurs couches cachées, qui varient en fonction de la difficulté de la tâche à accomplir. Ces couches cachées aident le réseau neuronal

à comprendre l'information à plusieurs niveaux, en fusionnant les caractéristiques apprises par chaque couche pour créer des représentations plus complexes et plus avancées.

d) Poids et biais :

En apprentissage automatique, un modèle est généralement représenté par un graphe de calcul. Chaque nœud du graphe de calcul représente une opération mathématique, et les arêtes représentent le flux de données entre ces opérations.

Dans les réseaux de neurones, les nœuds représentent les neurones et les opérations mathématiques qu'ils effectuent. Les poids et les biais sont les paramètres de ces neurones, qui sont ajustés pendant l'entraînement du modèle pour minimiser la fonction de perte.

Les poids représentent l'influence que chaque entrée a sur la sortie du neurone. Ils sont initialement définis de manière aléatoire, puis ajustés durant l'apprentissage à l'aide d'un algorithme d'optimisation tel que la descente de gradient ou ADAM.Chaque neurone est doté d'un biais qui influence sa sortie. Ce biais est modifié au cours du processus d'apprentissage.

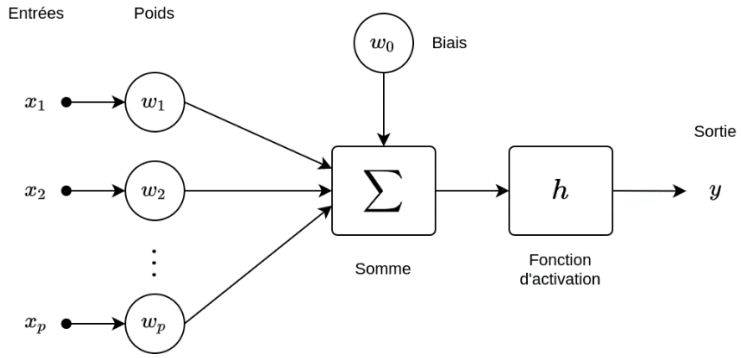

Figure 6 Anatomie d'un neurone formel.

e) Fonctions d'activation :

Les fonctions d'activation sont utilisées pour introduire une non-linéarité dans les sorties des neurones dans un réseau de neurones. Tel que toutes les entrées du neurone sont multipliées par des poids spécifiques et sommées pour obtenir une valeur appelée "activation pondérée". Cette activation pondérée représente la contribution totale des entrées pondérées au neurone. Une fois que l'activation brute est calculée, elle est ensuite transformée en une valeur de sortie par l'application de la fonction d'activation. La valeur de sortie obtenue après l'application de la fonction d'activation est transmise aux neurones de la couche suivante du réseau en tant qu'entrées pour leurs propres calculs. Il existe plusieurs fonction d'activation Voici quelques-unes les plus couramment utilisées en apprentissage profond[15] [16]:

1. Fonction sigmoïde : La fonction sigmoïde est une fonction en forme de S qui prend une valeur réelle en entrée et renvoie une valeur comprise entre 0 et 1 en sortie. Elle est souvent utilisée dans les réseaux de neurones pour la classification binaire.
   Elle est définie mathématiquement : $f(x) = 1/(1 + \exp(-x))$.

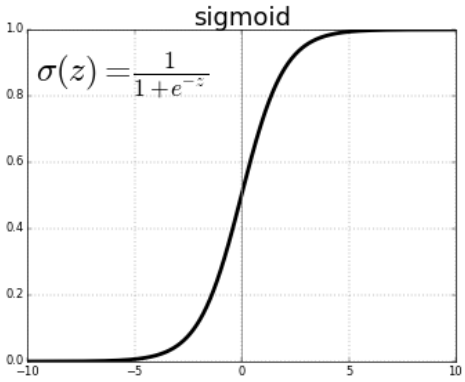

Figure 7 Courbe de la fonction sigmoïde.[17]

2. Fonction tanh : la fonction tanh est une fonction en forme de S qui prend une valeur réelle en entrée et renvoie une valeur comprise entre -1 et 1 en sortie. Elle est fréquemment utilisée dans les réseaux de neurones pour la classification multiclasse.

   Elle est définie mathématiquement : f(x) = (exp(x) - exp(-x))/(exp(x) + exp(-x)).

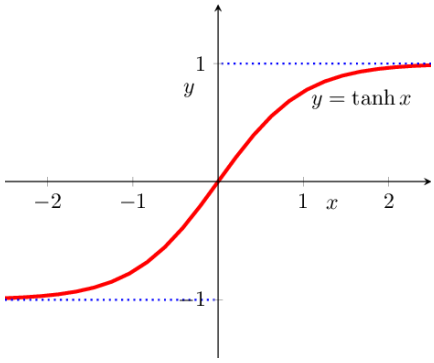

Figure 8 Courbe de la fonction tanh.

3. Fonction ReLU : la fonction ReLU (Rectified Linear Unit) est une fonction qui renvoie 0 si la valeur en entrée est négative et la valeur d'entrée elle-même si elle est positive. Elle est actuellement la fonction d'activation la plus couramment utilisée en raison de son efficacité et de sa simplicité.

   Elle est définie mathématiquement : f(x) = max(0,x) .

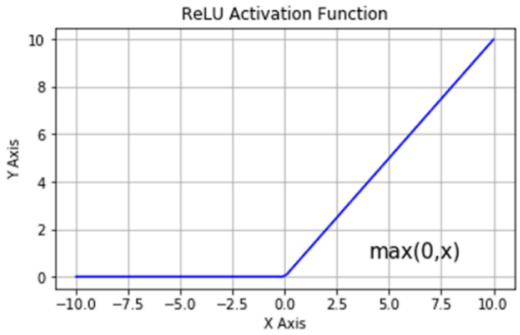

Figure 9 Courbe de la fonction ReLu.[17]

4. Fonction softmax : la fonction softmax est souvent utilisée dans les réseaux de neurones pour la classification multiclasse. Elle transforme les sorties de neurones en une distribution de probabilités sur les différentes classes.
   Elle est définie mathématiquement : softmax (x_i) = exp(x_i) / sum(exp(x_j)).

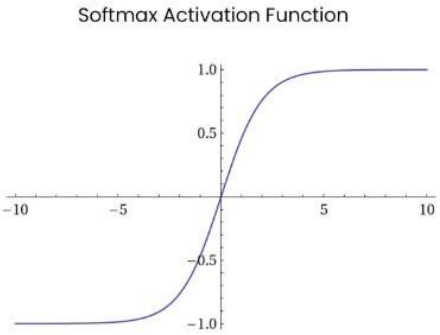

Figure 10 Courbe de la fonction softmax.

III. 3 Avantages et limites de l'apprentissage profond

L'apprentissage profond est une méthode de machine learning de plus en plus répandue dans divers domaines, y compris l'analyse des sentiments. Cette technique permet de traiter des quantités massives de données et de les utiliser pour prendre des décisions ou pour prédire des résultats futurs. Il offre plusieurs avantages, mais présente également des limites qu'il est important de prendre en compte pour une utilisation efficace. Dans le tableau ci-dessous, nous examinerons certains des avantages et des limites de l'apprentissage profond afin de mieux comprendre son impact dans le domaine de l'analyse des sentiments.

Avantages de l'apprentissage profond	**Limites de l'apprentissage profond**
Capable de traiter de grandes quantités de données complexes.	Besoin de grandes quantités de données pour une précision élevée.
Peut-être utilisé pour des tâches variées telles que la reconnaissance de la parole, la vision par ordinateur, la traduction automatique, etc.	Requiert une puissance de calcul importante, rendant son utilisation coûteuse.
Permet une grande précision pour les tâches pour lesquelles il est bien entraîné.	Probablement difficile à interpréter, ce qui limite sa transparence et sa compréhension.
Capable d'apprendre des représentations de données non-linéaires, ce qui le rend plus performant que les méthodes linéaires.	Peut-être sensible aux données d'entraînement et ne pas généraliser bien aux nouvelles données.
Peut-être utilisé pour l'analyse de texte, en permettant d'extraire des informations et des relations entre les mots.	Peut-être sujet à des biais liés aux données d'entraînement et à la qualité des données.

Tableau 4 Les avantages et les limites de l'apprentissage profond.

III. 4 Traitement des données d'entrée pour l'analyse des sentiments :

Le traitement des données est une étape cruciale dans l'analyse de sentiments en utilisant des techniques d'apprentissage profond. Cette étape consiste en la collecte et en la préparation des données nécessaires pour l'analyse de sentiments.

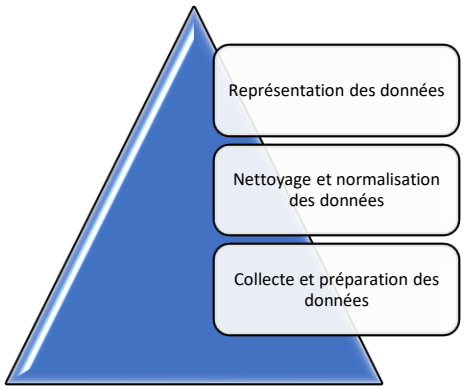

Figure 11 Une pyramide représente les trois étapes de traitement sur les données d'entrée.

III. 4. 1 Collecte et préparation des données

La première étape consiste à collecter les données nécessaires pour la détection des opinions ou l'analyse des sentiments. Les données peuvent être collectées à partir de sources telles que les réseaux sociaux, les blogs, les forums, les sites d'avis, etc. Il est important de sélectionner des données pertinentes pour le domaine d'application et d'assurer leur qualité.

III. 4. 2 Nettoyage et normalisation des données

Au cours de cette étape, les données sont nettoyées et normalisées afin d'éliminer les éléments indésirables tels que les bruits, les erreurs, les caractères spéciaux, les chiffres et autres informations inutiles. Le processus de nettoyage sera expliqué en détail dans le chapitre suivant.

III. 4. 3 Représentation des données

Les données doivent être transformées en une représentation numérique pour être utilisées dans un modèle de classification. Les techniques les plus courantes sont le sac de mots (bag-of-words), la représentation en vecteur TF-IDF, et les embéguines de mots (word embeddings).

a) Sac de mots (Bag-of-words) :

Est une technique de représentation de données textuelles qui consiste à convertir un document en une collection de mots, en ignorant leur ordre et leur structure syntaxique. Le texte est considéré comme un "sac" de mots, d'où le nom "sac de mots".

Pour créer un sac de mots, le texte est d'abord divisé en tokens, qui sont les unités de base du texte (par exemple, les mots). Ensuite, un vocabulaire est créé en listant tous les mots uniques présents dans les documents. Pour chaque document, un vecteur est créé en comptant

le nombre d'occurrences de chaque mot dans le vocabulaire. Ainsi, chaque document est représenté par un vecteur de dimensions égales au nombre de mots uniques dans le vocabulaire.

Exemple : Document 1 : "Le chat est mignon".

Document 2 : "Le chien est fidèle".

Document 3 : "Le chien et le chat sont amis".

Document	le	chat	est	mignon	chien	fidèle	et	Sont	amis
Document 1	1	1	1	1	0	0	0	0	0
Document 2	1	0	1	0	1	1	0	0	0
Document 3	2	1	0	0	1	0	1	1	1

Tableau 5 Représentation de documents en sac de mots.

Le sac de mots est une méthode simple et efficace pour représenter des données textuelles pour des tâches telles que la classification de texte, la catégorisation de documents, etc. Cependant, il présente des limitations, telles que l'ignorance de l'ordre des mots et la perte d'informations sémantiques et contextuelles importantes. C'est pourquoi des méthodes plus avancées, telles que les embeddings de mots, ont été développées pour améliorer la représentation de données textuelles.

b) TF-IDF (term frequency-inverse document frequency) :

Est une technique de pondération utilisée dans le traitement automatique du langage naturel pour évaluer l'importance d'un mot dans un document ou un corpus de documents. Le poids de chaque terme est calculé en multipliant sa fréquence dans le document (TF) par l'inverse de sa fréquence dans le corpus (IDF). Ainsi, les termes qui apparaissent fréquemment dans un document, mais rarement dans l'ensemble du corpus auront un poids plus élevé, car ils sont considérés comme plus importants pour la compréhension du contenu du document. La formule de calcul de TF-IDF est la suivante : TF-IDF(t, d, D) = TF(t, d) * IDF(t, D).

Où : t : le terme dont on calcule le poids

d : le document dans lequel le terme apparaît

D : l'ensemble des documents

TF(t, d) : la fréquence du terme t dans le document d

IDF(t, D) : l'inverse de la fréquence du terme t dans l'ensemble des documents D (calculée comme IDF(t, D) = log(N / (1 + n_t)), où N est le nombre total de documents dans le corpus et n_t est le nombre de documents dans le corpus qui contiennent le terme t)

Le résultat final est une matrice de poids où chaque ligne représente un document et chaque colonne représente un terme, avec les valeurs des poids TF-IDF correspondantes. Cette matrice peut être utilisée pour représenter les documents dans un modèle de classification ou de clustering.

Exemple :

En prend le même exemple de la technique Bag-of-words, pour simplifier les calculs, nous allons ignorer les mots de liaison (stop words) comme "le", "et", etc.

Terme	TF_D1	TF_D2	TF_D3	IDF	TF*IDF(D1)	TF*IDF(D2)	TF*IDF(D3)
Chat	1	0	1	log(3/2)= 0.176	0.176	0	0.176
Chien	0	1	1	log(3/2)= 0.176	0	0.176	0.176
Mignon	1	0	0	log(3/1)= 0.477	0.477	0	0
Fidèle	0	1	0	log(3/1)= 0.477	0	0.477	0
Amis	0	0	1	log(3/1)= 0.477	0	0	0.477

Tableau 6 Exemple de TF-IDF.

c) Incorporation de mots (word embeddings) :

L'incorporation de mots (ou "word embeddings" en anglais) est une technique courante utilisée en traitement automatique du langage naturel pour représenter des mots sous forme de vecteurs de nombres réels dans un espace vectoriel de dimensions spécifiées. L'idée derrière les embeddings de mots est que des mots qui ont des significations similaires ou qui apparaissent dans des contextes similaires dans un corpus de texte devraient avoir des représentations vectorielles similaires dans l'espace vectoriel.

Les embeddings de mots sont souvent utilisés dans les tâches de traitement du langage naturel telles que la classification de texte, la traduction automatique, la recherche d'information et la génération de texte. Ils sont également utiles pour résoudre des tâches spécifiques telles que l'identification des entités nommées, la reconnaissance des relations entre les entités et la classification de sentiments.

Il existe plusieurs méthodes pour créer des embeddings de mots, voici quelques-unes des plus courantes :

- Word2Vec : est une méthode populaire pour créer des embéguines de mots elle a été réalisé par [18]. Elle utilise un réseau de neurones pour prédire le contexte d'un mot donné[19]. Word2Vec crée des vecteurs de mots de taille fixe, où chaque dimension représente un attribut sémantique différent. Word2Vec peut être utilisé pour créer des représentations de mots en utilisant différents algorithmes tels que : CBOW (Continous Bag-of-Words) et Skip-Gram. Dans CBOW, le modèle tente de prédire un mot en utilisant le contexte qui l'entoure, tandis que dans Skip-Gram, le modèle tente de prédire le contexte à partir du mot donné. Ces deux variantes peuvent être entraînées avec différentes tailles de fenêtres contextuelles et avec différentes dimensions de vecteurs de mots.

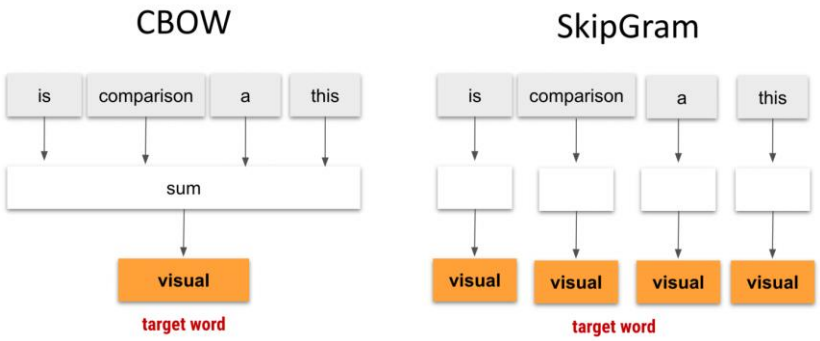

Figure 12 Exemple comparatif de Skip-Gram et CBOW pour Word2Vec.

En ce qui concerne l'entraînement des modèles Word2Vec, il existe deux approches principales : l'entraînement local et l'utilisation de modèles pré-entraînés. L'entraînement local implique l'entraînement du modèle Word2Vec à partir de zéro sur un corpus de texte spécifique. Cette approche nécessite un ensemble de données de texte suffisamment grand et peut prendre du temps et des ressources de calcul considérables. D'autre part, les modèles pré-entraînés Word2Vec sont des modèles qui ont été pré-entraînés sur des corpus de textes volumineux, tels que Wikipédia, et qui peuvent être utilisés pour extraire des vecteurs de mots pour des tâches spécifiques. Les modèles pré-entraînés peuvent être téléchargés gratuitement et facilement à partir d'Internet. Cela facilite l'utilisation de Word2Vec pour des tâches spécifiques, car il n'est pas nécessaire de former le modèle à partir de zéro.

