



المدرسة العليا للتكنولوجيا - سافي
ÉCOLE SUPÉRIEURE DE TECHNOLOGIE-SAFI



Université Cadi Ayyad
École Supérieure De Technologie-Safi
Département : Informatique
Filière : GI

Rapport De Projet De Fin D'étude
Génie informatique

Deep learning pour l'analyse automatique du sentiment des tweets
arabes

Réalisé par :
HARIB Douaa

MOUAINE Rabab
ZAHEDI Chaymaa

Encadré par :
Pr. Soufiane Hourri

Tutoré par :
LAYOUNE Ghita
ZAFER kenza

ANNÉE UNIVERSITAIRE : 2021/2022

Remerciements

La réalisation de ce projet n'aurait pas été possible sans l'aide et le soutien continu de plusieurs personnes. Avant toute chose, je tiens à exprimer ma profonde gratitude envers notre encadrant Dr Soufiane Hourri pour nous offrir la chance d'approfondir notre connaissance dans le domaine de L'intelligence artificielle, sa sollicitude et ses nombreux conseils ont été indispensables pour la rédaction de ce rapport. On souhaite également exprimer notre reconnaissance envers Mlle LAYOUNE Ghita et Mlle ZAFER Kenza pour nous avoir aidé à mieux comprendre le fonctionnement des parties techniques dans le domaine de l'apprentissage automatique. Les sources détaillées dont elles partageaient avec nous, ont grandement aidé à mieux assimiler la matière technique. Finalement, on aimerait remercier de tout cœur nos parents, nos frères et nos amies pour leur soutien incommensurable au cours de ces deux années universitaires passionnantes.

Astract

Avec l'épanouissement du E-commerce, l'opinion et la position des utilisateurs et des clients ne cessent de gagner de l'importance. Ceci vise à accroître son efficacité en se basant sur les opinions des clients. Ainsi, plusieurs méthodes pour analyser ces données ont vu le jour. L'analyse des sentiments, en particulier, sert à détecter la présence des sentiments au sein de textes sur divers niveaux. Ce projet, dédiée à l'analyse des sentiments par apprentissage automatique, est divisée en deux parties. La première partie présente tout d'abord une étude théorique de l'analyse des sentiments en se focalisant sur ceux avec la langue arabe sur les réseaux sociaux, tout particulièrement Twitter. La deuxième partie sera une réalisation de celle-ci.

Table des matières

Liste des Abréviations	7
Chapitre I Introduction	8
1 Introduction	8
2 Problématique	8
3 Importance d'analyse de sentiments	9
4 Importance de l'intelligence artificielle et du Deep learning	10
5 Cahier de charge	10
Chapitre II L'état de l'art	18
1 Analyse des sentiments	18
2 Apprariantissage profond-Deep Learning	20
Chapitre III Méthodologie	26
1 Les réseaux de neurones récurrents bidirectionnels (BRNN)	26
Chapitre IV Expérimentations et résultats	30
1 Introduction	30
2 Dataset	31
3 Outils et bibliothèques utilisées	31
4 Prétraitement	34
5 Extraction et présentation des descripteurs	36
6 Résultat obtenue par le model	42
7 Comparaison et discussion	45
Chapitre V Conclusion et perspective	46
1 Perspective	46
2 Conclusion générale :	47
Biographie	49

Table des figures

I.1	"Sentiment Analysis", l'outil de marketing sur-mesure.	10
I.2	Buts et objectifs du projet et résultats opérationnels.	11
I.3	Modèle de developpemnet en Cascade.	12
I.4	Jalon du projet	14
I.5	organigramme« la structure de l'équipe de projet ».	14
I.6	organigramme.	15
I.7	Tableau de responsabilité	15
I.8	Exemple de traitement des tweets.	17
I.9	Spécifique techninqe utilisé.	17
II.1	Un diagramme montrant à quel point le deep learning est une sorte d'apprentissage par représentation, qui est à son tour une sorte de ma- chine learning, utilisé pour de nombreuses approches de l'intelligence artificielle, mais pas toutes.	21
II.2	exemple de reseau multicouche	23
II.3	Illustration du modèle d'un neurones biologique.	23
II.4	fonction de l'agrégation pondérée de ses nombreuses entrées : x_0, \dots, x_N , où W_i est le poids de l'entrée X_i , f est une fonction d'activation, et b est le biais.	24
II.5	Exemple de marquage POS	25
II.6	Exemple de reconnaissance d'entité nommée	25
III.1	démonstration de RNN bidirectionnelle	27
III.2	Structure générale des réseaux de neurones récurrents bidirectionnels .	28
IV.1	La méthodologie realiser pour l'analyse des sentiments sur les tweets arabes	30
IV.2	la distribution des Tweets	31
IV.3	capture d'écran des bibliotheques utilisées	34
IV.4	Exemples des Tweets avant et après pretraitement	35
IV.5	capture d'écran de fonction créer pour le nettoyage de dataset	36
IV.6	Processus de prétraitement et transformation du texte.	37
IV.7	exemple de tokenization d'un tweet de notre dataset	37

IV.8	exemple de stemming d'un tweet qui a déjà subi la tokenization de notre dataset	38
IV.9	capture d'écrans de fonction utiliser pour la presentation vectoriel . . .	38
IV.10	exemple de Réseau de Neurons Standard avant et après l'application de Dropout	40
IV.11	capture d'écran de model RNN réalisé	40
IV.12	capture d'écrans de summary de model RNN	41
IV.13	capture d'écrans de fonction de division de dataset	41
IV.14	capture d'écrans de training de model	42
IV.15	valeur de precision, recall et f1-score obtenu de notre model	43
IV.16	graph illustre les résultats moyens de performance des classifieurs BRNN pour training et validation accuracy	44
IV.17	graph illustre les résultats moyens de performance des classifieurs BRNN pour training et validation loss	44

Liste des Abréviations

BM : Boltzmann Machine.

BRNN : bidirectional recurrent neural networks .

CNN : Convolutional Neural Network.

DNN : Deep Neural Network.

GAN : Generative adversarial networks.

GRU : Gated Recurrent Unit.

LSTM : long short-term memory.

NLP : natural language processing.

RBM : Restricted Boltzmann Machine.

RNN : Réseau de neurones récurrents.

Chapitre I

Introduction

1 Introduction

L'analyse des sentiments Twitter signifie utiliser des techniques avancées d'exploration de texte pour étudier le sentiment du texte (ici, tweet) dans le type positif, négatif et neutre. il s'appelle également Opinion Mining, est principalement destiné à analyser les conversations, les opinions et le partage de points de vue (le tout dans le cadre de tweets) pour décider de la stratégie commerciale, de l'analyse politique et également pour évaluer les actions publiques. Les analyses de sentiment visent souvent à identifier les tendances dans le contenu des tweets, qui sont ensuite analysés par des algorithmes d'apprentissage automatique. L'analyse des sentiments est un outil crucial dans le domaine du marketing des médias sociaux, car elle discutera de la façon dont elle sera habituée à prédire le comportement de la personnalité en ligne d'un utilisateur. . L'analyse des sentiments est utilisée pour enquêter sur le sentiment d'un message donné ou sur un sujet donné. Outils populaires dans le marketing des médias sociaux.