- GloVe (Global Vectors for Word Representation) : est une méthode d'apprentissage non supervisé pour créer des embeddings de mots. Elle utilise la matrice de mot-contexte pour créer des vecteurs de mots. GloVe crée également des vecteurs de mots de taille fixe.
- FastText : est une méthode d'embeddings de mots développée par Facebook. Elle utilise une approche similaire à Word2Vec, mais au lieu de créer des embeddings pour des mots individuels, FastText crée des embeddings pour des sous-mots. Cela permet d'obtenir des représentations de mots plus robustes, même pour des mots rares ou mal orthographiés.
- ELMo (Embeddings from Language Models) : est une méthode d'embeddings de mots basée sur des modèles de langage. Elle crée des embeddings de mots en utilisant les sorties des couches cachées d'un réseau de neurones qui a été entraîné sur une grande quantité de données textuelles. Les embeddings de mots ELMo sont donc sensibles au contexte et peuvent prendre en compte des informations syntaxiques et sémantiques complexes.
- BERT (Bidirectional Encoder Representations from Transformers) : est une méthode d'embeddings de mots basée sur les réseaux de neurones Transformers. Elle utilise une approche d'apprentissage supervisé pour créer des embeddings de mots en utilisant des tâches de prévision de mots masqués et de classification de séquences. Les embeddings

de mots BERT sont également sensibles au contexte et peuvent prendre en compte des informations syntaxiques et sémantiques complexes.

III. 5 Techniques d'apprentissage profond pour l'analyse des sentiments

Les techniques d'apprentissage profond ont connu un grand succès dans le domaine de classification textuelle en particulier pour l'analyse des sentiments, parmi ces techniques on trouve les réseaux de neurones convolutifs et les réseaux de neurones récurrents avec ces deux variations les LSTMs et les BiLSTMs.

III. 5. 1 Réseaux de neurones convolutifs (CNN)

Sont une architecture de réseau de neurones qui utilise des filtres pour extraire des caractéristiques importantes d'une image ou d'une séquence de données. Les CNN ont été initialement développés pour la vision par ordinateur, mais ils peuvent également être utilisés pour le traitement du langage naturel[20], notamment pour l'analyse de sentiment.

Voici les principales caractéristiques des CNN :

1. Couches de convolution :
   Les CNN reposent sur des couches de convolution. Ces couches utilisent des filtres ou des noyaux pour analyser une séquence des données d'entrée en la multipliant avec le filtre. Chaque filtre détecte diverses caractéristiques de l'entrée, telles que les motifs, les textures et les arêtes. Pour extraire des caractéristiques plus avancées, les couches de convolution peuvent être empilées.
2. Couches de pooling :
   Les couches de mise en commun sont un outil couramment utilisé pour simplifier les résultats des couches de convolution en conservant les données les plus importants. Les deux types de mise en commun les plus courants sont la mise en commun maximale, qui permet d'extraire la valeur la plus élevée d'une zone spécifique et la mise en commun moyenne, qui permet d'extraire la valeur moyenne d'une zone donnée.
3. Couches entièrement connectées :
   Les couches entièrement connectées sont souvent utilisées à la fin du réseau pour effectuer une classification ou une régression. Elles prennent en entrée la sortie de la dernière couche de convolution et la transforment en une prédiction.

Les réseaux neuronaux convolutifs (CNN) se sont révélés très efficaces dans diverses tâches de vision artificielle telles que la classification d'images, la détection d'objets et la segmentation d'images. Ces réseaux sont également largement utilisés dans le traitement du langage naturel pour des tâches telles que l'analyse des sentiments, la classification et la génération de textes[20]. En outre, ils peuvent être formés à partir de zéro ou affinés à partir de modèles préexistants afin d'améliorer les performances et de minimiser le temps de formation.

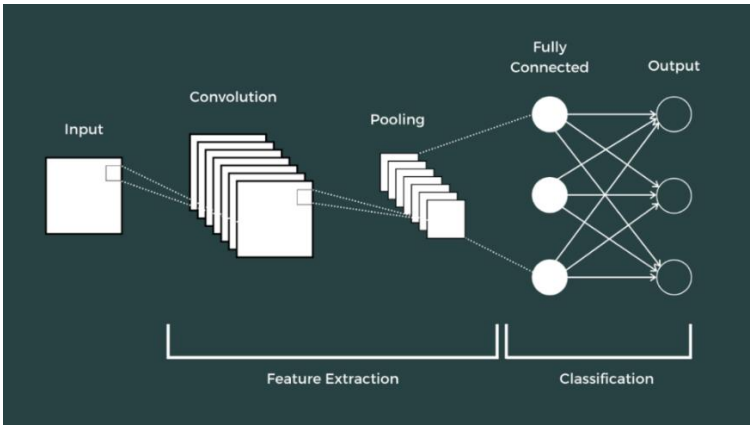

Figure 13 Les couches d'un réseau neurone convolutif.

III. 5. 2 Réseaux de neurones récurrents (RNN)

Les réseaux neuronaux récurrents (RNN) appartiennent à une catégorie de réseaux neuronaux artificiels spécialisés dans le traitement de données séquentielles, telles que des textes, des sons ou des images. Ils se distinguent des réseaux neuronaux traditionnels par la présence de connexions entre les neurones qui leur permettent de prendre en compte le contexte d'une séquence.

Les RNN sont idéaux pour l'analyse des sentiments, car ils peuvent reconnaître la structure temporelle des données et comprendre les connexions entre les différents composants d'une séquence de texte. En outre, les RNN peuvent être formés sur des données non étiquetées, ce qui les rend utiles dans les situations où les données étiquetées sont limitées.

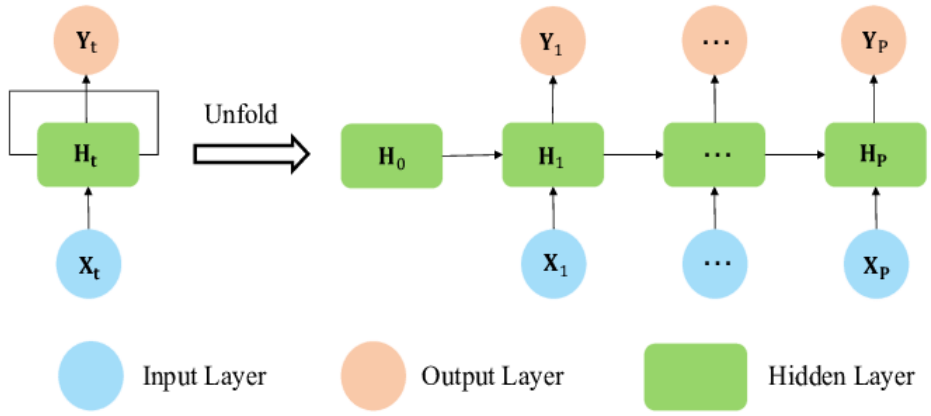

Figure 14 Architecture globale d'un réseau de neurone récurrent.

Les types d'entrée et de sortie pour les RNN dépendent du type de problème et de la structure de la couche RNN :

- Many-to-One : une séquence d'entrée est fournie en entrée à la couche RNN, et une seule sortie est produite à la fin de la séquence.
- One-to-Many : une seule entrée est fournie en entrée à la couche RNN, et une séquence de sortie est produite.

- Many-to-Many : une séquence d'entrée est fournie en entrée à la couche RNN, et une séquence de sortie est produite.
- One-to-One : une seule entrée est fournie en entrée à la couche RNN, et une seule sortie est produite.

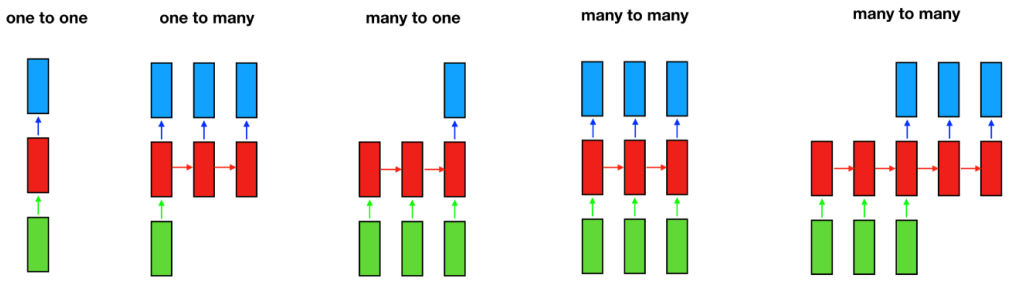

Figure 15 Les différents types d'entrée d'un réseau neurone reçurent.

Dans le cas de l'analyse des sentiments, le type d'entrée le plus courant est le many-to-one, où une séquence de mots est fournie en entrée à la couche RNN, et une seule sortie est produite pour classifier le sentiment de la phrase ou du document.

Cependant, l'entraînement des RNN peut être complexe en raison de la rétropropagation à travers le temps et du problème de disparition du gradient. Pour résoudre ces problèmes, des variantes de RNN ont été proposées, telles que les réseaux de neurones LSTM (Long Short-Term Memory) et les réseaux de neurones GRU (Gated Recurrent Unit).

III. 5. 3 Réseaux de neurones à mémoire à court terme (LSTM)

Le LSTM (Long Short-Term Memory) est un type de réseau de neurones récurrents (RNN) qui a été proposé pour résoudre le problème de la vanishing gradient, qui est une difficulté courante dans l'entraînement de réseaux de neurones profonds.

Le LSTM dispose de trois portes principales pour contrôler l'information qui est stockée dans l'état caché et qui est oubliée. Ces portes sont des couches de neurones qui contrôlent l'information qui est conservée ou oubliée à chaque étape de la séquence d'entrée.

- La porte d'oubli (Forget Gate) : Cette porte détermine la quantité d'information à oublier de l'état caché précédent en fonction de l'entrée actuelle.
- La porte d'entrée (Input Gate) : Cette porte détermine la quantité d'information à ajouter à l'état caché en fonction de l'entrée actuelle.
- La porte de sortie (Output Gate) : Cette porte détermine la quantité d'information à envoyer à la sortie en fonction de l'état caché actuel.

L'état caché du LSTM peut ainsi stocker des informations sur de longues périodes, sans subir de pertes d'informations importantes au fil du temps. Le LSTM est devenu très populaire pour l'analyse des sentiments, car il permet de mieux traiter les dépendances à long terme entre

les mots d'une phrase, qui sont souvent nécessaires pour comprendre le contexte et les nuances des opinions exprimées.

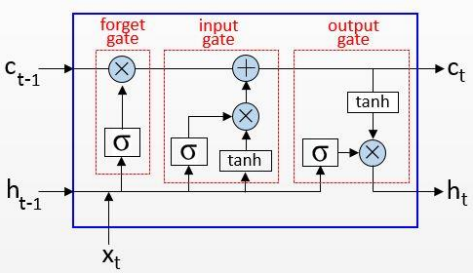

Figure 16 Architecture globale d'un réseau LSTM.

III. 5. 4 Réseaux de neurones récurrents bidirectionnels à mémoire à court terme (BiLSTM)

BiLSTM est une version avancée de RNN qui permet de prendre en compte les données passées et futures dans une analyse de séquence. Le principe de base du BiLSTM est similaire à celui d'un LSTM standard, mais avec deux couches de neurones, une pour considérer l'information passée et une autre pour considérer l'information future. Chacune de ces couches est constituée de cellules LSTM qui ont des portes de contrôle pour réguler l'information entrante et sortante.

Lors de la phase d'apprentissage, les données d'entrée sont présentées dans les deux directions, à la fois de gauche à droite et de droite à gauche, ce qui permet aux deux couches de BiLSTM de prendre en compte à la fois les informations passées et futures. Ensuite, les sorties des deux couches sont combinées pour obtenir une représentation plus complète de la séquence.

Le BiLSTM est un outil populaire pour l'analyse des sentiments, car il peut saisir les informations contextuelles et temporelles qui sont essentielles pour l'interprétation d'une séquence de texte. En outre, il est pratique pour traiter les structures syntaxiques complexes ou les mots qui sont ouverts à l'interprétation et peuvent avoir des significations différentes en fonction du contexte.

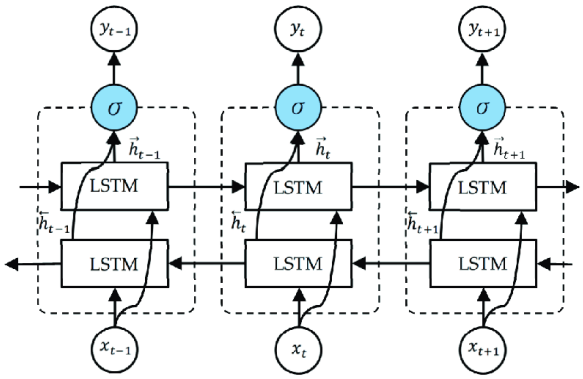

Figure 17 Architecture globale d'un réseau BiLSTM.

III. 6 Évaluation des modèles d'analyse des sentiments

L'évaluation des modèles d'analyse de sentiments est un processus important pour mesurer la performance des modèles construits et sélectionner le modèle le plus performant pour la tâche d'analyse de sentiments. Il existe plusieurs métriques couramment utilisées pour évaluer ces performances, notamment la matrice de confusion, cette dernière utilisée pour évaluer la performance d'un modèle de classification. Les prédictions du modèle sont comparées aux étiquettes réelles des données afin d'évaluer leur précision. Cependant la matrice de confusion est utilisée pour présenter le nombre de vrais positifs, de vrais négatifs, de faux positifs et de faux négatifs.

	Classe réelle positive	Classe réelle négative
Prédiction positive	Vrai positif (VP)	Faux positif (FP)
Prédiction négative	Faux négatif (FN)	Vrai négatif (VN)

Tableau 7 Matrice de confusion.

Par la suite, ces informations sont employées pour calculer plusieurs métriques d'évaluation utilisées dans le processus d'évaluation des modèles d'analyse de sentiments, notamment :

Exactitude (Accuracy) : mesure le taux de prédictions correctes sur l'ensemble des prédictions effectuées par le modèle.

accuracy = (vrais positifs + vrais négatifs) / (vrais positifs + faux positifs + faux négatifs + vrais négatifs)

Precision (precision) : mesure la proportion de vrais positifs parmi les prédictions positives.

precision = vrais positifs / (vrais positifs + faux positifs)

Rappel (recall) : mesure le taux de prédictions correctes sur l'ensemble des données positives du jeu de données.

recall = vrais positifs / (vrais positifs + faux négatifs)

Le score F1 (F1 score) : une mesure de précision et de rappel combinée, qui fournit une mesure globale de la performance du modèle.

F1 score = 2 * (precision * recall) / (precision + recall)

IV Travaux connexes

L'analyse des sentiments est un domaine de recherche en constante évolution. De nombreux travaux ont été menés pour améliorer les performances des modèles d'analyse des sentiments, notamment en ce qui concerne la langue arabe. Dans cette section, nous présentons un aperçu des travaux connexes réalisés dans ce domaine, en se concentrant sur les approches d'analyse des sentiments utilisées, les ensembles de données utilisés, ainsi que les résultats obtenus.

IV. 1 Revue de la littérature sur l'analyse des sentiments en langue arabe

Plusieurs recherches ont été menées dans le domaine de l'analyse de sentiments afin d'améliorer la précision et l'efficacité des modèles existants, de développer de nouvelles techniques et approches, et d'explorer de nouvelles applications pour l'analyse de sentiments, parmi lesquelles on peut citer :

1. Elfaik et Nfaoui.[21]Ils ont utilisé le modèle d'apprentissage profond BiLSTM avec la capacité d'extraire l'information contextuelle pour prédire le sentiment du texte arabe. Son modèle atteint une exactitude de 79.25 % sur l'ensemble de données ASTD, 91.82 % sur l'ensemble de données ArTwitter et 92.61 % sur l'ensemble de données Main-AHS, 89.70% sur l'ensemble de données Multi-Domain, 75.85% sur l'ensemble de données MPQA, 80.70% sur l'ensemble de données LABR. Ils ont constaté que BiLSTM avec le Sent2Seq donne des résultats plus élevés que d'autres modèles d'apprentissage profond (CNN et LSTM) sur la majorité des ensembles de données. Ceci est dû au fait que BiLSTM peut apprendre plus efficacement le contexte de chaque mot dans le texte, il accède à la fois aux fonctionnalités contextuelles précédentes et suivantes en combinant une couche cachée vers l'avant et une couche cachée vers l'arrière.et que BiLSTM peut découvrir des informations sémantiques plus riches et utiliser pleinement l'information contextuelle que LSTM.
2. Le travail [22] a été réalisé afin d'analyser les opinions sur les services de santé (Main_AHS), ils ont comparé deux modèles de deep learning à savoir, DNN et CNN avec word2vec,sur un corpus de données de 2026 hashtags tweets, et c'est un ensemble de données déséquilibré qui a 1398 tweets négatifs et 628 tweets positifs. Le CNN avait donné la meilleure exactitude.Cependant, le modèle CNN a été entrainé sur un petit ensemble de données et les deux modèles n'ont pas réglé le problème de négation de mots arabes. Afin d'augmenter la limitation de modèle CNN sur un petit ensemble de données, les mêmes auteures, ils ont proposé un autre modèle en [23]formé par une combinassent de CNN et de lexique sur word2vec construit à partir d'un grand corpus acquis de plusieurs revues arabes (Abu El-Khair Corpus). L'exactitude de leur modèle est passée de 90 % à 92 %.
3. Cette recherche [24] porte sur l'utilisation de différents algorithmes pour transformer un ensemble de données textuelles en vecteurs. Elle compare les algorithmes d'embedding de mots GloVe et fastText, ainsi que les réseaux de neurones à mémoire à court et long terme (LSTM) avec une, deux, et trois couches. Enfin, cette recherche évalue leur précision pour l'analyse d'opinions sur un ensemble de données en langue arabe. Elle

utilise l'ensemble de données ASAD constitué de 55 000 tweets annotés dans trois classes, positif, négatif et neutre, augmenté pour obtenir des proportions égales pour chaque classe. Selon les résultats de l'évaluation, le LSTM à trois couches avec fastText a atteint la meilleure précision de test, à 90,9%, dépassant tous les autres scénarios expérimentaux.

4. Deux modèles d'apprentissage profond ont été utilisés dans [25] avec le Word Embedding : la mémoire à court terme (LSTM) et les réseaux neurones convolutifs (CNN), en plus de trois techniques traditionnelles : Naïve Bayes, K-plus proche voisin (KNN), arbres de décision pour analyse des sentiments. Puis, ils ont combiné un modèle de CNN et (RNN) où ce modèle recueille des caractéristiques locales par l'intermédiaire de CNN comme entrée pour RNN pour analyse du sentiment arabe des textes courts. Une préparation appropriée des données a été effectuée pour chaque ensemble de données utilisé. Les expériences menées pour chaque ensemble de données par rapport au classificateur traditionnel d'apprentissage automatique KNN, NB, et les modèles d'apprentissage approfondis ; CNN et LSTM, ont produit des résultats impressionnants grâce à sa proposition modèle combiné (CNN-LSTM) avec un taux de réussite d'exactitude (accuracy) moyenne de 85,83 %, 86,88 % pour les ensembles de données HTL (ensemble composé de 15K commentaires arabes,13K utilisateurs ont effectué ces commentaires pour 8 100 Hôtels) et LABR (ensemble de données à plus de 16 448 critiques de livres en arabe) respectivement.
5. Dans [26]. Les auteurs ont utilisé trois modèles CNN, LSTM, CNN+LSTM sur un corpus de données ASTD qui se compose de 10,000 tweets, distribués en 4 classes (positive, négative, neutre et objective), puis ils ont supprimé les tweets de classe objective. Donc le vrai nombre de tweets utilisé est 3 315, avec une répartition des classes plus équilibrée (24 % positifs, 51 % négatifs et 25 % neutre). Ils ont utilisé le Word embedding (AraVec skip gram) le f1 score des modèles CNN, LSTM, CNN+LSTM sont respectivement 64.09%,62.08%,64.46%. Le modèle n'utilise aucune ingénierie de fonctionnalités pour extraire des fonctionnalités spéciales ou des modules complexes comme une banque d'arbres de sentiments.
6. Dans cette étude [27], les auteurs ont utilisé trois topologies de réseaux de neurones à apprentissage profond pour des modèles de classification : (LSTM), (CNN), et un modèle d'ensemble combinant les avantages des deux modèles pour améliorer les performances de prédiction, avec le Word Embedding. Ils ont également utilisé une méthode d'estimation d'optimisation d'hyperparamètres pour améliorer les performances des réseaux de neurones. Les modèles ont été entraînés et testés sur un ensemble de données composé de textes en arabe standard moderne et dialectal collectés sur Twitter (AraSenTi dataset) étiquetés avec quatre sentiments : positif, négatif, neutre et mixte. Cependant, le nombre de tweets mixtes dans le jeu de données est relativement faible (1822 tweets), comparé aux autres sentiments qui varient entre 4 900 et 6 200 tweets. Par conséquent, dans cette étude, les tweets avec un sentiment mixte ont été ignorés pour éviter tout problème de classification de données déséquilibrées qui pourrait entraîner un biais de classification en faveur des trois autres étiquettes. Trois étiquettes ont donc été utilisées pour les expériences menées : positif, négatif et neutre.

   Le modèle d'ensemble a obtenu la précision la plus élevée, avec un score de 96.7% sur l'ensemble de test.

7. Dans leur document[28], Ombabi et al. Proposent un modèle d'apprentissage en profondeur pour l'analyse des sentiments en arabe, qui combine avec succès une architecture CNN à une couche avec deux couches LSTM. Cette architecture est soutenue par le modèle d'embedding de mots FastText en tant que couche d'entrée. Les expériences menées sur un corpus multi-domaines (ensemble de données composé de 4.000 avis répartis en 2.000 positifs et 2.000 négatifs) ont montré les performances remarquables de ce modèle avec des exactitudes, rappels, scores F1 et précisions respectives de 89,10%, 92,14%, 92,44% et 90,75%. Cette étude a également validé l'effet des techniques d'embedding de mots sur la classification des sentiments en arabe et a conclu que le modèle FastText est une alternative plus pertinente pour apprendre les informations sémantiques et syntaxiques.

Article	Année	Dataset	Modèle	Résultat
Alwehaibi et al.[27]	2022	AraSenTi (15 945 tweets)	CNN-LSTM	F1 score = 96.7%
Setyanto et al.[24]	2022	ASAD (55 000 tweets)	LSTM	Exactitude=90,9%
Elfaik et Nfaoui.[21]	2021	ASTD(10k) ArTwitter Main-AHS(2026 tweets) Multi-Domain MPQA LABR(16448 critiques)	BILSTM	Exactitude = 79.25 %(ASTD) Exactitude=91.82 %(ArTwitter) Exactitude=92.61%(Main-AHS) Exactitude=89.70%(Multi-Domain) Exactitude=75.85%(MPQA) Exactitude=80.70%(LABR)
Ombabi et al.[28]	2020	multi-domaines(4 000 avis)	CNN-LSTM	Exactitude= 89.10% Rappel= 92.14% F1 score= 92.44% Précision= 90.75%
Elzayady, Badran, et Salama.[25]	2020	HTL(15k) LABR(16448)	CNN-LSTM	Exactitude =85.83 %(HTL) Exactitude =86.88 %(LABR)
Heikal et al.[26]	2018	ASTD(10k)	CNN LSTM CNN-LSTM	F1 score=64.09% F1 score=62.08% F1 score=64.46%
Alayba et al.[23]	2018	Main_AHS (2026 tweets)	CNN	Exactitude = 92 %
Alayba et al.[22]	2017	Main_AHS (2026 tweets)	CNN	Exactitude = 90 %

Tableau 8 Résumé sur les travaux connexe à l'analyse des sentiments.

IV. 2 Synthèse des travaux connexes

D'après les travaux liés à l'analyse de sentiments qui ont été citées, on peut sortir avec les points suivants :

- Les travaux précédents sur l'analyse des sentiments et la détection d'opinions ont porté sur différentes approches variées entre les techniques supervisées et non supervisées. Les modèles d'apprentissage profond donnent des meilleurs résultats par rapport aux techniques traditionnelles, comme cela a été mentionné dans le quatrième travail.
- Les travaux connexes sur l'analyse des sentiments en langue arabe, ressort que l'utilisation de réseaux de neurones convolutive (CNN) et de réseaux de neurones

récurrents (RNN) combinés peut conduire à de meilleurs résultats lorsqu'on représente les données en utilisant des embeddings de mots. En effet, cette combinaison de modèles permet de capturer à la fois les informations locales et les informations de contexte dans les données textuelles, ce qui peut améliorer la performance de l'analyse de sentiment, cette combinaison a été largement utilisée dans la littérature pour résoudre des tâches de l'analyse des sentiments, et a souvent donné des résultats prometteurs.
Par exemple, dans les travaux 4,5,6 et 7, un combinaison de CNN et LSTM donne des bons résultats en raison de la capacité de LSTM à capturer les dépendances temporelles à long terme dans une séquence de mots, tandis que le CNN est capable de capturer les caractéristiques locales et les motifs dans un texte. En combinant ces deux architectures, on peut donc obtenir des modèles plus robustes et performants, en exploitant à la fois l'information de contexte et l'information locale des mots dans un texte.

- D'après les travaux 2, 5, 6 et 7, nous avons pu conclure que l'ensemble de données joue un rôle important pour obtenir de bons résultats, notamment :
  - ✓ Les données d'entraînement pour chaque classe doivent être équilibrées au maximum.
  - ✓ Plus l'ensemble de données est grand, plus le modèle nous donne de meilleures performances.
- Plusieurs techniques d'embedding de mots ont été utilisées pour la classification des sentiments en arabe, telles que GloVe, fastText et Word2vec. Les résultats des expériences menées avec ces techniques varient selon les ensembles de données et les algorithmes utilisés. Cependant, en général, les chercheurs ont constaté que l'utilisation de ces techniques d'embedding de mots améliore les performances de classification des sentiments en arabe, en particulier lorsqu'elles sont associées à des modèles d'apprentissage en profondeur tels que CNN, LSTM et BiLSTM.

Donc, ça serait intéressant de travailler sur les deux axes, l'axe de représentation des données et l'axe d'utilisation de modèles de Deep learning.

V Conclusion

En conclusion, ce premier chapitre nous a permis de faire un état de l'art sur l'analyse des sentiments, en mettant en évidence les différents niveaux d'analyse, les approches existantes, ainsi que les défis spécifiques à la langue arabe. De plus, nous avons abordé les bases du Deep Learning, ses techniques utilisées pour l'analyse des sentiments, telles que les réseaux de neurones convolutifs (CNN) et les réseaux de neurones récurrents (RNN), et souligné l'importance de l'évaluation des modèles d'analyse des sentiments. Enfin, nous avons passé en revue la littérature existante sur l'analyse des sentiments en langue arabe et présenté les travaux connexes.

Cette revue de littérature nous fournit une base solide pour la conception et la mise en œuvre de nos propres modèles d'analyse des sentiments dans le chapitre suivant.

Chapitre II Conception des modèles d'analyse des sentiments

I Introduction

Dans ce chapitre, nous allons présenter globalement l'architecture de notre système d'analyse des sentiments en langue arabe, ainsi que ses différents composants. Nous allons également décrire les modèles proposés pour effectuer la détection d'opinions.

II Description de l'architecture globale de notre système

Pour concevoir nos modèles, on a suivi l'architecture suivante :

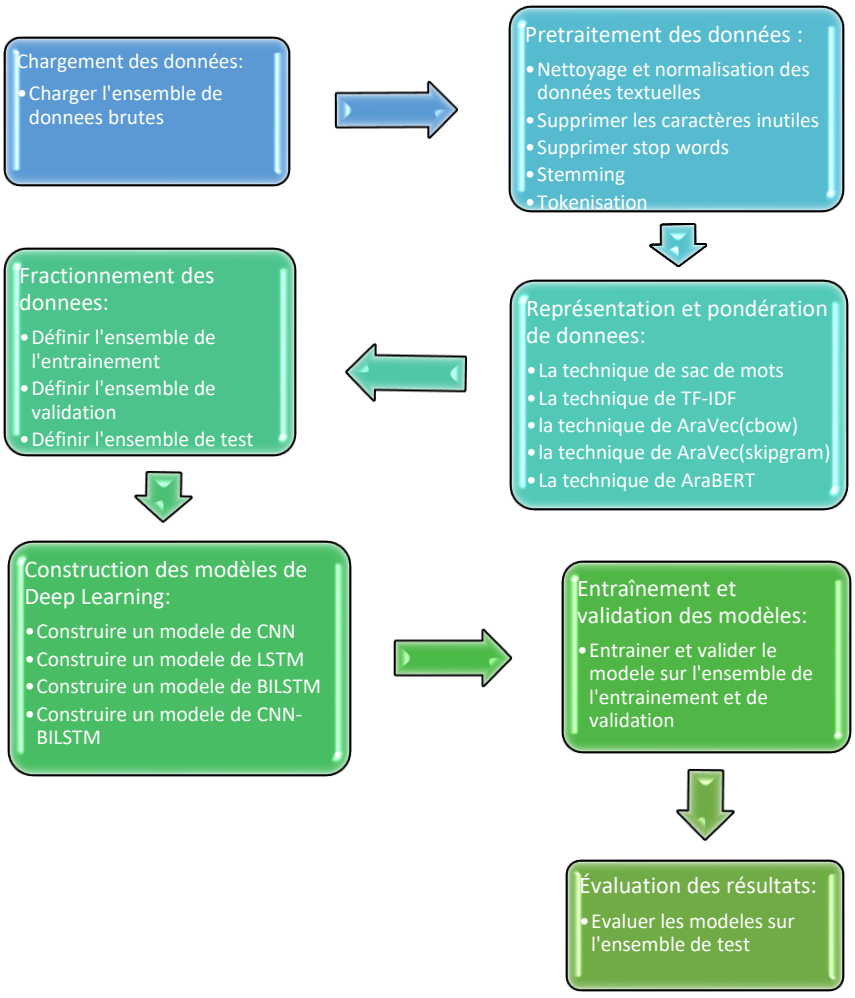

Figure 18 Architecture globale du système.

II. 1 Chargement des données

Avant de commencer par le prétraitement de données, il faut charger l'ensemble de données qu'est une étape essentielle de l'analyse des données en Deep Learning. C'est le processus de récupération des données textuelles brutes à partir de fichiers ou des sources en ligne et leur stockage dans une structure de données appropriée pour l'analyse. Les données textuelles peuvent être stockées dans différents formats tels que CSV, TSV, JSON, TXT, XML ou base de données, selon leur origine.

II. 2 Prétraitement des données

II. 2. 1 Nettoyage et normalisation des données textuelles

Dans le cas de la langue arabe, le nettoyage et la normalisation sont particulièrement importants en raison de la présence de diacritiques et de lettres ayant différentes formes selon leur position dans le mot. Dans cette section, nous allons décrire les différentes étapes de nettoyage et de normalisation des données textuelles en arabe, tel que :

- **La suppression du diacritique :**
  Les diacritiques sont des signes qui sont placés au-dessus ou en dessous des lettres dans certaines langues, y compris l'arabe, pour indiquer des variations de prononciation ou de signification. Cependant, dans certaines tâches d'analyse de texte, le diacritique peut ne pas être nécessaire et peut même ajouter du bruit au texte. Dans ce cas, la suppression du diacritique peut être effectuée pour réduire la complexité du texte et faciliter le traitement par des modèles de Deep Learning.
- **La suppression de grand espace :**
  Cette étape consiste à supprimer tous les grands espaces ou plusieurs espaces consécutifs dans le texte.
- **La normalisation :**
  La normalisation en langue arabe est une étape importante pour le traitement des données en langue arabe et peut améliorer la précision et la qualité des modèles de traitement du langage naturel en arabe, consiste à remplacer les lettres qui se ressemblent ou qui ont des formes différentes selon leur position dans le mot par une forme standard.

Caractère	Remplacement
"ى"	"ي"
"ؤ"	"ء"
"ئ"	"ء"
"ة"	"ه"
"گ"	"ك"
"وال"	"و ال"
[إأآا]	"ا"

Tableau 9 La normalisation des caractères en langue arabe.

II. 2. 2 Suppression des caractères inutiles

Dans cette optique, plusieurs étapes de prétraitement peuvent être effectuées pour supprimer les caractères inutiles :

- **La suppression de la ponctuation :**
  La suppression de ponctuation est une étape de prétraitement du texte dans laquelle on élimine tous les signes de ponctuation présents dans le texte. Cela inclut des caractères ponctuations courantes en arabe et en anglais tels que les points, les virgules, les points-virgules, les guillemets, les parenthèses, les crochets, les tirets, les points d'exclamation, etc.

- **La suppression des émojis :**
  Cette étape permettre de supprimer tous types d'émojis tels que les émoticônes, les symboles, les drapeaux, etc.

Émoji	Description
😀	Visage souriant : utilisé pour exprimer la joie ou l'amusement
😂	Visage avec larmes de joie : utilisé pour exprimer un rire fort ou une grande hilarité.
😍	Visage souriant avec des yeux en forme de cœur : utilisé pour exprimer un amour ou une admiration profonde
😕	Visage pensif : utilisé pour exprimer une réflexion ou une incertitude
🙏	Mains jointes : utilisé pour exprimer une prière ou une demande
👍	Pouce en l'air : utilisé pour exprimer une approbation ou un accord

Tableau 10 les émojis avec leurs significations

- **La suppression des caractères répétitifs :**
  Il peut y avoir des caractères répétitifs dans un texte, tels que des lettres, des chiffres ou des symboles, qui peuvent perturber l'analyse. Supprimer ces caractères peut améliorer la qualité du texte.

Texte original	Texte après suppression des caractères répétitifs
مرحباااا كيف حالكمممممم ؟؟؟؟	مرحبا كيف حالكم ؟
أحببب هذا الفيلم كثيييراً، كان ممتعاً جداً	أحب هذا الفيلم كثيراً، كان ممتعاً جداً

Tableau 11 Comparaison entre des textes originaux et nettoyés.

- **La suppression des caractères non-arabes :**
  Pour certaines tâches d'analyse de la langue arabe, il peut être nécessaire de supprimer les caractères qui ne font pas partie de la langue arabe. Cela peut inclure des lettres étrangères [A...Z, a …z]. Des symboles ou des chiffres [١….٩, 1….9, I.…IX].