2 Problématique

Twitter est un site Web de réseautage social populaire où les membres créent et interagissent avec des messages connus sous le nom de « tweets ». Cela permet aux individus d'exprimer leurs pensées ou leurs sentiments sur différents sujets. Diverses parties différentes telles que les consommateurs et les spécialistes du marketing ont effectué une analyse des sentiments sur ces tweets pour recueillir des informations sur les produits ou pour effectuer une analyse de marché. De plus, grâce aux progrès

récents des algorithmes d'apprentissage automatique, j'ai pu améliorer la précision de nos prédictions d'analyse des sentiments. Dans ce rapport, je tenterai d'effectuer une analyse des sentiments sur les « tweets » à l'aide de divers algorithmes d'apprentissage automatique différents. Si le tweet contient à la fois des éléments positifs et négatifs, le sentiment le plus dominant doit être choisi comme étiquette finale. J'ai utilisé l'ensemble de données qui a été exploré et étiqueté positif/négatif. Les données fournies sont accompagnées d'émoticônes, de noms d'utilisateur et de hashtags qui doivent être traités et convertis en un format standard. J'ai également besoin d'extraire des caractéristiques utiles du texte telles que les unigrammes et les bigrammes qui sont une forme de représentation du « tweet » Utilisé divers algorithmes d'apprentissage automatique pour effectuer une analyse des sentiments à l'aide des caractéristiques extraites. Cependant, se fier uniquement à des modèles individuels n'a pas donné une grande précision, j'ai donc choisi les meilleurs modèles pour générer un ensemble modèle. L'assemblage est une forme de technique d'algorithme de méta-apprentissage où j'ai combiné différents classifieurs afin d'améliorer la précision de la prédiction. Enfin, je rapporte mes résultats expérimentaux et mes conclusions à la fin.

3 Importance d'analyse de sentiments

Certains rejetaient cet engouement pour l'analyse des données en ligne tel l'analyse des sentiments, les adeptes de cette pratique ont persévéré et inutile de dire qu'ils ont vu juste. Le 21ème siècle a été marqué par un intérêt sans précédent pour les réseaux sociaux. De nos jours, passer 24 heures sans entrer en contact d'une manière ou d'une autre avec ces derniers est devenu chose difficile. Cette affirmation vaut d'autant plus pour la partie la plus jeune de la population. Facebook, Twitter, Instagram, Snapchat, ces plateformes sociales font désormais partie du quotidien.

Mis à part ce développement numérique, une grande avancée technologique s'est également développée. Les anciens algorithmes logiques comme les arbres à décisions ont petit à petit fait place aux algorithmes statistiques comme la classification naïve bayésienne. Bien que complexes, les études réalisées à leurs sujets ne cessent de découvrir de nouvelles méthodes pour puiser dans leur potentiel. Grâce à ces algorithmes, l'apprentissage automatique devient de plus en plus performant, ce qui a augmenté la rapidité et l'ampleur avec laquelle des données peuvent être traitées. Désormais, des masses énormes de données peuvent être analysées en un temps record. En dehors de son utilité dans le domaine de la recherche, cette aptitude à rapidement analyser les données a également trouvé sa voie dans les domaines commerciaux. Entreprises,

bureaux publicitaires et réseaux sociaux, tous désirent récolter le plus d'informations possible à propos de leurs (futurs) clients. Sentiment, opinion, âge, sexe, taille, intérêts, éducation, profession, situation familiale, tous ces paramètres cachent à présent une valeur commerciale. Dans ce rapport, nous conduisons une étude pilote pour découvrir l'efficacité de l'apprentissage automatique supervisé en ce qui concerne l'analyse des sentiments appliquée à des tweets arabes.

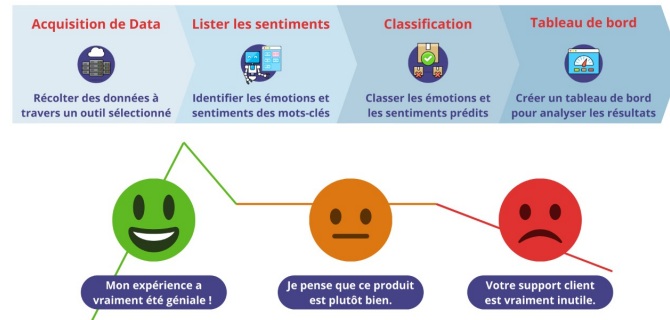


FIGURE I.1 – "Sentiment Analysis", l'outil de marketing sur-mesure.

4 Importance de l'intelligence artificielle et du Deep learning

L'arrivée de l'ère numérique, et surtout du web, a changé de manière significative le comportement de la société contemporaine. Avec un monde de plus en plus automatisé, les endroits où la technologie ne s'est pas encore imposée restent rares. Désormais, ce développement, qui peut être considéré comme une démocratisation informatique, génère une riche mine de données exploitables appelée « Big Data ». Comparable en quelque sorte à la façon dont était convoité l'or noir au 19ème siècle, les puits de « Big Data » sont devenus tout aussi convoités dans le domaine des affaires, de la recherche et de l'informatique. Suite à ce développement, les chercheurs ont depuis tenté de conceptualiser des méthodes afin d'analyser ce « Big Data » d'une façon méthodique, structurée et efficace. Ainsi, parmi de nombreuses autres méthodes, l'analyse des sentiments a été développée pour analyser spécifiquement la polarité des textes postés en ligne sur divers niveaux.

5 Cahier de charge

5.1 Aperçu du projet :

Résumé du projet :

Avec la démocratisation de l'espace web, l'opinion et la position des utilisateurs web ne cessent de gagner de l'importance. Ainsi, plusieurs méthodes pour analyser ces données ont vu le jour. L'analyse des sentiments, en particulier, sert à détecter la présence des sentiments au sein de textes sur divers niveaux. Cette thèse, dédiée à l'analyse des sentiments par apprentissage automatique, est divisée en deux parties. La première partie présente tout d'abord une étude théorique de l'analyse des sentiments en se focalisant sur ceux avec la langue arabe sur les réseaux sociaux, tout particulièrement Twitter. La deuxième partie une réalisation de celle-ci.

Buts et objectifs du projet et résultats opérationnels :

Dans ce travail de recherche, notre objectif consiste à étudier et analyser les sentiments dans les tweets arabes en utilisant les techniques les plus récentes pour le classement automatique du texte : les machines à support de vecteurs (SVM), Les réseaux de neurones convolutifs (CNN), les réseaux de neurones récurrents (RNN), Logistique Régression (LR) et les réseaux récurrents à mémoire court et long terme (LSTM) et par les techniques de l'approche lexicale. Nous avons développé une application web, qui sera utilisée après, pour construire de nouveaux corpus et de les analyser. Pour cela, nous avons ajouté les modèles d'apprentissage mentionnés comme outils disponibles dans cette application.

N°	Buts	Objectifs	Résultats opérationnels
1	Analyse des sentiments des tweets arabe	<ul style="list-style-type: none">• Permet la machine de réalisé un deep learning pour analyser les tweet	<ul style="list-style-type: none">• Algorithme qui permet le traitement de ces tweets
2	Réaliser une interface graphique	<ul style="list-style-type: none">• Utilisé un Framework afin d'ajouter l'interface graphique	<ul style="list-style-type: none">• Interface graphique qui facilite le traitement des tweets

FIGURE I.2 – Buts et objectifs du projet et résultats opérationnels.