- **La suppression des URLS, des hashtags et des mentions :**
  Cette étape consiste à supprimer tous les liens URL, les hashtags (#) et les mentions (@) dans le texte. Ces éléments sont souvent utilisés dans les médias sociaux et les plateformes en ligne pour identifier des sujets spécifiques.

II. 2. 3 Suppression des stop words

La suppression des stop words est une étape importante du prétraitement des données pour l'analyse des sentiments en arabe, tout comme pour d'autres langues. Les stop words sont des mots vides tels que "و" (et), "أنا" (je), "هذا" (ce), etc., qui ne contribuent pas beaucoup au sens global de la phrase et peuvent être ignorés sans perdre trop d'informations.

Il existe plusieurs bibliothèques de traitement de texte pour la langue arabe qui incluent des listes de stop words prédéfinies, telles que NLTK-Arabic et Arabic Stop Words List. Cependant, il est important de vérifier et de personnaliser la liste de stop words en fonction du corpus de texte spécifique et de la tâche d'analyse de sentiment. Par exemple, on peut garder les mots de négations, car ils peuvent inverser le sens de la phrase et affecter le sentiment général exprimé. Comme la phrase "لم أحب ذلك" ("Je n'ai pas aimé ça") exprime un sentiment négatif, alors que "أحب ذلك" ("J'aime ça") exprime un sentiment positif.

II. 2. 4 Stemming

Est un processus de traitement automatique du langage naturel (NLP) qui est utilisé pour réduire les mots à leur racine, appelée "stem" en anglais. Il vise à normaliser les mots en supprimant les suffixes et les préfixes les plus courants afin de regrouper les mots qui partagent la même racine, même s'ils sont écrits différemment. On trouve deux types de stemming, le stemming léger (light stemming) et le stemming lourd (heavy stemming) ,ils sont utilisés en fonction du niveau d'analyse souhaité[29].

a) **Le stemming léger :** consiste à supprimer les affixes tels que les préfixes et les suffixes, mais les variations morphologiques plus complexes sont conservées.
   Dans notre travail, on a utilisé le stemming léger tel que le stemmer ISRI et Tashaphyne :

- Le stemmer ISRI : est un stemmer pour la langue arabe de l'Institut de recherche en sciences de l'information (ISRI) partage de nombreuses caractéristiques avec le stemmer Khoja.[30] Cependant, il n'utilise pas de dictionnaire de racines pour la recherche. De plus, si un mot ne peut pas être raciné, il est normalisé par le stemmer ISRI (par exemple, en supprimant certains déterminants et motifs de fin) au lieu de laisser le mot inchangé. De plus, il définit des ensembles de marques diacritiques et de classes d'affixes. Le stemmer ISRI a été montré pour donner de bonnes améliorations aux tâches linguistiques telles que le regroupement de documents[31].
- Le stemmer Tashaphyne : Tashaphyne est un stemmer et segmenteur léger pour la langue arabe. La fonction première de cet outil est d'aider à l'extraction légère en éliminant les préfixes et les suffixes. Il propose également toutes les segmentations possibles pour améliorer le processus. Cela signifie qu'il a la capacité d'examiner toutes

les variations possibles de préfixes et de suffixes pour dériver diverses structures de mots arabes. Contrairement à d'autres stemmers tels que Khoja stemmer, ISRI stemmer, Assem stemmer et Farasa stemmer, Tashaphyne offre à la fois le stemming et l'extraction de la racine.

Il propose des préfixes et des suffixes prédéfinis ainsi que la possibilité d'utiliser des listes personnalisées de préfixes et de suffixes. Cela permet de mieux contrôler les différents aspects et de créer des stemmers uniques sans avoir à modifier le code[1].

b) **Le stemming lourd :** est une approche qui vise à supprimer non seulement les affixes courants, mais aussi les variations morphologiques plus complexes.

II. 2. 5 Tokenisation

La tokenisation est une étape importante dans l'analyse de texte, car elle permet de convertir le texte en une forme que les modèles de Deep Learning peuvent traiter. En divisant le texte en unités discrètes, qui peuvent être ensuite utilisées comme entrée pour les modèles d'analyse de sentiments. La tokenisation permet aux modèles de comprendre la structure du texte et de repérer les relations entre les différents mots.

II. 3 Représentation et pondération de données

Dans cette partie, nous abordons les différentes techniques de représentation et de pondération des données utilisées dans notre travail pour l'analyse de sentiments en langue arabe. Les méthodes comprennent le sac de mots, la pondération TF-IDF, ainsi que les modèles de représentation de mots tels que Word2Vec et AraBERT. Chacune de ces techniques permet d'extraire des informations importantes des données textuelles en vue d'une analyse plus approfondie des sentiments exprimés.

II. 3. 1 Sac de mots (bag-of-words)

Le sac de mots est une technique de représentation de texte qui consiste à considérer chaque document comme un ensemble de mots et à représenter le document par un vecteur qui indique la présence ou l'absence de chaque mot dans le document. Les avantages de cette méthode sont sa simplicité et sa facilité de mise en œuvre, mais elle ne prend pas en compte l'ordre des mots ni les relations sémantiques entre eux.

II. 3. 2 Tf-IDF

Comme mentionné dans la section précédente, la méthode TF-IDF évalue l'importance d'un terme dans un document en prenant en compte sa fréquence d'apparition dans le document et la fréquence du terme dans l'ensemble du corpus. De cette façon, les termes qui apparaissent fréquemment dans le document, mais pas dans le corpus, auront un score plus élevé que les termes qui apparaissent fréquemment dans le corpus, mais pas dans le document.

II. 3. 3 Word embedding

On a utilisé deux modèles pour l'embedding des mots : AraVec et AraBERT. AraVec est un modèle de représentation de mots basé sur Word2Vec, qui permet de représenter chaque

[1] « Tashaphyne · PyPI ». https://pypi.org/project/Tashaphyne/ (consulté le 14 mai 2023).

mot en tant que vecteur dense de nombres réels. D'un autre côté, AraBERT est un modèle pré-entraîné de BERT spécialement conçu pour la langue arabe, qui permet également de représenter chaque mot en tant que vecteur de haute dimension. Les deux modèles ont été utilisés pour extraire les représentations vectorielles de mots dans le cadre de notre étude sur l'analyse de sentiments en langue arabe.

- **AraVec :**
  Nous avons utilisé le word embedding AraVec [32] qui a été pré-entraîné sur les données Twitter suivant l'approche Word2vec. AraVec est un projet open source de représentation distribuée de mots (embedding de mots) pré-entraîné qui vise à fournir à la communauté de recherche en NLP arabe des modèles d'embedding de mots puissants et gratuits à utiliser. Il est construit sur trois domaines de contenu arabe différents : les tweets, les pages du Web et les articles arabes de Wikipédia.
  Chaque tweet est représenté par un vecteur 2D de dimension n×d. Où n est le nombre de mots dans le tweet et de la longueur de dimension de la représentation vectorielle du mot. Nous utilisons le modèle AraVec skip gram et cbow de dimension 300.

Modèle	Nb documents	Nb vocabulaires	Dimension	Technique
Twitter-CBOW	66 900 000	331 679	300	CBOW
Twitter-Skipgram	66 900 000	331 679	300	Skip-Gram

Tableau 12 Les caractéristiques des modèles Twitter CBOW et Skip-gram.

- **AraBERT :**
  Est un modèle de langage pré-entraîné pour la langue arabe, développé par une équipe de chercheurs de l'Université américaine de Beyrouth et du laboratoire de recherche en intelligence artificielle "AUB MC Lab "[33]. Le modèle est construit sur l'architecture de Transformer et est entraîné sur un vaste corpus de texte en arabe. Il est capable de saisir les subtilités de la langue arabe, y compris les variations grammaticales et les mots conjugués, et de capturer les relations sémantiques et syntaxiques dans la langue en apprenant une représentation vectorielle dense pour chaque mot. De plus, le modèle est disponible gratuitement en open source, permettant ainsi un accès ouvert à tous[2].

II. 4 Fractionnement des données

L'étape de fractionnement des données, est une étape importante dans l'analyse des données pour éviter le surapprentissage (overfitting) et pour garantir que le modèle est capable

[2] « arabert/arabert at master · aub-mind/arabert · GitHub ». https://github.com/aub-mind/arabert/ (consulté le 17 mai 2023).

de généraliser sur des données qu'il n'a jamais vues auparavant. Elle consiste à diviser un ensemble de données en plusieurs ensembles distincts pour pouvoir les utiliser de manière optimale dans l'apprentissage d'un modèle.

En général, les données sont divisées en trois ensembles distincts : l'ensemble d'entraînement (training set), l'ensemble de validation (validation set) et l'ensemble de test (test set). L'ensemble d'entraînement est utilisé pour entraîner le modèle tandis que l'ensemble de validation est utilisé pour ajuster les paramètres du modèle. Enfin, l'ensemble de test permet d'évaluer les performances du modèle sur des données qu'il n'a jamais rencontrées précédemment.Dans notre cas, on a divisé le dataset en : 80 % pour l'entrainement, 10%pour la validation et 10% pour le test.

II. 5 Construction des modèles

Le processus de construction d'un modèle en général comprend plusieurs étapes. La première consiste à déterminer l'ensemble d'entrée qu'il soit structuré comme des tableaux de données ou non structuré comme des images. Ensuite, ces entrées passent par la phase d'entraînement, où le modèle apprend à partir de ces données et ajuste ses paramètres tels que les poids et les biais à chaque époque pour minimiser la fonction de perte en utilisant des algorithmes d'optimisation. Pendant l'entrainement, il est important de vérifier la performance du modèle à chaque époque afin de détecter s'il y'a surapprentissage ou toute autre problème. Cela se fait en utilisant un ensemble de validation distinct. La sortie du modèle est définie en fonction de la tâche qu'il effectue. Par exemple, pour un modèle de classification d'images la sortie peut être la classe prédite de l'image, tandis que pour un modèle de détection d'opinions la sortie peut être une classe indiquant si l'opinion est positive ou négative. Enfin, on peut considérer qu'un modèle est réalisé lorsqu'il a été entrainé avec succès sur les données d'entraînement et qu'il a montré de bonnes performances sur les données de validation, cela peut être observé à travers le diagramme de la fonction de perte et des métriques d'évaluation.

En s'inspirant des modèles utilisés et mentionnés dans les travaux connexes concernant l'analyse des sentiments en langue arabe présentés dans le chapitre de l'état de l'art, nous avons développés plusieurs modèles d'analyse des sentiments avec des hyperparamètres différents. En particulier, nous avons opté pour l'utilisation de réseaux de neurones convolutions (CNN), de réseaux de neurones récurrents (RNN) tels que les Long Short-Term Memory (LSTM) et Bidirectional LSTM (BiLSTM), ainsi que pour une combinaison de CNN avec RNN (nous avons choisi le BiLSTM). Nous avons utilisé une combinaison de CNN avec BiLSTM pour exploiter à la fois l'information de contexte et l'information locale des mots dans un texte.

Les paramètres et les hyperparamètres sont deux concepts importants en apprentissage automatique. À savoir que les paramètres sont des variables ajustées pendant l'entraînement du modèle afin de minimiser la fonction de perte, comme le poids et le biais. Mais les hyperparamètres sont des variables choisies avant l'entraînement et ajustées selon la performance du modèle.

Avant de plonger dans l'entraînement d'un modèle de Deep Learning, il est important de déterminer les hyperparamètres, qui représentent les paramètres clés de l'architecture du

modèle. Il est donc crucial de choisir judicieusement les hyperparamètres pour obtenir des résultats fiables et précis lors de la phase de test. Parmi ces hyperparamètres, on retrouve :

- Dimensionnalité de l'espace d'embedding : La dimensionnalité de l'espace d'embedding est un hyperparamètre important pour les modèles de TALN. Lorsqu'on entraîne un modèle , les données textuelles sont représentées par des vecteurs de nombres réels dans un espace d'embedding avec un nombre de dimension prédéfini. Une dimensionnalité plus élevée permet de capturer des relations complexes entre les mots et de représenter efficacement la sémantique des phrases. Cependant, une dimensionnalité excessive peut entraîner un surapprentissage et des temps de calcul plus longs.
- Nombre de couches : c'est le nombre de couches du réseau de neurones utilisées pour apprendre la représentation des séquences de mots.
- Nombre de filtres : c'est le nombre de filtres à appliquer lors de la convolution des séquences de mots. En d'autres termes, chaque filtre est une petite fenêtre qui glisse sur la matrice d'entrée (dans ce cas, la matrice d'embedding de mots) pour extraire des caractéristiques importantes.
- Taux d'abandon : c'est le taux de dropout appliqué pour régulariser le modèle et éviter le surapprentissage.
- La fenêtre de pooling : la fenêtre de pooling est une technique de réduction de dimensionnalité couramment utilisée dans les réseaux de neurones pour extraire les caractéristiques importantes d'une représentation vectorielle. Elle consiste à diviser une séquence de vecteurs en segments de taille fixe et à appliquer une opération de réduction sur chaque segment pour obtenir une seule valeur représentative. Cette opération de réduction peut être une moyenne, une somme, un maximum, etc. La fenêtre de pooling peut être utilisée pour réduire la taille de la représentation vectorielle et améliorer les performances de classification ou de régression en évitant le surapprentissage.
- Fonction d'optimisation : la fonction d'optimisation est utilisée pour ajuster les poids du modèle de manière à minimiser la fonction de perte (loss function) pendant l'entraînement. En d'autres termes, elle permet de trouver les meilleurs paramètres du modèle pour obtenir les résultats les plus précis possibles. Il existe de nombreuses fonctions d'optimisation différentes, chacune ayant ses avantages et ses inconvénients. Parmi les plus courantes :
    - Gradient Descent (descente de gradient) : c'est de l'algorithme d'optimisation le plus simple qui consiste à mettre à jour les poids en fonction du gradient de la fonction de perte. Cependant, il peut être convergé lentement et il peut ne pas être capable de trouver le minimum global de la fonction de perte.
    - Adam : l'algorithme d'optimisation Adam est utilisé pour la formation de modèles d'apprentissage profond. Il s'agit d'une extension de la descente de gradient stochastique. Dans cet algorithme d'optimisation, les moyennes courantes des gradients et des seconds moments des gradients sont utilisés. Il est utilisé pour calculer les taux d'apprentissage adaptatifs pour chaque

paramètre[3] . Il est généralement plus rapide que la descente de gradient standard et peut mieux éviter les minima locaux.

- BATCH_SIZE : correspond à la taille de l'échantillon de données d'entraînement qui est présenté à chaque itération lors de l'entraînement du modèle. En général, si le BATCH_SIZE est trop petit, l'entraînement sera plus lent, car le modèle devra effectuer de nombreuses itérations sur de petites quantités de données. En revanche, si le BATCH_SIZE est trop grand, le modèle risque de manquer de mémoire pour stocker tous les exemples en même temps, ce qui peut entraîner des erreurs de calcul et un temps d'entraînement plus long. En théorie, la taille du batch devrait avoir un effet sur le temps d'entraînement plutôt que sur les performances de test. Il est possible de l'optimiser indépendamment des autres paramètres en comparant les courbes d'entraînement une fois que les autres paramètres et hyperparamètres ont été sélectionnés. En théorie, la taille du batch devrait avoir un effet sur le temps d'entraînement plutôt que sur les performances de test. Il est possible de l'optimiser indépendamment des autres paramètres en comparant les courbes d'entraînement une fois que les autres paramètres et hyperparamètres ont été sélectionnés[34]. Il est courant de choisir une valeur entre 16 et 512 pour le BATCH_SIZE.
- Nombre d'époques : spécifie le nombre de fois où le modèle est exécuté sur l'ensemble de données pendant l'entraînement. Commencez avec un nombre faible d'époques, puis on observe les performances du modèle sur l'ensemble de validation. Si les performances continuent de s'améliorer, l'augmentation du nombre d'époques peut être bénéfique. Cependant, il est important de surveiller la courbe d'apprentissage et de s'arrêter lorsque les performances du modèle sur les données de validation commencent à se dégrader.

II. 5. 1 Le modèle CNN

L'architecture de notre modèle CNN est composée de plusieurs couches. Tout d'abord, nous avons une couche d'entrée qui spécifie la longueur maximale des séquences d'entrée. Ensuite, on a utilisé une couche d'embedding pour représenter les mots sous forme de vecteurs denses dans un espace continu. Cela est particulièrement utile dans le cas de la classification en utilisant des embeddings de mots.

Après la couche d'embedding, on a ajouté deux couches de convolutions. La première couche a 64 filtres, et la deuxième couche a 32 filtres. Chaque couche de convolution est suivie d'une couche de dropout, où 20% des neurones sont désactivés aléatoirement pour éviter le surapprentissage. Ensuite, on a utilisé une couche de max pooling en utilisant la fonction MaxPooling1D. Cette opération permet de réduire la dimensionnalité des caractéristiques extraites tout en préservant les informations les plus importantes.

[3] « Optimisation Adam — DataFranca ». https://datafranca.org/wiki/Optimisation_Adam (consulté le 16 mai 2023).

Après la couche de max pooling, on a utilisé une couche de flatten pour transformer la sortie en un vecteur à une seule dimension, afin de le rendre compatible avec la couche dense qui suit avec 32 unités. En plus de cela, nous avons introduit une autre couche de dropout pour éviter le surapprentissage.

Enfin, on a ajouté une couche dense avec deux neurones, ce qui indique qu'il y a deux classes de sortie possibles.

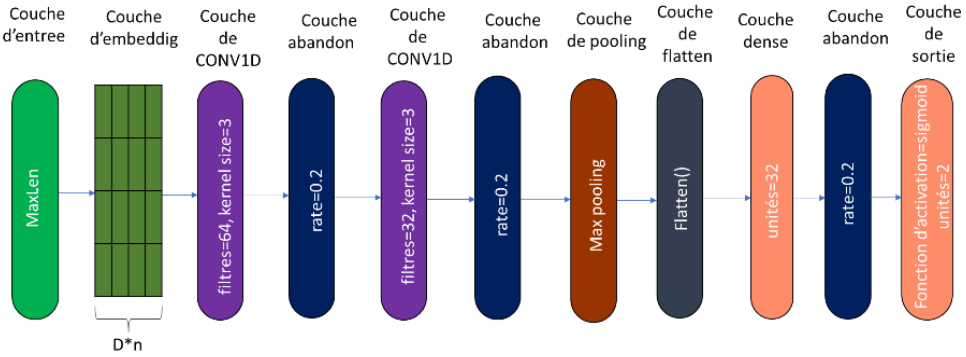

Figure 19 Architecture de CNN.

II. 5. 2 Le modèle LSTM

L'architecture de notre modèle LSTM a été construite en utilisant plusieurs couches. Nous avons commencé par une couche d'entrée qui a été configurée pour spécifier la longueur maximale des séquences d'entrée. Ensuite, une couche d'embedding a été employée pour représenter les mots sous forme de vecteurs denses dans un espace continu, similaire à ce qui a été fait dans le modèle CNN.

Par la suite, nous avons introduit une couche LSTM comprenant 64 unités. Cette couche permet de saisir les relations temporelles et séquentielles dans les données. Après cette couche de LSTM, une couche de dropout a été ajoutée afin de désactiver 20% des neurones et d'éviter le surapprentissage. Nous avons ensuite inséré une deuxième couche LSTM avec 32 unités, suivie d'une autre couche de dropout.

Une fois les couches LSTM complétées, nous avons inclus une couche dense contenant 32 unités. Nous avons également appliqué une couche de dropout supplémentaire pour prévenir le surapprentissage.

Enfin, nous avons ajouté une couche de sortie dense composée de deux neurones, indiquant qu'il existe deux classes de sortie possibles. La fonction d'activation sigmoid a été utilisée dans cette couche afin d'obtenir les probabilités de chaque classe.

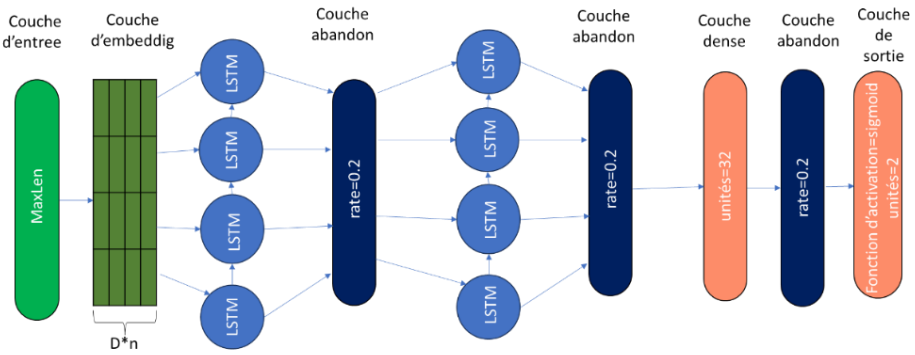

Figure 20 Architecture de LSTM.

II. 5. 3 Le modèle BiLSTM

Nous avons utilisé une architecture similaire à celle du modèle LSTM, à une différence près : nous avons remplacé les deux couches LSTM par deux couches LSTM bidirectionnelles. La première couche LSTM bidirectionnelle compte 64 unités, tandis que la deuxième couche LSTM bidirectionnelle compte 32 unités. Ces couches permettent de prendre en compte à la fois les informations passées et futures lors de la modélisation des séquences.

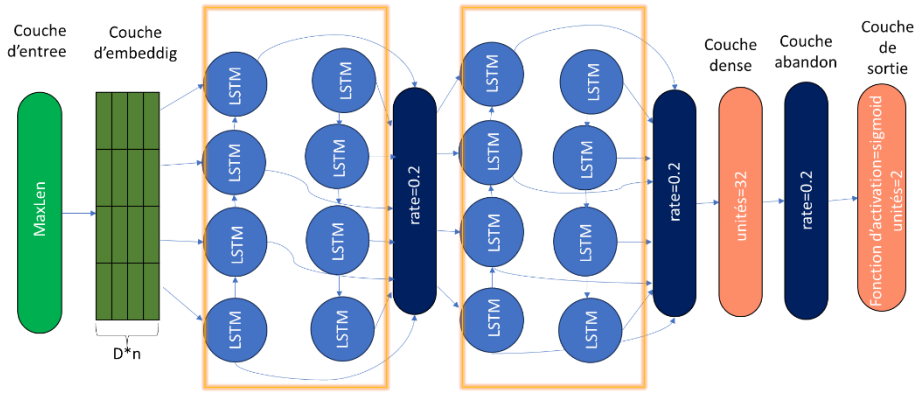

Figure 21 Architecture de BiLSTM.

II. 5. 4 Le modèle CNN-BiLSTM

L'architecture de notre modèle CNN-BiLSTM a été constituée de différentes couches pour la classification des données. Tout d'abord, nous avons inclus une couche d'entrée qui a spécifié la longueur maximale des séquences d'entrée. Ensuite, nous avons utilisé une couche d'embedding pour représenter les mots sous forme de vecteurs denses dans un espace continu dans le cas de la classification en utilisant des embeddings de mots.

Par la suite, une couche de convolution 1D avec 64 filtres a été ajoutée pour extraire les caractéristiques locales des données textuelles. Et une couche BiLSTM avec 32 unités ajoutée pour capturé les informations contextuelles dans les deux directions, ce qui a permis une meilleure compréhension des séquences de texte. Une couche de dropout a été appliquée pour désactiver 20% des neurones et prévenir le surapprentissage après les deux couches de BiLSTM et CNN. Une couche dense avec 32 unités a été ajoutée pour transformer les

caractéristiques extraites par la BiLSTM en un format approprié pour la classification. Pour réduire le surapprentissage, une dernière couche de dropout a été appliquée.

Enfin, une couche de sortie dense avec 2 neurones et une fonction d'activation sigmoid a été utilisée pour obtenir les prédictions de classification.

Le tableau 14 présente un aperçu des hyperparamètres pour chaque modèle, ainsi que pour toutes les techniques de représentation des données d'entrée. Tandis que la dimensionnalité de l'espace d'embedding est de 300 pour les modèles préentraînés araVec (CBOW et Skipgram) et de 768 pour AraBERT.

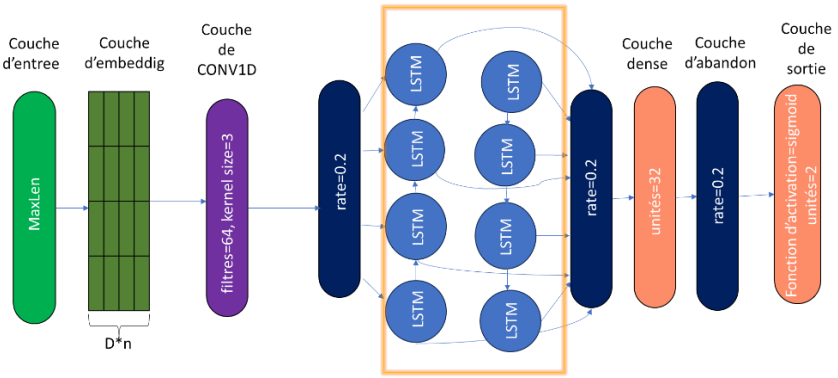

Figure 22 Architecture de CNN-BiLSTM.

	CNN	LSTM	BiLSTM	CNN-BiLSTM
Nombre de couches (conv1D,LSTM,BiLSTM)	2	2	2	2
Nombre de filtres de convolution (filters)	64-32	-	-	64
Nombre d'unités LSTM(units)	-	64-32	64-32	32
Taille de noyau de convolution (kernel_size)	3	-	-	3
Fonction d'activation de convolution (activation)	Relu	-	-	Relu
Taux d'abandon (rate)	0.2	0.2	0.2	0.2
Taille de la fenêtre de pooling (pool_size)	32	-	-	-
Nombre d'unités de la couche dense (units)	32	32	32	32
Fonction d'activation de la couche de sortie	sigmoid	sigmoid	sigmoid	sigmoid
Fonction d'optimisation	Adam	Adam	Adam	Adam
BATCH_SIZE	64	64	64	64

Tableau 13 Tableau récapitulatif des hyperparamètres des modèles.

II. 5. 5 Choix des hyperparamètres :

Pour déterminer les meilleurs hyperparamètres pour chaque modèle, plusieurs tests ont été réalisés, ces tests concernent le nombre optimal de couches (conv1D, LSTM ,BiLSTM), le

choix entre deux fonctions d'optimisation du gradient : la descente de gradient stochastique (SGD) et Adam et enfin le nombre optimal de batch size.

	Une seule couche	Deux couches
Exactitude (accuracy)%		
Convolution 1D	93.05	93.85
LSTM	93.18	93.29
BiLSTM	93.98	94.07
Précisions		
Convolution 1D	93.03	93.82
LSTM	93.24	93.33
BiLSTM	94.00	94.08
Rappel (recall)%		
Convolution 1D	93.04	93.89
LSTM	93.10	93.27
BiLSTM	93.95	94.03
F1 score%		
Convolution 1D	93.04	93.84
LSTM	93.17	93.29
BiLSTM	93.98	94.06

Tableau 14 Comparaison de la performance des modèles avec une et deux couches.

D'après le tableau 16 qui représente une comparaison de la performance des modèles avec une et deux couches, on remarque que la performance en utilisant deux couches est plus élevée à celles qui utilisent une seule couche, c'est pour cette raison qu'on a continué nos tests en utilisant deux couches.

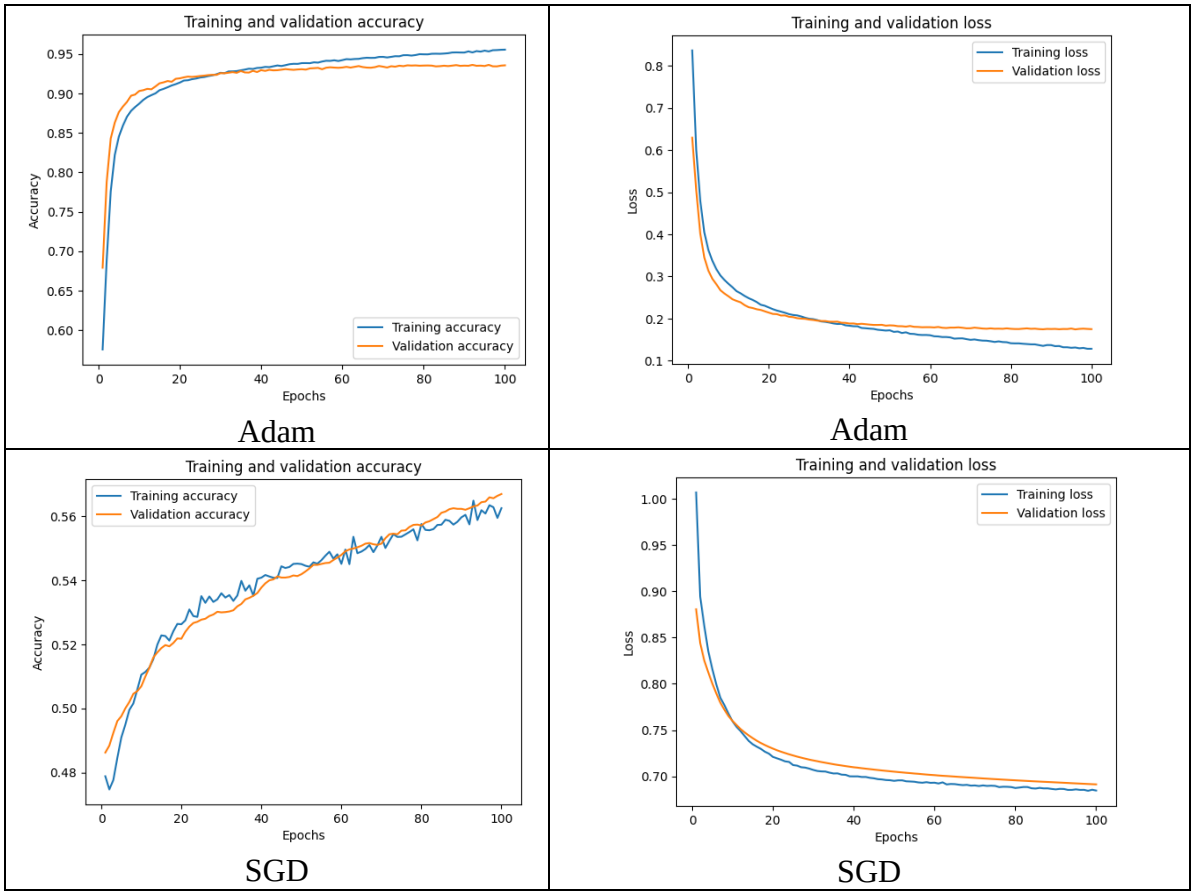

Figure 23 L'exactitude et la perte en utilisant la fonction d'optimisation Adam et SGD.

En ce qui concerne le choix de l'optimiseur, nous avons opté pour Adam, car il est plus rapide que la descente de gradient, comme on peut le voir sur la figure ci-dessus. En effet, en utilisant le même nombre d'époques, l'exactitude converge rapidement lorsque l'on utilise Adam, et la fonction de perte atteint rapidement son minimum. Nous avons donc choisi d'utiliser la fonction Adam comme optimiseur pour les tests qui suivent.

Finalement, on a essayé trois possibilités de batch size, on a commencé par 32, puis on a essayé avec 64. On a remarqué quand on utilise 64 le modèle converge rapidement par rapport à 32, ensuite on a essayé avec 128, mais malheureusement la mémoire disponible est insuffisante. Donc, on a choisi 64 comme un nombre optimal de batch size.

III Conclusion

En conclusion, ce chapitre a présenté l'architecture globale de notre système pour l'analyse des sentiments en langue arabe. Nous avons présenté toutes les étapes de prétraitement ainsi les modèles proposés pour la classification des sentiments et leur fonctionnement. Ce chapitre constitue une étape cruciale dans la réalisation de notre projet, car il permet de définir clairement l'architecture et les différentes composantes de notre système, ce qui nous permettra de passer à l'étape suivante, à savoir les tests et les résultats pour évaluer l'efficacité de notre approche.

Chapitre III Tests et résultats

I Introduction

La phase des tests et résultats est une étape essentielle dans tout projet de développement de système. Elle permet d'évaluer la performance et la qualité du système développé en utilisant des mesures et des métriques prédéfinies. Dans ce chapitre, nous allons détailler les tests réalisés sur notre système d'analyse des sentiments en utilisant différentes ressources matérielles et logicielles. Nous présenterons également les différents résultats obtenus lors des tests. Enfin, nous interpréterons ces résultats et discuterons de leurs implications pour l'amélioration du système.

II Environnement de travail

II. 1 Matériel

Pour accomplir notre travail, nous avons utilisé la configuration matérielle suivante :

	Processeur	**RAM**
Machine 01	Intel(R) Core (TM) i5-6300U CPU @ 2.40 GHz 2.50 GHz	8 Go
Machine 02	Intel(R) Core (TM) i5-7300U CPU @ 2.60 GHz 2.70 GHz	8 Go

Tableau 15 Les caractéristiques des matérielles utilisées

II. 2 Logiciels et bibliothèques

a) **Google Colab :** une plateforme cloud de développement et d'exécution de code Python avec des ressources matérielles telles que le processeur, la mémoire 12.7 GB de RAM et 78.2 GB de disque fournis gratuitement par Google.
b) **Jupyter Notebook :** un environnement de développement interactif pour la programmation en Python qui permet de créer et d'exécuter des notebooks contenant du code, des visualisations et des explications en texte.
c) **TensorFlow :** une bibliothèque open source de calcul numérique utilisée pour la création de modèles d'apprentissage automatique, y compris les réseaux de neurones profonds. Elle est développée par Google et permet de travailler sur des tâches telles que la classification, la régression, la reconnaissance d'image, la génération de texte,

d) etc. TensorFlow est utilisé pour la création de modèles d'apprentissage en utilisant des langages de programmation tels que Python, C++, etc.[4]

e) Keras : une interface de haut niveau pour la création de modèles de machine learning, basée sur TensorFlow.

f) NLTK (Natural Language Toolkit) : est une bibliothèque Python open-source qui permet de traiter et d'analyser des données textuelles en langage naturel. Elle propose des fonctionnalités pour la tokenisation, la lemmatisation, la suppression des stopwords, la reconnaissance d'entités nommées, la classification de texte, la création de modèles de langage, et bien plus encore. NLTK est une bibliothèque populaire utilisée par les chercheurs en traitement automatique du langage naturel et les développeurs pour l'analyse de texte et la création de chatbots, entre autres applications.

g) Pandas : est une bibliothèque open source pour Python qui permet la manipulation et l'analyse de données. Elle est très utile pour le nettoyage, la transformation et l'analyse des données avant l'entraînement des modèles de machine learning.

h) NumPy : une bibliothèque de calculs numériques en Python, utilisée pour effectuer des opérations mathématiques sur des tableaux et des matrices.

i) Matplotlib : une bibliothèque de visualisation de données en Python, utilisée pour créer des graphiques et des figures[5].

j) Gensim : est une bibliothèque open-source pour la modélisation de sujets et la similarité sémantique de documents en utilisant des algorithmes d'apprentissage automatique. Elle permet d'effectuer des tâches telles que la construction de modèles Word2Vec pour la représentation vectorielle de mots.