Contexte et dépendances :

Le succès ou l'échec d'une marque ne dépend pas seulement des ventes, mais aussi de l'opinion des clients. Il est très important de comprendre ce que les clients actuels et potentiels pensent de votre marque, qu'ils aient déjà acheté un produit/service

ou non. La marque s'inscrit-elle dans les tendances actuelles ? La perception de la marque par le public cible est-elle positive ou négative ? Voilà des questions que toute entreprise devrait se poser régulièrement. D'où notre projet de fin d'étude L'analyse des sentiments est fait pour déterminer comment un public cible accueille et perçoit une marque. Voyez-vous que même les experts du marché boursier utilisent l'analyse du sentiment pour estimer les variations d'actions en fonction du comportement d'achat et du sentiment général des investisseurs du marché.

Le processus de développement :

Afin de réaliser ce projet on est amenée à planifier un algorithme qui nous servira à organiser le travail. D'où vient l'idée de travailler avec un modèle de développement en cascade. Le modèle de cycle de vie en cascade a été mis au point dès 1966, puis formalisé aux alentours de 1970. Dans ce modèle le principe est très simple : Chaque phase ne commence qu'une fois les résultats de la phase précédente validés. Le point fort de cette approche est de garantir l'existence d'une documentation bien structurée. Plusieurs variantes du modèle existent, dont l'ajout d'une phase de planification en amont, la réalisation préalable d'un prototype, la décomposition de la phase de validation, et le retour aux phases précédentes en cas de défauts découverts en aval. Dans le domaine du développement logiciel, la phase de conception détermine l'architecture du système, la mise en œuvre correspond principalement aux activités de programmation, et la phase de validation comprend pour une grande part des tests.

Le processus de développement en cascade « Waterfall » :

La figure suivante la représentation de celui-ci :

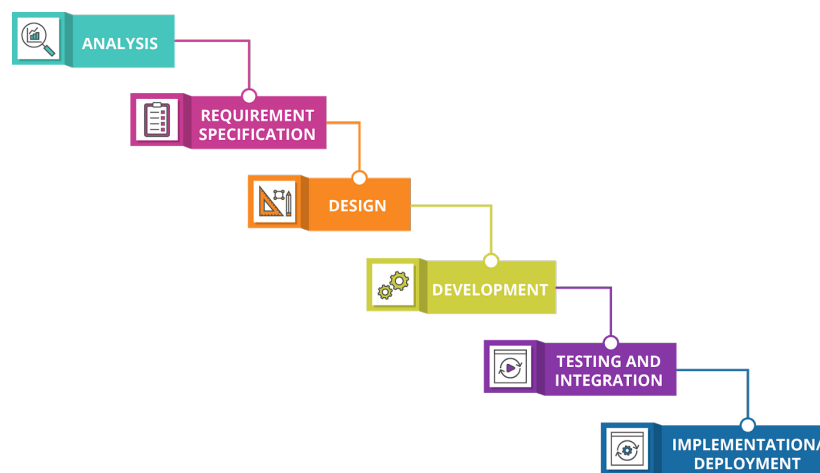


FIGURE I.3 – Modèle de developpemnet en Cascade.

Avantages de cette méthode :

- Simple et facile à comprendre. ;
- Force la documentation : une phase ne peut se terminer avant qu'un document soit validé. ;
- Le test est inhérent à chaque phase. ;
- Les progrès sont tangibles (pour l'équipe de développement).

Limites de cette méthode :

- Non compréhensibles par les clients. ;
- Manque de flexibilité (ne traite pas les évolutions, notamment des exigences). ;
- Problèmes découverts en phase de validation. ;
- Irréaliste dans de nombreux cas.

Contraintes du projet :

Comme toute méthode de recherche, l'analyse des sentiments connaît bien entendu certains problèmes potentiels. Une possibilité d'analyse si détaillée peut dans certains cas s'avérer un handicap. Bien que les phrases courtes soient les meilleures sources de données pour de tels systèmes, ces dernières ne fournissent que peu de contexte à l'annotateur. De plus, comprendre le fil des idées d'une personne inconnue en si peu de mots reste un challenge à ne pas sous-estimer. Le doute sur la polarité de certaines entités peut des fois s'imposer. Similairement, si les textes contiennent des mots vagues comme «cet endroit, cette chose, ce truc...», trouver l'entité à laquelle l'utilisateur renvoie peut devenir compliqué.

Produits livrables :

- Il s'agit de mettre en place un système qui permet de :
- Teindre au courant ce que les gens aiment et n'aiment pas .
 - Classifier les tweets arabes en deux catégories :
 - Teindre au courant ce que les gens aiment et n'aiment pas .
 - Classifier les tweets arabes en deux catégories : positif et négative.

Jalons et planning prévisionnel :

Les Jalon marquent une rupture pour valider une étape. Ils sont très importants dans le cycle de vie du projet . Ils fixent des objectifs intermédiaires et contribuent à la prévention de l'effet tunnel. En effet, les validations partielles sont l'occasion d'une rencontre entre les différentes parties prenantes pour s'assurer et valider que les travaux vont dans le bon sens.

Jalon du projet	Description	Date prévue
1. étude théorique	Consiste l'étude de fonctionnement de deep learning et machine learning.	10/02/2022
2. exécution de l'algorithme	Compilation de l'algorithme avec une base de donne précise.	21/02/2022
3. implémentation de l'interface graphique	A l'aide d'un Framework, il faut ajouter une interface graphique qui facilite le fonctionnement de l'application	10/03/2022

FIGURE I.4 – Jalon du projet

5.2 Organisation du projet

Structure de l'équipe du projet :

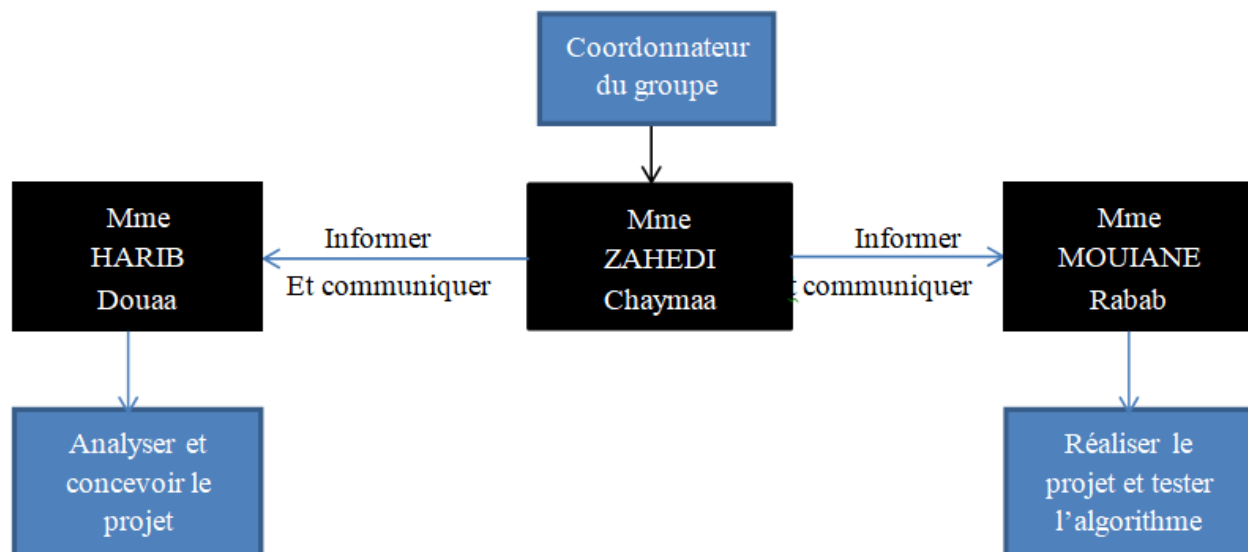


FIGURE I.5 – organigramme« la structure de l'équipe de projet ».