III Description du dataset

Le choix d'un dataset approprié est important pour la réussite de toute tache de traitement du langage naturel. Dans notre travail, nous avons sélectionné un dataset en langue arabe équilibré qui contient une grande quantité de données pertinentes et variées. Ce dataset a été collecté à partir du site 'Booking.com' durant les mois de juin et juillet 2016, qui est l'une des plus grandes plateformes de réservation d'hôtels dans le monde, il est disponible sur GitHub [6].

Le dataset HARD utilisé dans cette étude est constitué de 105 698 avis d'hôtels, divisés en deux catégories positives et négatives. Le dataset contient 2 532 758 mots en total et de 196838 mots unique. Les avis ont été rédigés en arabe standard moderne ainsi qu'en arabe dialectal. Le dataset est équilibré, c'est-à-dire que chaque catégorie contient approximativement le même nombre d'avis. Chaque avis a été annoté manuellement par des experts en linguistique

[4] « TensorFlow ». https://www.tensorflow.org/ (consulté le 2 mai 2023).

[5] « Matplotlib — Visualization with Python ». https://matplotlib.org/ (consulté le 7 juin 2023)

[6] « elnagara/HARD-Arabic-Dataset: Hotels Arabic-Reviews Dataset ». https://github.com/elnagara/HARD-Arabic-Dataset (consulté le 2 mai 2023).

pour l'analyse des sentiments, en utilisant une échelle de notation de 4 à 5 pour les avis positifs et de 1 à 3 pour les avis négatifs.

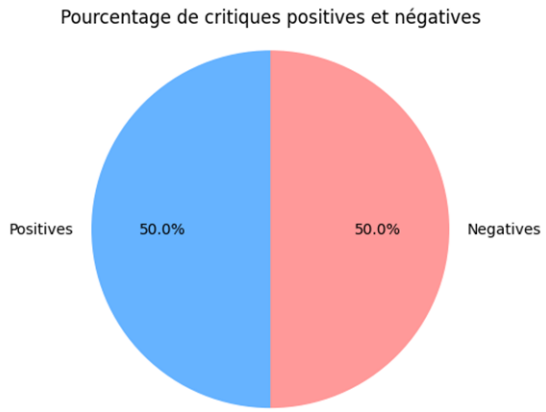

Figure 24 Le pourcentage de critiques positives et négatives dans le dataset.

Le dataset originale, est composé de huit colonnes présentées comme suit :

	no	Hotel name	rating	user type	room type	nights	review
0	2	فندق 72	5	مسافر منفرد	غرفة ديلوكس مزدوجة أو توأم	أقمت ليلة واحدة	."ممتاز". النظافة والطاقم متعاون"
1	3	فندق 72	5	زوج	غرفة ديلوكس مزدوجة أو توأم	أقمت ليلة واحدة	...استثنائي. سهولة إنهاء المعاملة في الاستقبال. ل
2	16	فندق 72	5	زوج	-	أقمت ليلتين	...استثنائي. انصح بأختيار الاسويت و بالاخص غرفه ر
3	20	فندق 72	1	زوج	غرفة قياسية مزدوجة	أقمت ليلة واحدة	... استغرب تقييم الفندق كخمس نجوم". لا شي. يستحق"
4	23	فندق 72	4	زوج	غرفة ديلوكس مزدوجة أو توأم	أقمت ليلتين	...جيد. المكان جميل و هادىء. كل شي جيد ونظيف بس كا

Figure 25 Les cinq premières lignes du dataset original.

Dans cette étude, nous avons utilisé un dataset initial composé de 8 colonnes pour entraîner et tester nos modèles de classification des sentiments. Nous avons ensuite appliqué les étapes de prétraitement décrites dans le chapitre précédent, sauf l'étape de stemming, ainsi que la suppression des colonnes inutiles et l'ajout d'une colonne label pour spécifier le sentiment du texte en fonction de la colonne "rating comme illustré dans la figure 22.

	text	label	cleaned_text
0	."ممتاز". النظافة والطاقم متعاون"	Positive	ممتاز النظافه الطاقم متعاون
1	...استثنائي. سهولة إنهاء المعاملة في الاستقبال. ل	Positive	استثناءي سهوله انهاء المعامله الاستقبال لاشيء
2	...استثنائي. انصح بأختيار الاسويت و بالاخص غرفه ر	Positive	... استثناءي انصح باختيار الاسويت بالاخص غرفه رقم
3	... استغرب تقييم الفندق كخمس نجوم". لا شي. يستحق"	Negative	استغرب تقيم الفندق كخمس نجوم لا شي يستحق نجمه
4	...جيد. المكان جميل و هادىء. كل شي جيد ونظيف بس كا	Positive	...جيد المكان جميل و هادىء شي جيد ونظيف حوض السباح

Figure 26 Les cinq premières lignes du dataset après le prétraitement et sans stemming.

Dans le cadre de notre étude, nous avons évalué l'impact du stemming sur la performance de nos modèles. Nous avons appliqué deux types de stemmers (ISRI et Tashaphyne) au dataset prétraité pour créer deux nouveaux ensembles de données qui ont été

utilisés ensuite pour tester la performance de nos modèles après l'application du stemming, comme le montrent les figures 24 et 25. Nous allons comparer les performances de ces ensembles afin d'évaluer l'effet du stemming sur la classification des sentiments.

	text	label	cleaned_text
0	."ممتاز". النظافة والطاقم متعاون"	Positive	متز نظف طقّم تعا
1	...استثنائي. سهولة إنهاء المعاملة في الاستقبال. ل	Positive	استثناءي سهل هاء عمل قبل اشء
2	...استثنائي. انصح بأختيار الاسويت و بالاخص غرفه ر	Positive	استثناءي نصح خير اسي اخص غرف رقم نوع ارض
3	... استغرب تقييم الفندق كخمس نجوم". لا شي. يستحق"	Negative	غرب تقّم ندق كخمس نجم لا شي سحق نجم
4	...جيد. المكان جميل وهاديء. كل شي جيد ونظيف بس كا	Positive	...جيد كان جمل هاديء شي جيد نظف حوض سبح عمل فتر ك

Figure 27 Les cinq premières lignes du dataset après le prétraitement et avec le stemmer ISRI.

	text	label	cleaned_text
0	."ممتاز". النظافة والطاقم متعاون"	Positive	متز ظف طقّم عا
1	...استثنائي. سهولة إنهاء المعاملة في الاستقبال. ل	Positive	استثناء سهل هاء عمل قبل شء
2	...استثنائي. انصح بأختيار الاسويت و بالاخص غرفه ر	Positive	استثناء صح خير سي خص غرف رقم وع رض
3	... استغرب تقييم الفندق كخمس نجوم". لا شي. يستحق"	Negative	غرب قم دق خمس جم لا شي سحق جم
4	...جيد. المكان جميل وهاديء. كل شي جيد ونظيف بس كا	Positive	... جيد كان جمل هاديء شي جيد ظف حوض سبح عمل تر لم

Figure 28 Les cinq premières lignes du dataset après le prétraitement et avec le stemmer Tashaphyne.

IV Expérimentations et analyses des performances

IV. 1 Expérimentations des modèles proposés sur le dataset HARD :

Trois expériences ont été menées sur l'ensemble de test pour évaluer la performance de différents modèles dans le contexte de l'analyse des sentiments. La première expérience consistait à évaluer la performance des modèles CNN, LSTM, BiLSTM et CNN-BiLSTM en utilisant ou non le stemming, pour chaque technique de représentation des données. La deuxième expérience a évalué la performance de chaque méthode de pondération de données, telles que bag of words, TF-IDF, AraVec et AraBERT, en utilisant les modèles déjà mentionnés. Enfin, la troisième expérience a comparé les performances des modèles pour chaque méthode de représentation de données utilisée. Ces expériences ont permis de déterminer les performances les plus élevées pour chaque modèle et chaque méthode de représentation des données, ainsi que de mettre en évidence les avantages et les inconvénients de chaque approche.

IV. 1. 1 Impact du stemming

Afin d'évaluer l'impact de l'utilisation du stemming sur les performances de nos modèles, nous avons réalisé plusieurs tests dont les résultats sont présentés dans les tableaux suivants :

Sans stemming	CNN					LSTM					BiLSTM					CNN-BiLSTM				
	Bag of Words	Tf-idf	AraVec	Skip Gram	AraBERT	Bag of Words	Tf-idf	CBOW	Skip Gram	AraBERT	Bag of Words	Tf-idf	CBOW	Skip Gram	AraBERT	Bag of Words	Tf-idf	CBOW	Skip Gram	AraBERT
Exactitude%	74.63	74.10	93.59	94.21	95.32	74.40	73.92	92.98	81.67	95.29	74.08	74.27	93.91	93.73	95.20	74.34	74.38	93.65	94.10	95.30
Précision%	76.00	77.74	93.59	94.14	95.40	79.15	76.75	93.00	81.76	95.31	78.00	80.14	93.91	93.76	95.15	78.20	76.78	93.68	94.13	95.14
Rappel %	71.99	77.74	93.49	93.49	95.23	66.26	76.75	92.91	92.91	95.23	67.09	80.14	93.90	93.90	95.23	67.49	76.78	93.63	93.63	95.40
F1 score%	73.94	72.28	93.58	94.20	95.31	72.13	72.47	92.97	81.42	95.28	72.13	71.50	93.90	93.72	95.20	72.45	73.19	93.65	94.09	95.29

Tableau 16 Évaluation de la performance des modèles de deep learning sans stemming.

ISRI	CNN					LSTM					BiLSTM					CNN-BiLSTM				
	Bag of Words	Tf-idf	CBOW	Skip Gram	AraBERT	Bag of Words	Tf-idf	CBOW	Skip Gram	AraBERT	Bag of Words	Tf-idf	CBOW	Skip Gram	AraBERT	Bag of Words	Tf-idf	CBOW	Skip Gram	AraBERT
Exactitude%	81.23	81.54	93.85	94.38	95.02	80.92	80.81	93.29	93.76	95.02	80.11	81.01	94.07	94.00	95.16	80.43	81.12	94.19	94.31	95.01
Précision%	83.02	84.17	93.82	94.23	95.01	82.92	81.03	93.33	93.71	95.05	82.08	83.18	94.08	94.02	95.14	82.71	82.35	94.17	94.28	94.96
Rappel %	78.54	84.17	93.89	93.89	94.99	78.80	81.03	93.27	93.27	95.03	77.04	83.18	94.03	93.97	95.20	76.95	82.35	94.20	94.20	94.96
F1 score%	80.71	80.80	93.84	94.27	95.02	80.51	80.74	93.29	93.75	95.02	79.48	79.55	94.06	94.00	95.15	79.72	80.75	94.19	94.53	95.01

Tableau 17 Évaluation de la performance des modèles de deep learning en utilisant ISRI stemmer.

Tashaphyne	CNN					LSTM					BiLSTM					CNN-BiLSTM				
	Bag of Words	Tf-idf	AraVec	Skip Gram	AraBERT	Bag of Words	Tf-idf	CBOW	Skip Gram	AraBERT	Bag of Words	Tf-idf	CBOW	Skip Gram	AraBERT	Bag of Words	Tf-idf	CBOW	Skip Gram	AraBERT
Exactitude%	83.06	82.50	94.15	94.49	94.84	82.00	82.32	92.76	91.04	95.09	81.03	82.32	93.87	93.72	95.15	81.96	81.77	93.87	94.27	95.16
Précision%	86.65	84.20	94.19	94.55	94.88	83.05	86.56	92.64	91.12	95.13	83.23	82.32	93.82	93.70	95.15	83.95	79.51	93.82	94.28	95.17
Rappel %	78.16	80.01	94.10	94.46	94.82	80.41	76.53	92.92	90.96	95.09	77.77	80.92	93.86	93.77	95.14	79.03	85.61	93.86	94.20	95.17
F1 score%	82.19	82.06	94.15	94.49	94.84	80.71	81.24	92.76	91.03	95.08	80.38	82.07	93.86	93.71	95.14	81.42	82.45	93.86	94.26	95.15

Tableau 18 Évaluation de la performance des modèles de deep learning en utilisant Tashaphyne stemmer.

En analysant les résultats des trois tableaux (16, 17 et 18), les observations suivantes peuvent être faites :

- Dans le tableau 17, les résultats dépassent ceux du tableau 16 pour tous les modèles et toutes les métriques de performance, à l'exception de la technique de représentation des données AraBERT combinée avec tous les modèles, où les résultats du tableau 16 surpassent ceux du tableau 17.
- De même, dans le tableau 18, les résultats dépassent également ceux du tableau 16 pour tous les modèles et toutes les métriques de performance, sauf pour la technique de représentation des données AraBERT combinée avec tous les modèles, où les résultats du tableau 16 surpassent ceux du tableau 18.
- En comparant les résultats du tableau 17 en utilisant le stemmer ISRI avec les résultats du tableau 18 en utilisant le stemmer Tashaphyne, on observe une variation des résultats. Parfois, les résultats sont bons en utilisant Tashaphyne, tandis que d'autres fois, ils sont bons en utilisant ISRI. Par exemple, en prenant la métrique de performance l'exactitude pour faire une comparaison, on constate que :

- Pour le modèle CNN, Tashaphyne donne de meilleurs résultats pour toutes les techniques de représentation des mots, sauf pour AraBERT, avec une différence ne dépassant pas 2%.
- Pour les trois modèles LSTM, BiLSTM et CNN-BiLSTM, combinés avec la technique de représentation AraVec, c'est ISRI qui nous donne de meilleurs résultats. Cependant, pour les autres techniques telles que Bag of Words, Tf-idf et AraBERT, le modèle utilisant le stemmer Tashaphyne a donné de meilleurs résultats, à l'exception du cas de BiLSTM avec AraBERT, où la meilleure performance a été obtenue en utilisant le stemmer ISRI.

Par conséquent, nous pouvons conclure que, pour le dataset HARD, l'utilisation d'AraBERT sans l'étape de stemming lors du traitement des données donnent de bons résultats. En revanche, lorsque nous utilisons les techniques de représentation des mots Tf-idf et Bag of Words, ainsi que les techniques d'embedding des mots CBOW et Skip-gram, l'utilisation du stemmer joue un rôle important dans les performances du modèle.

IV. 1. 2 Impact des techniques de représentation des mots

Maintenant, regardons les résultats sous un angle différent, en nous concentrant sur les différentes techniques de représentation des mots. Les résultats sont présentés dans les tableaux suivants :

Bag of Words	CNN			LSTM			BILSTM			CNN-BILSTM		
	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne
Exactitude(accuracy)%	74.63	81.23	83.06	74.40	80.92	82.00	74.08	80.11	81.03	74.34	80.43	81.96
Précision%	76.00	83.02	86.65	79.15	82.92	83.05	78.00	82.08	83.23	78.20	82.71	83.95
Rappel (recall)%	71.99	78.54	78.16	66.26	78.80	80.41	67.09	77.04	77.77	67.49	76.95	79.03
F1 score%	73.94	80.71	82.19	72.13	80.51	80.71	72.13	79.48	80.38	72.45	79.72	81.42

Tableau 19 Performances des modèles de Deep Learning en utilisant bag of words avec et sans stemming.

Tf-idf	CNN			LSTM			BILSTM			CNN-BILSTM		
	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne
Exactitude(accuracy)%	74.10	81.54	82.50	73.92	80.81	82.32	74.27	81.01	82.32	74.38	81.12	81.77
Précision%	77.74	84.17	84.20	76.75	81.03	86.56	80.14	83.18	82.32	76.78	82.35	79.51
Rappel (recall)%	77.74	84.17	80.01	76.75	81.03	76.53	80.14	83.18	80.92	76.78	82.35	85.61
F1 score%	72.28	80.80	82.06	72.47	80.74	81.24	71.50	79.55	82.07	73.19	80.75	82.45

Tableau 20 Performances des modèles de Deep Learning en utilisant TF-idf avec et sans stemming.

AraVec(CBOW)	CNN			LSTM			BILSTM			CNN-BILSTM		
	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne
Exactitude(accuracy)%	93.59	93.85	94.15	92.98	93.29	92.76	93.91	94.07	93.87	93.65	94.19	94.09
Précision%	93.59	93.82	94.19	93.00	93.33	92.64	93.91	94.08	93.82	93.68	94.17	94.06
Rappel (recall)%	93.49	93.89	94.10	92.91	93.27	92.92	93.90	94.03	93.86	93.63	94.20	94.09
F1 score%	93.58	93.84	94.15	92.97	93.29	92.76	93.90	94.06	93.86	93.65	94.19	94.08

Tableau 21 Performances des modèles de Deep Learning en utilisant AraVec(cbow) avec et sans stemming.