Contrairement à la plupart des gens qui choisissent l'autonomie même au sein de leur groupe nous avons optez pour la corporation et l'intégration, de chaque

membre de notre groupe dans tous les aspects et les volets traités. En cours de la réalisation du projet on a pris en considération la bonne organisation qui doit permettre de coordonner et gérer dans le temps, des moyens matériels et des moyens humains. Chaque participant du groupe de projet à un rôle précis qui contribue à

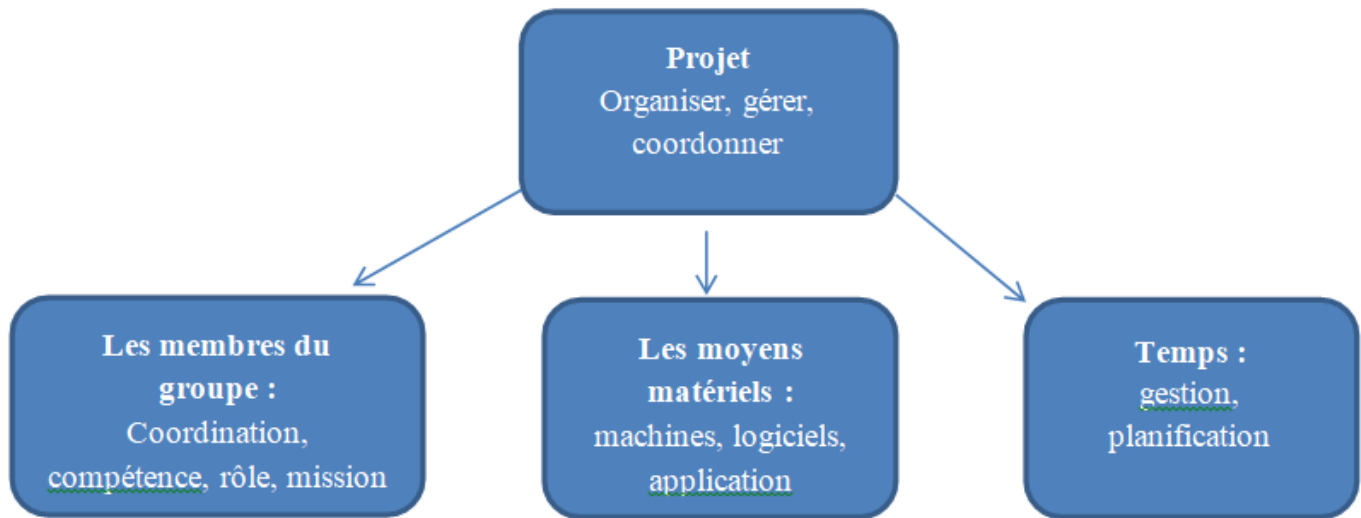


FIGURE I.6 – organigramme.

l'organisation et au bon fonctionnement du groupe. Chaque membre participe aux revues de projets qui permettent de faire le point sur l'avancement des travaux et la présentation des résultats.

Rôles et responsabilités :

La responsabilité en milieu de travail se produit lorsque chaque employé est tenu responsable de ses actions et de son rendement global. Un milieu de travail responsable est un milieu de travail qui possède une culture de confiance et de travail d'équipe. D'où on tient à distribuer à chaque membre du groupe un travail tel ce qui suit :

Rôle dans le cadre du projet	Responsabilités	Assigné à
[Gestionnaire de projet]	Veille sur le bon fonctionnement du groupe et gère le temps ainsi que les ressources du projet.	ZAHEDI Chaymaa
[Analyste des activités]	Se consacre au volet théorique du projet et la maintenance de l'équilibre dans les besoins du projet	HARIB Douaa
[Comité d'examen du projet]	Réalise les tests, règle les erreurs et maintient le projet.	MOUANE Rabab

FIGURE I.7 – Tableau de responsabilité

Installations et ressources du projet :

Au niveau de ce côté-là le projet a eu besoin de peu de ressources qui ont été géré par Mme MOUANE Rabab tel Google colab que nous avons utilisé sans l'installation de l'environnement Anaconda ou aucune autre source Colaboratory, souvent raccourci en "Colab", est un produit de Google Research. Colab qui permet à n'importe qui d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. C'est un environnement particulièrement adapté au machine learning, à l'analyse de données et à l'éducation. Ainsi, on a utilisé le langage python pour réaliser ce projet puisqu'il est le langage le plus populaire dans le monde de l'intelligence artificielle. Python est orienté objet et se veut relativement facile d'accès. Et finalement on a eu recours à la plateforme Github pour chercher certaines information concernant le projet.

5.3 Spécifications fonctionnelles :

L'analyse des sentiments est un processus pendant lequel la polarité (positive, négative ou neutre) d'un texte donné est déterminée. Nous nous intéressons dans ce travail à l'analyse des sentiments des tweets arabes par une approche basée sur l'apprentissage automatique, l'apprentissage profond et plusieurs algorithmes comme CNN, LSTM, RNN et GAN . Ce processus commence par la collecte des tweets et leur annotation à l'aide du crowdsourcing suivi d'une phase de prétraitement du texte afin d'extraire des mots arabes réduits à leur racine. Ces mots vont être utilisés pour la construction des variables d'entrée en utilisant plusieurs combinaisons de schémas d'extraction et de pondération. Pour réduire la dimensionnalité, une méthode de sélection de variables est appliquée. Les résultats obtenus des expérimentations sont très prometteurs.

```

[53] predict_sentiment(" مساء الورد يا جميل ")
1/1 [=====] - 0s 21ms/step
0.8164892 إيجابي
('Satisfied', 0.8164892)

[54] predict_sentiment(" ممكن اتغير على انسان بنسبة ??? لمجرد انه ابن وسخة بيبي الجميل والجدنة التي كانت يتعامل معها ههههه ")
3/3 [=====] - 0s 13ms/step
0.37584797 سلبي
('Unsatisfied', 0.37584797)

[55] predict_sentiment(" من الخير نفسه ")
1/1 [=====] - 0s 22ms/step
0.8164892 إيجابي
('Satisfied', 0.8164892)

predict_sentiment(" قهروني حرام ")
1/1 [=====] - 0s 48ms/step
0.37584797 سلبي
('Unsatisfied', 0.37584797)

```

FIGURE I.8 – Exemple de traitement des tweets.