AraVec(Skip Gram)	CNN			LSTM			BILSTM			CNN-BILSTM		
	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne
Exactitude(accuracy)%	94.21	94.38	94.49	81.67	93.76	91.04	93.73	94.00	93.72	94.10	94.31	94.27
Précision%	94.14	94.23	94.55	81.76	93.71	91.12	93.76	94.02	93.70	94.13	94.28	94.28
Rappel (recall)%	93.49	93.89	94.46	92.91	93.27	90.96	93.90	93.97	93.77	93.63	94.20	94.20
F1 score%	94.20	94.27	94.49	81.42	93.75	91.03	93.72	94.00	93.71	94.09	94.53	94.26

Tableau 22 Performances des modèles de Deep Learning en utilisant AraVec (skipgram) avec et sans stemming.

AraBert	CNN			LSTM			BILSTM			CNN-BILSTM		
	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne
Exactitude(accuracy)%	95.32	95.02	94.84	95.29	95.02	95.09	95.20	95.16	95.15	95.30	95.01	95.16
Précision%	95.40	95.01	94.88	95.31	95.05	95.13	95.15	95.14	95.15	95.14	94.96	95.17
Rappel (recall)%	95.23	94.99	94.82	95.23	95.03	95.09	95.23	95.20	95.14	95.40	94.96	95.17
F1 score%	95.31	95.02	94.84	95.28	95.02	95.08	95.20	95.15	95.14	95.29	95.01	95.15

Tableau 23 Performances des modèles de Deep Learning en utilisant AraBert avec et sans stemming.

En analysant attentivement les résultats présentés dans les tableaux (19, 20, 21, 22, 23), nous pouvons tirer plusieurs observations significatives. Tout d'abord, en ce qui concerne les techniques traditionnelles Tf-idf et Bag of Words elles ne montrent pas de grande différence de performance, avec une variation générale inférieure à 1%. De même, les deux approches de Word2Vec (CBOW et Skip-gram) présentent des performances comparables.

Cependant, la véritable différence de performance se manifeste lorsque nous comparons les techniques de word embedding, comme présenté dans les tableaux (21, 22, 23), avec les approches traditionnelles telles que Bag of Words et Tf-idf, représentés dans les tableaux (22, 23), ou les résultats en utilisant les embedding surpassent généralement les méthodes traditionnelles telles que bag of words et TF-IDF. En particulier, les résultats obtenus avec

l'embedding des mots AraBERT representes dans le tableau 23 surpassent de manière significative toutes les autres techniques de représentation étudiées.

En résumé, ces résultats soulignent l'importance de l'utilisation de techniques de word embedding avancées, telles que AraBERT, pour obtenir des performances supérieures dans la détection d'opinion.

IV. 1. 3 Impact des modèles de Deep Learning

Le dernier aspect a examiné concerne les modèles de deep learnng, dont les résultats sont présentés dans les tableaux :

CNN	Bag of Words			Tf-idf			CBOW			Skip Gram			AraBert		
	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne
Exactitude(accuracy)%	**74.63**	**81.23**	**83.06**	74.10	**81.54**	**82.50**	93.59	93.85	**94.15**	**94.21**	**94.38**	**94.49**	**95.32**	95.02	94.84
Précision%	76.00	83.02	86.65	77.74	84.17	84.20	93.59	93.82	94.19	94.14	94.23	94.55	95.40	95.01	94.88
Rappel (recall)%	71.99	78.54	78.16	77.74	84.17	80.01	93.49	93.89	94.10	93.49	93.89	94.46	95.23	94.99	94.82
F1 score%	73.94	80.71	82.19	72.28	80.80	82.06	93.58	93.84	94.15	94.20	94.27	94.49	95.31	95.02	94.84

Tableau 24 Performances de modèle CNN en utilisant toutes les techniques de représentation des données.

LSTM	Bag of Words			Tf-idf			CBOW			Skip Gram			AraBert		
	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne
Exactitude(accuracy)%	74.40	80.92	82.00	73.92	80.81	82.32	92.98	93.29	92.76	81.67	93.76	91.04	95.29	95.02	95.09
Précision%	79.15	82.92	83.05	76.75	81.03	86.56	93.00	93.33	92.64	81.76	93.71	91.12	95.31	95.05	95.13
Rappel (recall)%	66.26	78.80	80.41	76.75	81.03	76.53	92.91	93.27	92.92	92.91	93.27	90.96	95.23	95.03	95.09
F1 score%	72.13	80.51	80.71	72.47	80.74	81.24	92.97	93.29	92.76	81.42	93.75	91.03	95.28	95.02	95.08

Tableau 25 Performances de modèle LSTM en utilisant toutes les techniques de représentation des données.

BiLSTM	Bag of Words			Tf-idf			CBOW			Skip Gram			AraBert		
	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne
Exactitude(accuracy)%	74.08	80.11	81.03	74.27	81.01	82.32	**93.91**	94.07	93.87	93.73	94.00	93.72	95.20	**95.16**	95.15
Précision%	78.00	82.08	83.23	80.14	83.18	82.32	93.91	94.08	93.82	93.76	94.02	93.70	95.15	95.14	95.15
Rappel (recall)%	67.09	77.04	77.77	80.14	83.18	80.92	93.90	94.03	93.86	93.90	93.97	93.77	95.23	95.20	95.14
F1 score%	72.13	79.48	80.38	71.50	79.55	82.07	93.90	94.06	93.86	93.72	94.00	93.71	95.20	95.15	95.14

Tableau 26 Performances de modèle BiLSTM en utilisant toutes les techniques de représentation des données.

CNN-BiLSTM	Bag of Words			Tf-idf			CBOW			Skip Gram			AraBert		
	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne	Sans stem	ISRI	Tashaphyne
Exactitude(accuracy)%	74.34	80.43	81.96	**74.38**	81.12	81.77	93.65	**94.19**	94.09	94.10	94.31	94.27	95.30	95.01	**95.16**
Précision%	78.20	82.71	83.95	76.78	82.35	79.51	93.68	94.17	94.06	94.13	94.28	94.28	95.14	94.96	95.17
Rappel (recall)%	67.49	76.95	79.03	76.78	82.35	85.61	93.63	94.20	94.09	93.63	94.20	94.20	95.40	94.96	95.17
F1 score%	72.45	79.72	81.42	73.19	80.75	82.45	93.65	94.19	94.08	94.09	94.53	94.26	95.29	95.01	95.15

Tableau 27 Performances de modèle CNN-BiLSTM en utilisant toutes les techniques de représentation des données.

En observant les tableaux (24,25,26,27) on remarque qu'il n'y a pas de modèle unique qui donne les meilleurs résultats mais plutôt tous les modèles donnent des résultats presque similaires pour chaque technique de représentation des données. En se basant sur la métrique de l'exactitude, on peut noter ce qui suit :

- Le modèle CNN obtient de meilleures performances pour les techniques traditionnelles telles que le sac de mots (bag of words) et le TF-IDF, par rapport aux autres modèles. Cependant, pour la technique du TF-IDF sans stemming, la combinaison de CNN-BiLSTM affiche une légère amélioration par rapport au CNN. Ces observations sont particulièrement visibles dans les tableaux 24 et 27.
- Pour la technique de représentation des mots Skip-gram, le modèle CNN donne également de meilleurs résultats.
- Pour la technique de représentation des mots CBOW, le CNN donne de meilleurs résultats lorsque le stemmer utilisé est Tashaphyne, tandis que le BiLSTM donne de meilleurs résultats sans l'utilisation de stemming. Les résultats démontrent que l'utilisation du stemmer ISRI avec CNN-BiLSTM conduit à de meilleures performances.
- En ce qui concerne la technique de représentation des mots AraBERT, on observe que le BiLSTM affiche les meilleures performances lorsqu'il est associé au stemmer ISRI, tandis que le CNN-BiLSTM donne de meilleurs résultats avec le stemmer Tashaphyne. Cependant, sans utiliser de stemming, le modèle CNN présente une légère amélioration de 0.02% par rapport au CNN-BiLSTM, avec une exactitude de 95,32 %.

Finalement, d'après les analyses effectuées sur les tableaux (19, 20, 21, 22, 23) ainsi que sur les tableaux (17, 18, 19) ou on a conclu que AraBERT présente de bonne performance sans nécessiter l'utilisation du stemming sur l'ensemble de test HARD. Étant donné que les performances en termes d'exactitude des différents modèles sont presque similaires, avec une différence de moins de 0.1% lorsqu'on utilise la technique de représentation des mots AraBERT sans stemming, il devient un peu difficile de choisir le meilleur modèle. Pour prendre cette décision, on compare le taux d'erreur de chaque modèle. Les résultats sont présentés dans le tableau suivant :

MODELE	TAUX D'ERREUR
CNN	0.1395
LSTM	0.1507
BILSTM	0.1491
CNN-BILSTM	0.1437

Tableau 28 Le taux d'erreur dans les quatres modèles.

En analysant le tableau 28, on remarque que le taux d'erreur de CNN est le plus faible. Par conséquent, le meilleur modèle est le CNN avec une exactitude de 95.32%.

IV. 2 Expérimentation sur d'autres dataset

IV. 2. 1 Efficacité de notre modèle

Dans cette sous-section, nous évaluons l'efficacité des hyperparamètres de notre meilleur modèle CNN en utilisant la représentation des embeddings AraBERT sur un autre dataset appelé LABR. Ce dataset a été utilisé par Elfaik et Nfaoui [21], il est composé de plus de 63 000 critiques de livres en arabe. Ces critiques ont été collectées à partir d'un site web spécifique durant le mois de mars 2013, et elles ont été réparties en deux classes :8244 critiques pour la classe positive et 8244 critiques pour la classe négative.

Nous avons suivi le même fractionnement des données, en utilisant 80% pour l'entraînement et 20% pour le test. Il est important de mentionner que dans leur travail, ils ont exprimé leur intention d'utiliser AraBERT comme prochaine étape. Par conséquent, notre évaluation comparative vise à vérifier si l'utilisation d'AraBERT améliore la performance par rapport à leur approche précédente. Les résultats obtenus sont présentés dans le tableau suivant :

	BiLSTM[21]	Notre modèle (CNN)
Exactitude(accuracy)%	80.70	82.58
Précision%	79.79	82.85
Rappel (recall)%	80.04	81.64
F1 score%	79.41	82.52

Tableau 29 Comparaison entre la performance de modèle BiLSTM[21] et notre modèle.

Les résultats démontrent que notre modèle obtient de bons résultats pour toutes les métriques de performance par rapport au modèle proposé par Elfaik et Nfaoui. Cette amélioration est attribuée à la capacité de notre modèle à utiliser l'embedding Arabert.

Néanmoins il reste en dessous des résultats obtenus sur le dataset HARD. Cela est dû probablement au fait que le choix des hyperparamètres du modèle est spécifique pour ce dataset.

IV. 2. 2 Validation des résultats expérimentaux

Afin de valider les résultats obtenus visant à mesurer l'impact des techniques de représentation de données et des modèles de deep learning, nous avons choisi d'utiliser les modèles CNN, LSTM, BiLSTM et CNN-BiLSTM avec les techniques de représentation de données AraVec (cbow et skipgram) et Arabert, car ces modèles ont donné les meilleurs résultats par rapport aux techniques de représentation traditionnelles. Nous avons testé ces modèles sur trois datasets différents, varient en taille et en domaines (hôtels, politique, etc.), et qui représentent différents dialectes de l'arabe :

- Arabic Company Reviews[7] : sont une collection de plus de 40 046 avis en arabe collectés à des fins d'analyse de sentiments visant à produire une évaluation pour un restaurant. Ces données sont particulièrement utiles pour les entreprises souhaitant évaluer leur réputation en ligne et améliorer leur service client en utilisant des analyses de sentiments pour comprendre les opinions de leurs clients.
- ASTD[8] : est un ensemble de données compose d'environ 9 694 tweets, classés en quatre catégories subjectives positives, subjectives négatives, subjectives mixtes et objectives. Varies de différents domaines de politique, des avis sur des sujets pertinents etc.
- Arabic 100k Reviews [9] : un ensemble de données compose de 99k tweets en langue rabe, combine des avis d'hôtels, de livres, de films, de produits et de quelques compagnies aériennes. Il comporte trois classes (Mixte, Négatif et Positif). Les textes (avis) ont été nettoyés en supprimant les diacritiques arabes et les caractères non arabes. L'ensemble de données ne contient pas d'avis en double.

Dataset	Nombre de tweets avant le prétraitement	Nombre de tweets après le prétraitement	Nombre de classes avant le prétraitement	Nombre de classes après le prétraitement	Nombre des mots uniques	Nature de la langue
Arabic Company Reviews	40 046 avis	28 400 avis	3	2	32 523	MSA Et dielectale
ASTD	9 694 tweets	1 554 tweets	4	2	10 030	MSA Et dielectale
Arabic 100k Reviews	99 000 tweets	66 000tweets	3	2	213 089	MSA Et dielectale

Tableau 30 Un résumé sur les ensembles de données utilisées pour le test.

[7] « Arabic Company Reviews (عربي) | Kaggle ». https://www.kaggle.com/datasets/fahdseddik/arabic-company-reviews (consulté le 2 mai 2023).

[8] « ASTD/README.md~ at master · mahmoudnabil/ASTD ». https://github.com/mahmoudnabil/ASTD (consulté le 2 mai 2023).

[9]« Arabic 100k Reviews | Kaggle ». https://www.kaggle.com/datasets/abedkhooli/arabic-100k-reviews (consulté le 2 mai 2023).

Dans le tableau 25, nous présentons les résultats des tests réalisés sur différents modèles de réseaux de neurones appliqués aux ensembles de données Arabic Company Reviews, ASTD et Arabic 100k Reviews, afin d'évaluer leurs performances respectives.

Exactitude (accuracy)%												
	Arabic Company Reviews				ASTD				Arabic 100k Reviews			
	CNN	**LSTM**	**BILSTM**	**CNN-BILSTM**	**CNN**	**LSTM**	**BILSTM**	**CNN-BILSTM**	**CNN**	**LSTM**	**BILSTM**	**CNN-BILSTM**
Aravec(cbow)	78.38	74.83	77.50	**78.53**	61.52	59.78	58.94	**61.65**	76.90	76.00	76.90	**76.94**
Aravec (Skip-Gram)	79.35	78.36	79.30	**81.32**	63.64	62.29	**63.96**	62.93	**77.84**	76.65	77.58	77.75
Arabert	83.48	83.13	**84.35**	83.75	78.51	79.60	**80.37**	79.47	81.00	81.13	**82.11**	81.35
Précision%												
	Arabic Company Reviews				ASTD				Arabic 100k Reviews			
	CNN	**LSTM**	**BILSTM**	**CNN-BILSTM**	**CNN**	**LSTM**	**BILSTM**	**CNN-BILSTM**	**CNN**	**LSTM**	**BILSTM**	**CNN-BILSTM**
Aravec(cbow)	78.03	74.82	77.80	**78.52**	61.42	59.90	58.92	**61.68**	76.78	76.00	**76.95**	76.89
Aravec (Skip-Gram)	79.38	78.30	79.20	**81.25**	63.41	62.34	**63.81**	62.88	**77.77**	76.59	77.55	**77.77**
Arabert	83.89	83.31	**84.48**	83.53	79.04	79.94	**80.35**	79.43	81.36	81.25	**82.09**	81.18
Rappel (recall)%												
	Arabic Company Reviews				ASTD				Arabic 100k Reviews			
	CNN	**LSTM**	**BILSTM**	**CNN-BILSTM**	**CNN**	**LSTM**	**BILSTM**	**CNN-BILSTM**	**CNN**	**LSTM**	**BILSTM**	**CNN-BILSTM**
Aravec(cbow)	**78.98**	74.82	75.89	78.58	**62.48**	59.78	57.79	61.84	**77.15**	75.96	76.84	77.01
Aravec (Skip-Gram)	79.54	78.45	79.56	**81.38**	63.90	62.42	**64.09**	63.00	**77.98**	76.74	77.60	77.84
Arabert	82.92	82.89	**84.33**	84.19	77.67	79.21	**80.76**	80.24	80.38	80.93	**82.16**	81.61
F1 score%												
	Arabic Company Reviews				ASTD				Arabic 100k Reviews			
	CNN	**LSTM**	**BILSTM**	**CNN-BILSTM**	**CNN**	**LSTM**	**BILSTM**	**CNN-BILSTM**	**CNN**	**LSTM**	**BILSTM**	**CNN-BILSTM**
Aravec(cbow)	78.31	74.80	77.48	**78.48**	61.51	59.63	58.94	**61.64**	76.89	75.98	76.89	**76.93**
AraVec (Skip-Gram)	79.34	77.92	79.29	**81.25**	63.58	61.93	**63.88**	62.92	**77.82**	76.63	77.57	77.74
Arabert	83.48	83.06	**84.30**	83.60	78.41	79.49	**80.33**	79.47	80.99	81.06	**82.10**	81.33

Tableau 31 Les résultats de test de différents modèles de réseaux de neurones sur les dataset Arabic Company Reviews, ASTD et Arabic 100k Reviews.