5.4 Spécifications techniques :

Pour la réalisation de ce projet nous avons utilisés plusieurs techniques comme suite :

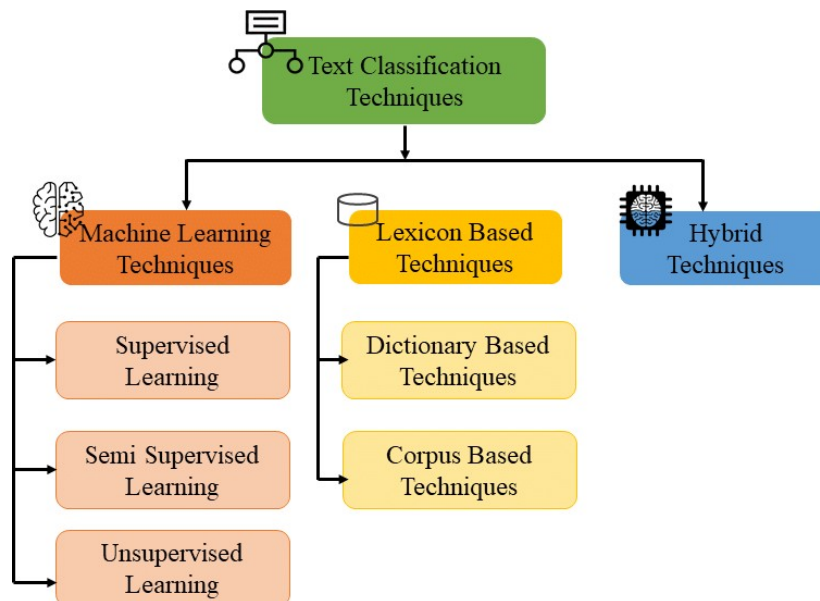


FIGURE I.9 – Spécifique techninqe utilisé.

Chapitre II

L'état de l'art

1 Analyse des sentiments

1.1 définition

Dans la littérature, Sentiment Analysis, Opinion Mining, Opinion Extraction, Sentiment Mining, Subjectivity Analysis, Affect Analysis, Emotion Analysis, Review Mining, Appraisal extraction, sont des termes utilisés pour désigner des technologies d'analyse automatique des discours, à partir de sources textuelles dématérialisées sur de grandes quantités de données.

La révolution de l'information et précisément l'explosion des plates-formes Web 2.0 telles que les forums de discussion, les blogs et les réseaux sociaux a permis aux utilisateurs de partager des idées et des opinions, d'exprimer leurs sentiments et bien plus encore. Cette révolution entraîne l'accumulation d'une énorme quantité de données pouvant contenir de nombreuses informations précieuses.

Tous les développements récents dans le domaine d'échange d'informations et d'opinions ont motivés la réalisation des applications informatiques conçues pour l'analyse et la détection des sentiments exprimés sur internet. Présentée dans la littérature sous le nom de « Opinion Mining » ou « Sentiment Analysis », l'analyse des sentiments s'utilise, entre autres, pour l'extraction d'opinions sur des sites web et des réseaux sociaux, l'éclaircissement du comportement des consommateurs, la recommandation de produits et l'explication des résultats des élections. Elle consiste à rechercher des textes évaluatifs sur Internet tels que des critiques et des recommandations à analyser de façon automatique ou manuelle et les sentiments qui y sont exprimés afin de mieux comprendre l'opinion publique.

Il a déjà été démontré par des études antérieures que l'analyse des sentiments s'avère particulièrement intéressante pour ceux qui ont intérêt à connaître l'opinion publique, que ce soit pour des raisons personnelles, commerciales ou politiques.

Ainsi, de nombreux systèmes autonomes ont déjà été développés pour l'analyse automatique des sentiments. Généralement, ces systèmes étaient entraînés aux textes évaluatifs traditionnels tels que les comptes rendus cinématographiques ou les critiques d'un livre. Toutefois, depuis quelques années se sont graduellement ajoutés à ces textes traditionnels, les textes non traditionnels tels que les messages envoyés via les réseaux sociaux. Ceux-ci constituent une source précieuse d'opinions échangées parmi les multiples internautes.

Par conséquent, il est important de concevoir des systèmes automatiques capable de rechercher et d'analyser les sentiments qui sont exprimés sur les réseaux sociaux.

A cet effet, une grande partie de cette étude sera consacrée à l'analyse automatique des sentiments exprimés dans des tweets arabe et permet de les classifiées en deux categorie :positifs et negatifs.

1.2 Domaine d'application de l'analyse des sentiments

L'importance de l'analyse des sentiments est présente dans plusieurs domaines ainsi plusieurs applications ont vu le jour dans ce contexte. Nous mentionnons brièvement quelques applications ci-dessous :

politique :

Aujourd'hui, les acteurs politiques ont suivi la tendance de l'analyse des sentiments, car avant de déclarer une nouvelle loi, les politiciens tentent de recueillir l'opinion des utilisateurs de médias sociaux sur cette loi. Il est hautement stratégique de connaître également l'opinion des internautes sur un politicien lors d'une élection présidentielle.

E-commerce :

Avant d'acheter un produit, la majorité des clients demandent conseil sur un produit ou un service donné et sont même disposés à payer plus pour un produit dont l'opinion est plus favorable qu'un autre, ce qui peut augmenter les ventes. Grâce à l'analyse des sentiments, les entreprises peuvent connaître l'opinion des clients sur leurs produits ou leurs services. Dans une perspective d'amélioration de leurs produits et d'augmentation de leurs ventes et revenus.

Education :

L'analyse des sentiments peut être utilisée pour extraire des informations utiles sur la méthodologie d'enseignement d'un enseignant et également sur le programme

du cours. Il identifie le degré d'apprentissage des étudiants, comprend leurs besoins, prévoit leurs performances et apporte des changements effectifs dans le style. Les résultats de l'analyse des sentiments aident les enseignants et les établissements à prendre des mesures correctives.

1.3 Analyse des sentiments avec twitter

Twitter est une plate-forme de communication basée sur le Web, qui permet à ses abonnés de diffuser des messages appelés « tweets » de 280 caractères maximum, leur permettant de partager des pensées, des liens ou des images. Par conséquent, Twitter est une source riche de données pour l'exploration d'opinion et l'analyse de sentiment. La simplicité d'utilisation et les services offerts par la plate-forme Twitter lui permettent d'être largement utilisée dans le monde arabe . Cette popularité nous donne accès à une mine riche d'informations qui peuvent servir comme base de données à l'analyse des tweets, qui nous fournissent des informations précieuses.

2 Apprentissage profond-Deep Learning

2.1 Intelligence artificielle, Machine learning et Deep learning

L'Intelligence Artificielle (IA) est la science dont le but est de faire par une machine des tâches que l'homme accomplit en utilisant son intelligence, telles que la perception visuelle, la reconnaissance de la parole, la prise de décision et la traduction entre les langues. Le cerveau sur lequel s'appuie l'IA est une technologie appelée apprentissage automatique (ou machine learning) qui est une technique de programmation informatique qui utilise des probabilités, statistiques pour donner aux ordinateurs la capacité d'apprendre par eux-mêmes sans programmation explicite . Le deep learning est un sous-domaine de machine learning, qui utilise les réseaux de neurones pour analyser différents facteurs avec une structure similaire au système neural humain.

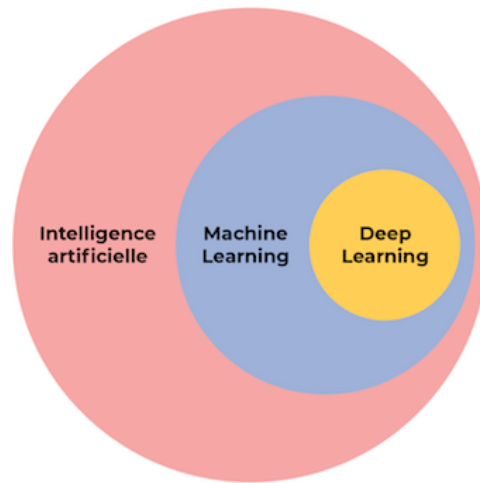


FIGURE II.1 – Un diagramme montrant à quel point le deep learning est une sorte d'apprentissage par représentation, qui est à son tour une sorte de machine learning, utilisé pour de nombreuses approches de l'intelligence artificielle, mais pas toutes.

2.2 Le Perceptron multicouche

Le perceptron multicouche (multilayer perceptron MLP) est un type de réseau neuronal formel organisé en plusieurs couches au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement, il s'agit donc d'un réseau à propagation directe (feedforward). Chaque couche est constituée d'un nombre variable de neurones, les neurones de la dernière couche (dite « de sortie ») étant les sorties du système global.

2.3 Algorithme d'apprentissage profond

Réseaux adversariaux génératifs (GAN)

Les GAN sont des algorithmes de deep learning génératifs qui créent de nouvelles instances de données qui ressemblent aux données d'apprentissage. Les GAN ont deux composants : un générateur, qui apprend à générer des données fausses, et un discriminateur, qui apprend à partir de ces fausses informations. Les développeurs de jeux vidéo utilisent les GAN pour améliorer les textures 2D à faible résolution des anciens jeux vidéo en les recréant en 4K ou à des résolutions plus élevées via l'apprentissage d'images.

Réseau de neurones récurrents(RNN)

Un réseau de neurones récurrents est un réseau de neurones artificiels présentant des connexions récurrentes. Un réseau de neurones récurrents est constitué d'unités

(neurones) interconnectées interagissant non-linéairement et pour lequel il existe au moins un cycle dans la structure. Les unités sont reliées par des arcs (synapses) qui possèdent un poids. La sortie d'un neurone est une combinaison non linéaire de ses entrées.

Réseaux de mémoire à long terme et à court terme (LSTM)

Les LSTM sont un type de réseau neuronal récurrent (RNN) qui peut apprendre et mémoriser des dépendances à long terme. Se souvenir d'informations passées pendant de longues périodes est le comportement par défaut.

Les LSTM conservent les informations dans le temps. Ils sont utiles pour la prédiction de séries chronologiques car ils se souviennent des entrées précédentes.

Les LSTM ont une structure en chaîne dans laquelle quatre couches en interaction communiquent de manière unique. Outre, les prédictions de séries chronologiques, les LSTM sont généralement utilisés pour la reconnaissance vocale, la composition musicale et le développement pharmaceutique.

Les réseaux neuronaux convolutifs (CNN)

CNN sont constitués de plusieurs couches et sont principalement utilisés pour le traitement des images et la détection des objets.

Les réseaux de neurones convolutifs ont une méthodologie similaire à celle des méthodes traditionnelles d'apprentissage supervisé : ils reçoivent des images en entrée, détectent les fonctionnalités de chacune d'entre elles, puis entraînent un classifieur dessus.

2.4 Les réseaux de neurones

Un réseau neuronal est l'association, en un graphe plus ou moins complexe, des objets élémentaires, les neurones formels. Les principaux réseaux se distinguent par l'organisation du graphe (en couches, complets. . .), c'est-à-dire leur architecture, son niveau de complexité (le nombre de neurones, présence ou non de boucles de rétroaction dans le réseau), par le type des neurones (leurs fonctions de transition ou d'activation) et enfin par l'objectif visé : apprentissage supervisé ou non, optimisation, systèmes dynamiques...

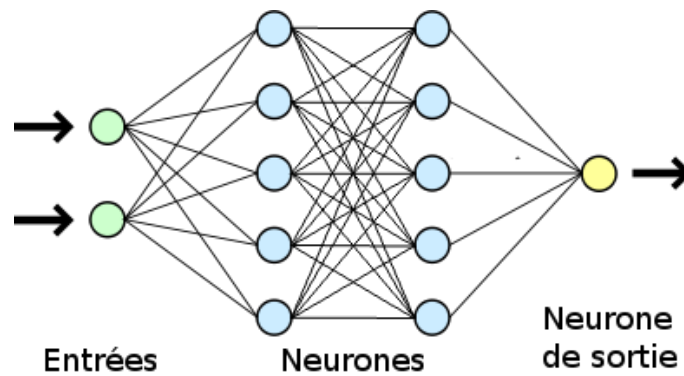


FIGURE II.2 – exemple de reseau multicouche

De façon très réductrice, un neurone biologique est une cellule qui se caractérise par :

- des synapses, les points de connexion avec les autres neurones, fibres nerveuses ou musculaires .
- des dendrites ou entrées du neurones .
- les axones, ou sorties du neurone vers d'autres neurones ou fibres musculaires .
- le noyau qui active les sorties en fonction des stimulations en entrée.

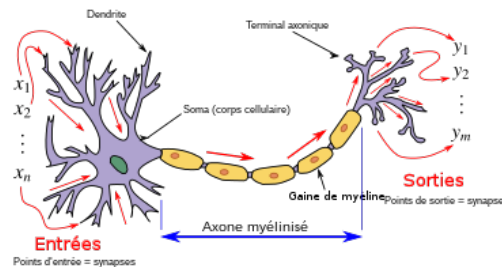


FIGURE II.3 – Illustration du modèle d'un neurones biologique.

Par analogie, le neurone formel est un modèle qui se caractérise par un neurones, le poids, le biase et la fonction d'activation.

Donc comment fonctionnent ce réseau de neurone ?

Au sein d'un réseau de neurones artificiels, le traitement de l'information suit toujours la même séquence : les informations sont transmises sous la forme de signaux aux neurones de la couche d'entrée, où elles sont traitées. À chaque neurone est attribué un « poids » particulier, et donc une importance différente. Associé à la fonction dite de transfert, le poids permet de déterminer quelles informations peuvent entrer dans le système.

À l'étape suivante, une fonction dite d'activation associée à une valeur seuil calculent et pondèrent la valeur de sortie du neurone. En fonction de cette valeur, un nombre plus ou moins grand de neurones sont connectés et activés.

Cette connexion et cette pondération dessinent un algorithme qui fait correspondre un résultat à chaque entrée. Chaque nouvelle itération permet d'ajuster la pondération et donc l'algorithme de façon à ce que le réseau donne à chaque fois un résultat plus précis et fiable.

$$y = f\left(\sum_i^N w_i x_i + b\right)$$

FIGURE II.4 – fonction de l'agrégation pondérée de ses nombreuses entrées : x_0, \dots, x_N , où W_i est le poids de l'entrée X_i , f est une fonction d'activation, et b est le biais.

2.5 Le traitement du langage naturel (NLP)

Definition

Le natural language processing (NLP) est une branche du machine learning qui vise à doter des programmes informatiques de la capacité de comprendre le langage humain naturel.