En observant les résultats des modèles testés sur les ensembles de données "Arabic Company Reviews", "ASTD" et "Arabic 100k Reviews" présentés dans le tableau 30, on constate que les tests effectués sur les trois ensembles de données donnent des résultats acceptables en général, cependant le test sur le dataset "Arabic Company Reviews" nous donne meilleurs performance, suivi par le dataset "Arabic 100k Reviews", puis l'ensemble "ASTD"

Le modèle BiLSTM combiné à la méthode de représentation des données AraBERT offre généralement les meilleurs résultats en termes d'exactitude, de précision, de rappel et de F1-score. Les scores obtenus dépassent 84% pour l'ensemble "Arabic Company Reviews", dépassent 80% pour l'ensemble "ASTD" et dépassent 82% pour l'ensemble"Arabic 100k Reviews", pour toutes les métriques.

V Interprétation des résultats

Nous avons effectué des tests initiaux sur différents modèles de Deep Learning en utilisant plusieurs techniques de représentation des mots, avec et sans stemming, afin d'évaluer l'impact de cette technique de prétraitement du texte sur les performances des modèles. Nous avons observé que l'utilisation du stemming améliore la performance des modèles TF-IDF, Bag of Words et Aravec. Cependant, pour le modèle réentraîné Arabert, l'utilisation du stemming diminue la performance des modèles. Cette différence peut s'expliquer par le fait que le stemming réduit les mots à leur forme racine, regroupant ainsi différentes variations d'un même mot. Cette approche peut être bénéfique pour les modèles tels que TF-IDF, Bag of Words et Aravec, qui ne sont pas capables de prendre en compte les variations de mots. Cependant, les embeddings d'Arabert sont déjà contextualisés et riches en informations, ce qui les rend plus adaptés pour représenter les mots tels qu'ils apparaissent dans notre dataset, sans avoir besoin de les réduire à leur forme racine.

Puis, on a constaté que les techniques d'embedding ont produits de meilleurs résultats par rapport aux techniques de représentation Tf-idf et Bag of Words. Cette amélioration peut être attribuée à la capacité des techniques d'embedding de mots à capturer les relations sémantiques entre les mots. Elles assignent des vecteurs de représentation aux mots de manière que des mots similaires aient des représentations similaires dans l'espace vectoriel. En conséquence, les techniques d'embedding de mots permettent de saisir de manière plus précise et informative les similarités et les relations entre les mots.

Pour les techniques de représentation basées sur les embeddings de mots, nous avons constaté que AraBERT génère de meilleurs résultats pour tous les modèles. La principale différence réside dans la nature des embeddings de mots. Avec Word2Vec (AraVec), les embeddings sont généralement statiques et représentent une seule représentation vectorielle pour chaque mot, indépendamment du contexte. En revanche, AraBERT génère des embeddings contextualisés qui capturent le sens des mots en fonction de leur contexte spécifique dans une phrase ou un document. Cela signifie qu'un mot peut avoir plusieurs représentations vectorielles (embeddings) différentes selon son utilisation et son contexte dans le texte [35].

L’utilisation de modèle CNN avec l’embedding de mots AraBERT, peut bénéficier à la fois de la précision contextuelle des embeddings d'AraBERT et de la capacité du CNN à extraire des caractéristiques locales. Cette combinaison peut conduire à des performances légèrement supérieures par rapport à d'autres combinaisons de modèles et de techniques de représentation des données, en particulier pour notre ensemble de données. Cependant, il est important de noter que les résultats peuvent varier en fonction du dataset utilisé. Lorsque nous

avons testé nos modèles sur les trois ensembles(Arabic Company Reviews, ASTD et Arabic 100k Reviews) , nous avons constaté que le modèle BiLSTM avec la représentation des mots AraBERT donnait de meilleurs résultats. D'après [21] cette différence s'explique par le fait que le modèle BiLSTM est capable d'apprendre de manière plus efficace le contexte de chaque mot dans le texte. Contrairement au modèle LSTM, le BiLSTM utilise à la fois des informations contextuelles précédentes et suivantes grâce à ses couches cachées en avant et en arrière, et le BiLSTM est capable d'extraire des informations sémantiques plus riches. Par conséquent, si le dataset de test est riche en informations et varié, comme dans les trois ensembles utilisés pour le test le BiLSTM serait le meilleur choix.

De plus, les tests montrent que les résultats obtenus sur l'ensemble de données "Arabic Company Reviews" sont meilleurs que ceux obtenus sur le dataset " Arabic 100k Reviews " qui sont eux meme meilleurs de ceux de l'ensemble "ASTD" Plusieurs. Plusieurs raisons expliquent ces résultats, notamment :

- On a entrainé notre modèle sur un dataset d'avis d'hôtel, tandis que le dataset "Arabic Company Reviews" touche le domaine de restauration, qui est un sous-domaine de l'hôtellerie. Une partie des données du dataset "Arabic 100k Reviews" concerne également le domaine de l'hôtellerie. Par conséquent, notre modèle donne de meilleurs résultats lorsqu'il est testé sur ces deux datasets par rapport à l'ensemble de données "ASTD".
- La diversité du vocabulaire des mots peut également avoir une influence sur les résultats des tests. C'est le cas pour le dataset "Arabic Company Reviews" où les mots sont moins variés par rapport au dataset "Arabic 100k Reviews", ce qui nous donne de meilleurs résultats. D'autre part, le vocabulaire est plus varié par rapport au dataset ASTD, mais il nous donne de meilleurs résultats car il concerne le domaine de l'hôtellerie, qui est proche du domaine sur lequel notre modèle a été entraîné. Il est important de noter que notre modèle a été entraîné sur un sous-ensemble de données composé de 27 992 mots uniques.
- Le modèle a été entraîné sur un ensemble de données comprenant à la fois de l'arabe classique et du dialecte. C'est pour cette raison que les tests effectués sur les trois ensembles de données donnent des résultats acceptables, car tous les ensembles sont également écrits en arabe classique et dialecte.

Donc l'efficacité d'un modèle peut varier en fonction de la nature des données de test utilisées. Certains modèles comme le BiLSTM peuvent mieux performer sur certains types de données ou les avis sont riche en information sémantiques, tandis que d'autres modèles comme le CNN peuvent être plus adaptés dans le cas où les données sont simples et invariantes ou une extraction des caractéristiques locales et suffisante. Il est donc important de sélectionner le modèle le plus approprié en fonction des caractéristiques spécifiques du dataset sur lequel il sera appliqué.

VI Conclusion

En résumé, ce chapitre a présenté les résultats de nos expérimentations sur la détection d'opinion en langue arabe en utilisant l'apprentissage profond, en examinant l'effet du stemming, des techniques de représentation des mots et des modèles de Deep Learning. De plus, nous avons validé nos résultats expérimentaux en les testant sur d'autres datasets. Ces analyses approfondies nous permettent de conclure quant à l'efficacité de nos approches dans le domaine de la détection d'opinion en langue arabe.

Conclusion générale

Ce mémoire avait pour objectif d'explorer différentes techniques de détection d'opinions en langue arabe, en mettant l'accent sur les modèles basés sur le Deep Learning. Tout au long de cette étude, nous avons examiné diverses techniques et entrainé plusieurs modèles basés sur les réseaux de neurones CNN, LSTM, BiLSTM et la combinaison de réseau de neurones CNN-BiLSTM, sur le dataset HARD. Cependant le modèle CNN nous a donné une meilleure exactitude sur le dataset HARD avec 95.32% en utilisant la représentation des mots AraBERT et sans utiliser l'étape de stemming.

Nous avons constaté que les techniques d'embedding de mots, notamment AraBERT, sont capables de capturer les relations sémantiques entre les mots, ce qui leur permet de fournir des représentations plus riches et précises pour la détection d'opinion. Comparées aux approches traditionnelles telles que TF-IDF et Bag of Words, elles offrent de meilleures performances. On a aussi constaté que l'efficacité d'un modèle de deep learning change dépend les données sur lesquelles il est entrainé et testé. Si un modèle a été entrainé sur un corpus grand et variées il peut nous donner de meilleure performance lors des tests et le BiLSTM serait le meilleur choix car il est capable d'extraire des informations contextuelles. Cependant, si les données ne présentent pas une grande diversité, le modèle CNN peut être suffisant pour extraire que les caractéristiques locales.

L'étude a également souligné l'importance de choisir un dataset le plus généralisés, le plus varié et surtout un équilibré pour l'entrainement afin d'obtenir de bons résultats de test.

Malgré ces résultats encourageants, il convient de noter certaines limitations de cette étude :

- Nous avons principalement travaillé avec un ensemble de données spécifique à un seul domaine, en raison de l'absence de grands ensembles de données variés en langue arabe pour la détection d'opinions.
- De plus, le domaine de la détection d'opinion en langue arabe reste un défi en raison des spécificités linguistiques et culturelles, ce qui ouvre la voie à de futures recherches dans ce domaine.

De ce fait, plusieurs perspectives de recherche futures peuvent être envisagées pour améliorer encore d'avantage cette étude. Il serait intéressant d'élargir nos expérimentations à d'autres domaines et à des ensembles de données plus vastes, provenant de différents domaines ou sources, tout en explorant d'autres modèles de Deep Learning tels que le mécanisme d'attention en utilisant les auto-encodeurs. En outre, il serait pertinent d'évaluer l'impact des techniques de représentation des données les plus récentes, telles qu'ELMO, ULMFiT, GPT et XLNet, sur la performance de nos modèles. Ces pistes de recherche permettraient d'enrichir notre compréhension et d'améliorer les performances de la détection d'opinions dans les textes arabes basée sur l'apprentissage profond.

Bibliographie

[1] « Most used languages online by share of websites 2023 | Statista ». https://www.statista.com (consulté le 31 mai 2023).

[2] B. Liu, « Sentiment Analysis and Opinion Mining », Morgan & Claypool Publishers, 2012.

[3] SCAD Institute of Technology, IEEE Electron Devices Society, et Institute of Electrical and Electronics Engineers, *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud (I-SMAC 2017) : 10-11, February 2017*.

[4] B. Pang, L. Lee, et S. Vaithyanathan, « Thumbs up? Sentiment Classification using Machine Learning Techniques », p. 79-86, 2002, doi: 10.3115/1118693.1118704.

[5] S. Kolkur, G. Dantal, et R. Mahe, « International Journal of Current Engineering and Technology Study of Different Levels for Sentiment Analysis », *International Journal of Current Engineering and Technology*, vol. 5, nº 2, 2015, Consulté le: 1 juin 2023. [En ligne]. Disponible sur: http://inpressco.com/category/ijcetGeneralArticle

[6] B. Pang et L. Lee, « A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts », 2004. [En ligne]. Disponible sur: www.cs.cornell.edu/people/pabo/movie-

[7] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, et S. Manandhar, « SemEval-2014 Task 4: Aspect Based Sentiment Analysis », *8th International Workshop on Semantic Evaluation, SemEval 2014 - co-located with the 25th International Conference on Computational Linguistics, COLING 2014, Proceedings*, p. 27-35, 2014, doi: 10.3115/V1/S14-2004.

[8] P. Subasic et A. Huettner, « Affect analysis of text using fuzzy semantic typing », *IEEE Transactions on Fuzzy Systems*, vol. 9, nº 4, p. 483-496, août 2001, doi: 10.1109/91.940962.

[9] A. F. M. N. H. Nahin, J. M. Alam, H. Mahmud, et K. Hasan, « Identifying emotion by keystroke dynamics and text pattern analysis », *Behaviour and Information Technology*, vol. 33, nº 9, p. 987-996, sept. 2014, doi: 10.1080/0144929X.2014.907343.

[10] M. N., I. M., A. H., et H. A., « Opinion Mining and Analysis for Arabic Language », *International Journal of Advanced Computer Science and Applications*, vol. 5, nº 5, 2014, doi: 10.14569/IJACSA.2014.050528.

[11] B. Hammo, H. Abu-Salem, et S. Lytinen, « QARAB », *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages* -, 2002, Consulté le: 1 juin 2023. [En ligne]. Disponible sur: https://www.academia.edu/7690602/QARAB_A_Question_Answering_System_to_Support_the_Arabic_Language.

[12] B. K. Bhavitha, A. P. Rodrigues, et N. N. Chiplunkar, « Comparative study of machine learning techniques in sentimental analysis », *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2017*, p. 216-221, juill. 2017, doi: 10.1109/ICICCT.2017.7975191.

[13] J. Patterson et A. Gibson, « Deep Learning A Practitioner's Approach », 2017. [En ligne]. Disponible sur: http://oreilly.com/safari

[14] « Définitions : neurone - Dictionnaire de français Larousse ». https://www.larousse.fr/dictionnaires/francais/neurone/54401 (consulté le 3 mai 2023).

[15] I. Goodfellow, Y. Bengio, A. Courville, et J. Heaton, « Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning », *Genetic Programming and Evolvable Machines 2017 19:1*, vol. 19, nº 1, p. 305-307, oct. 2017, doi: 10.1007/S10710-017-9314-Z.

[16] N. Buduma et N. Locascio, « The Neural Network », *Fundamentals of deep learning : designing next-generation machine intelligence algorithms*, p. 1-15, juin. 2017.

[17] « Activation Function in Deep learning | Analytics Vidhya ». https://medium.com/analytics-vidhya/activation-function-c762b22fd4da (consulté le 27 mai 2023).

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, et J. Dean, « Distributed Representations of Words and Phrases and their Compositionality », *Adv Neural Inf Process Syst*, vol. 26, 2013.

[19] A. B. Soliman, K. Eissa, et S. R. El-Beltagy, « AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP », *Procedia Comput Sci*, vol. 117, p. 256-265, janv. 2017, doi: 10.1016/J.PROCS.2017.10.117.

[20] Y. Lin, S. Shen, Z. Liu, H. Luan, et M. Sun, « Neural Relation Extraction with Selective Attention over Instances ».

[21] H. Elfaik et E. H. Nfaoui, « Deep Bidirectional LSTM Network Learning-Based Sentiment Analysis for Arabic Text », *Journal of Intelligent Systems*, vol. 30, nº 1, p. 395-412, janv. 2021, doi: 10.1515/jisys-2020-0021.

[22] A. M. Alayba, V. Palade, M. England, et R. Iqbal, « Arabic language sentiment analysis on health services », in *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, 2017, p. 114-118. doi: 10.1109/ASAR.2017.8067771.

[23] A. M. Alayba, V. Palade, M. England, et R. Iqbal, « Improving Sentiment Analysis in Arabic Using Word Representation », in *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)*, 2018, p. 13-18. doi: 10.1109/ASAR.2018.8480191.

[24] A. Setyanto *et al.*, « Arabic Language Opinion Mining Based on Long Short-Term Memory (LSTM) », *Applied Sciences (Switzerland)*, vol. 12, nº 9, mai 2022, doi: 10.3390/app12094140.

[25] H. Elzayady, K. M. Badran, et G. I. Salama, « Arabic Opinion Mining Using Combined CNN - LSTM Models », *International Journal of Intelligent Systems and Applications*, vol. 12, nº 4, p. 25-36, août 2020, doi: 10.5815/ijisa.2020.04.03.

[26] M. Heikal, M. Torki, et N. El-Makky, « Sentiment Analysis of Arabic Tweets using Deep Learning », *Procedia Comput Sci*, vol. 142, p. 114-122, janv. 2018, doi: 10.1016/J.PROCS.2018.10.466.

[27] A. Alwehaibi, M. Bikdash, M. Albogmi, et K. Roy, « A study of the performance of embedding methods for Arabic short-text sentiment analysis using deep learning approaches », *Journal of King Saud University - Computer and Information Sciences*, vol. 34, nº 8, p. 6140-6149, sept. 2022, doi: 10.1016/j.jksuci.2021.07.011.

[28] A. H. Ombabi, W. Ouarda, et A. M. Alimi, « Deep learning CNN–LSTM framework for Arabic sentiment analysis using textual information shared in social networks », *Soc Netw Anal Min*, vol. 10, nº 1, déc. 2020, doi: 10.1007/s13278-020-00668-1.

[29] S. Khoja et R. Garside, "Stemming Arabic text." Computing Department, Lancaster University, Lancaster, UK, 1999.

[30] K. Taghva, R. Elkhoury, et J. Coombs, « Arabic stemming without a root dictionary », *International Conference on Information Technology: Coding and Computing, ITCC*, vol. 1, p. 152-157, 2005, doi: 10.1109/ITCC.2005.90.

[31] A. H. Kreaa, A. S Ahmad, et K. Kabalan, « Arabic Words Stemming Approach Using Arabic Wordnet », *International Journal of Data Mining & Knowledge Management Process*, vol. 4, nº 6, p. 01-14, nov. 2014, doi: 10.5121/IJDKP.2014.4601.

[32] A. B. Soliman, K. Eissa, et S. R. El-Beltagy, « AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP », in *Procedia Computer Science*, Elsevier B.V., 2017, p. 256-265. doi: 10.1016/j.procs.2017.10.117.

[33] W. Antoun, F. Baly, et H. Hajj, « AraBERT: Transformer-based Model for Arabic Language Understanding », févr. 2020, [En ligne]. Disponible sur: http://arxiv.org/abs/2003.

[34] P. M. Radiuk, « Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets », *Information Technology and Management Science*, vol. 20, nº 1, janv. 2018, doi: 10.1515/ITMS-2017-0003.

[35] K. Guu, K. Lee, Z. Tung, P. Pasupat, et M. W. Chang, « REALM: Retrieval-Augmented language model pre-training », *37th International Conference on Machine Learning, ICML 2020*, vol. PartF168147-6, p. 3887-3896, 2020.