Pour ce faire, des programmes informatiques spécifiques sont développés. En effet, un ordinateur typique réclame qu'on lui parle dans un langage de programmation bien précis, balisé, structuré, sans ambiguïté. Le langage naturel humain est, lui confus. Pour permettre à un programme de comprendre le sens des mots, il faut employer des algorithmes capables d'analyser le sens et la structure pour "désambiguïser" les mots, de reconnaître certaines références, puis de générer du langage sur cette base.

Les étapes de NLP

- Tokenisation : permet de diviser le texte brut en petits morceaux, soit des mots soit des phrases. Si le texte est divisé en mots à l'aide d'une technique de séparation, cela s'appelle la segmentation des mots et la même séparation effectuée pour les phrases est appelée segmentation des phrases.

Il existe différentes méthodes et bibliothèques disponibles pour effectuer la tokenisation. NLTK, Gensim, Keras sont quelques-unes des bibliothèques qui peuvent être

utilisées pour accomplir la tâche.

- Stemming : un même mot peut se retrouver sous différentes formes en fonction du genre (masculin, féminin), du nombre (singulier, pluriel), la personne (moi, toi, eux...) etc. Le stemming désigne généralement le processus heuristique brut qui consiste à découper la fin des mots afin de ne conserver que la racine du mot.

Exemple : « trouverez » -> « trouv »

- lemmatization : cela consiste à réaliser la même tâche que stemming mais en utilisant un vocabulaire et une analyse fine de la construction des mots. La lemmatisation permet donc de supprimer uniquement les terminaisons inflexibles et donc à isoler la forme canonique du mot, connue sous le nom de lemme.

Exemple : « trouverez » -> trouvez.

- posTags : part of speech tags est un processus populaire de traitement du langage naturel qui fait référence à la catégorisation des mots dans un texte (corpus) en correspondance avec une partie particulière du discours, en fonction de la définition du mot et de son contexte.

Why	not	tell	someone	?
adverb	adverb	verb	noun	punctuation mark, sentence closer

FIGURE II.5 – Exemple de marquage POS

- named entity recognition : Reconnaissance d'entité nommée est le processus de détection des entités nommées telles que les noms de personnes, les noms de lieux, les noms de sociétés, etc.

Ousted	WeWork	founder	Adam Neumann	lists his	Manhattan	penthouse for	\$37.5 million
	[organization]		[person]		[location]		[monetary value]

FIGURE II.6 – Exemple de reconnaissance d'entité nommée

Chapitre III

Méthodologie

1 Les réseaux de neurones récurrents bidirectionnels (BRNN)

1.1 La nécessité d'une traversée bidirectionnelle

Un état typique dans un RNN (RNN, GRU ou LSTM simple) repose sur les événements passés et présents. Un état à la fois t dépend des états $x_1, x_2, \dots, x_{t-1}, x_t$. Cependant, il peut y avoir des situations où une prédiction dépend des événements passés, présents et futurs.

Par exemple, prédire qu'un mot sera inclus dans une phrase pourrait nous obliger à regarder vers l'avenir, c'est-à-dire qu'un mot dans une phrase pourrait dépendre d'un événement futur. De telles dépendances linguistiques sont habituelles dans plusieurs tâches de prédiction de texte.

Prenez la reconnaissance vocale. Lorsque vous utilisez un assistant vocal, vous prononcez d'abord quelques mots après quoi l'assistant interprète et répond. Cette interprétation peut ne pas dépendre entièrement des mots précédents ; toute la séquence de mots ne peut avoir de sens que lorsque les mots suivants sont analysés. l'application :

1.2 définition du BRNN

Les réseaux de neurones récurrents bidirectionnels (BRNN) signifient connecter deux couches cachées de directions opposées à la même sortie. Avec cette forme d'apprentissage profond génératif, la couche de sortie peut obtenir des informations des états passés et futurs en même temps. Les BRNN ont été introduits pour augmenter la quantité d'informations d'entrée sur le réseau. Également les BRNN n'ont pas besoin que leurs données d'entrée soient corrigées. De plus, leurs informations d'entrée futures sont accessibles à partir de l'état actuel. Les BRNN sont particulièrement

utiles lorsque le contexte de l'entrée est nécessaire.

1.3 Architecture de RNN bidirectionnelle

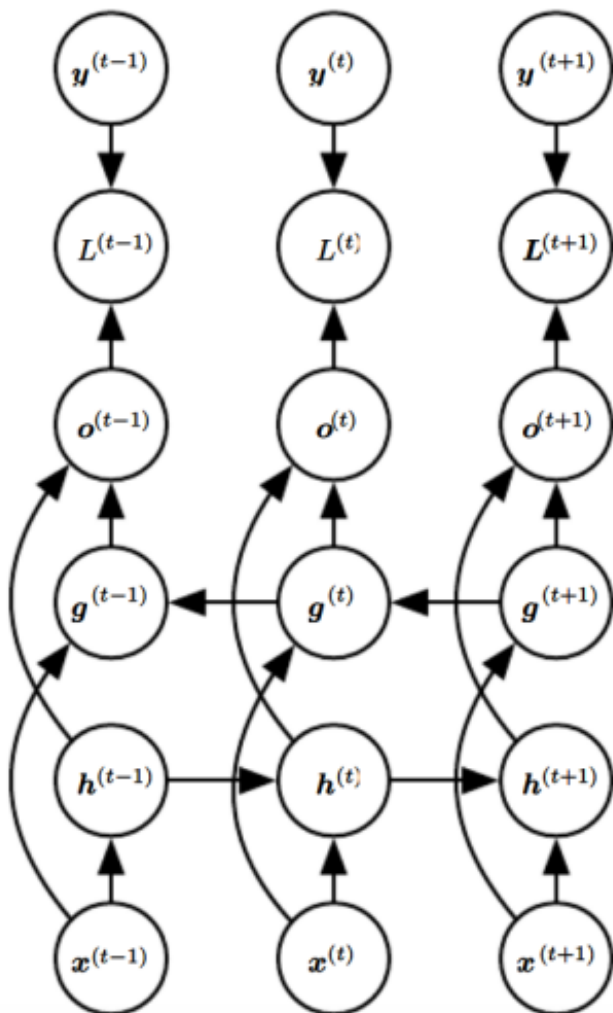


FIGURE III.1 – démonstration de RNN bidirectionnelle

Un BRNN mappe les séquences d'entrée x aux séquences cibles y avec une perte $L(t)$ à chaque étape t .

- h la récurrence se propage vers la droite
- g la récurrence se propage vers la gauche.

Cela permet aux unités de sortie $o(t)$ de calculer une représentation qui dépend à la fois du passé et l'avenir.

1.4 Le fonctionnement du BRNN

Les réseaux de neurones récurrents bidirectionnels ne font que mettre deux RNN indépendants ensemble. La séquence d'entrée est alimentée dans l'ordre temporel normal pour un réseau et dans l'ordre temporel inverse pour un autre réseau. Les sorties des deux réseaux sont généralement concaténées à chaque pas de temps, bien qu'il existe d'autres options comme la sommation. Les blocs réseau d'un BRNN peuvent être de simples RNN, GRU ou LSTM.

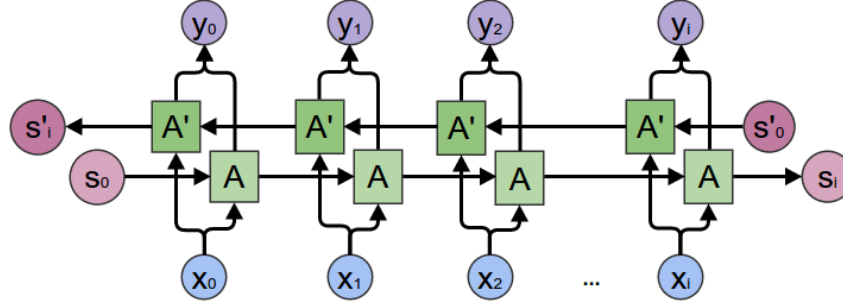


FIGURE III.2 – Structure générale des réseaux de neurones récurrents bidirectionnels

Cette structure permet au réseau d'avoir à la fois des informations en amont et en aval sur la séquence à chaque pas de temps.

Un BRNN dispose d'une couche cachée supplémentaire pour s'adapter au processus d'entraînement en amont. À tout moment t , les états masqués avant et arrière sont mis à jour comme suit :

$$A_t(Forward) = \phi(X_t * W_{XA}^{forward} + A_{t-1}(Forward) * W_{AA}^{forward} + b_A^{forward})$$

$$A_t(Backward) = \phi(X_t * W_{XA}^{backward} + A_{t+1}(Backward) * W_{AA}^{backward} + b_A^{backward})$$

où ϕ est la fonction d'activation, W , la matrice de poids, et b , le biais.

L'état caché à l'heure t est donné par une combinaison de $A_t(Forward)$ et $A_t(Backward)$. La sortie à un état caché donné est : $O_t = H_t W_A Y + b_Y$

Dans un BRNN, il existe des passes en avant et en arrière qui se produisent simultanément, la mise à jour des poids pour les deux processus peut se produire au même moment. Cela conduit à des résultats erronés. Ainsi, pour accueillir séparément les passes avant et arrière, l'algorithme suivant est utilisé pour l'entraînement d'un BRNN :

— Passe avant

États d'avant (à partir de $t = 1$ à N) et les états antérieurs (à partir de $t = N$ à 1) sont adoptés. Les valeurs des neurones de sortie sont transmises (à partir de $t = 1$ à N).

— **Passe en arrière**

Les valeurs des neurones de sortie sont transmises ($t = N$ à 1). États d'avant (à partir de $t = N$ à 1) et les états antérieurs (à partir de $t = 1$ à N) sont adoptés.

Chapitre IV

Expérimentations et résultats

1 Introduction

Nous présentons dans ce chapitre l'ensemble de déroulement de l'expérimentation que nous avons fait ,et les resultat obtenue.Dans notre experimentattion nous avons traiter des tweets ecrites en arabe standart et/ou dialecte egyptien.

Avant de detailler notre travail,nous representont dans la figure si dessous l'architecture globale de notre approche d'analyse des sentiments des tweets arabe.

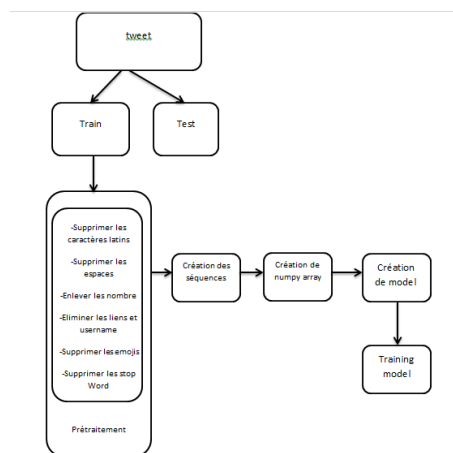


FIGURE IV.1 – La méthodologie realiser pour l'analyse des sentiments sur les tweets arabes

Comme la montre la figure 4.1 ,nous commençons par le nettoyage et la preparation de dataset,ensuite la creation des sequence puis les convertir en valeur numerique pour que la machine peut les traiter.Et enfin ,nous créons le 'model' pour executer le training.

2 Dataset

Nous avons utilisé un Dataset de tweets annoté pour l'analyse des sentiments qui composé de Tweets de sentiment negative et positive sous forme d'un dossier de 8 072 tweets a deux colonne :l'un pour les tweets et l'autre indique la polarités .On a obtenue cette Dataset afin de filtrer et concatiner plusieurs Dataset de plusieurs sources.

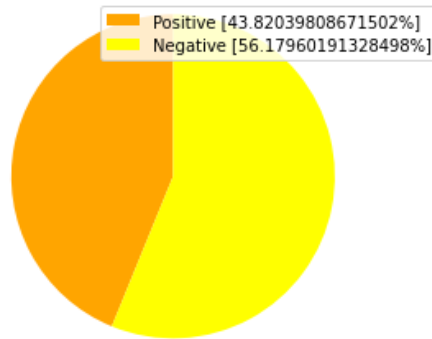


FIGURE IV.2 – la distribution des Tweets

la figure 4.2 represente la distribution des Tweets de notre Dataset qui est de 3 641 negatifs ,et 2 840 positifs

3 Outils et bibliotheques utilisées

python

Python est un langage de programmation puissant et facile a apprendre. Il dispose de structures de donnees de haut niveau et permet une approche simple mais efficace de la programmation orientee objet. Parce que sa syntaxe est elegante, que son typage est dynamique et qu'il est interprete, Python est un langage ideal pour l'ecriture de scripts et le developpement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes.

Google collab

Offert par Google on l'utilise google colab pour faciliter et economiser le materiel physique (hardware) , Cette plateforme permet d'entrainer des modeles de Machine Learning directement dans le cloud.Pour notre cas on la utilise pour compiler et executée l'ensemble de notre code.

Tensorflow

TensorFlow est un outil open source d'apprentissage automatique développé par Google. Le code source a été ouvert le 9 novembre 2015 par Google et publié sous licence Apache. Il est fondé sur l'infrastructure DistBelief, initiée par Google en 2011, et est doté d'une interface pour Python, Julia et R2. Cet outil dédié à l'apprentissage automatique est fortement utilisé dans le domaine de l'intelligence artificielle (IA). Ainsi, des professionnels comme des novices peuvent créer des modèles de machine learning ou de deep learning pour optimiser les capacités de leur matériel.

Pour faire simple, TensorFlow est une bibliothèque de Machine Learning, il s'agit d'une boîte à outils permettant de résoudre des problèmes mathématiques extrêmement complexes avec aisance. Elle permet aux chercheurs de développer des architectures d'apprentissage expérimentales et de les transformer en logiciels.

Cette bibliothèque permet notamment d'entraîner et d'exécuter des réseaux de neurones pour la classification de chiffres écrits à la main, la reconnaissance d'image, les plongements de mots, les réseaux de neurones récurrents, les modèles sequence-to-sequence pour la traduction automatique, ou encore le traitement naturel du langage.

On a fait recours à cette bibliothèque à fin d'utiliser les éléments nécessaires pour la création de modèle, tel que : GRU, Embedding, Dense, Input, Dropout, Bidirectional.

Keras

Keras est une bibliothèque open source de composants de réseaux de neuronaux écrits en Python. Elle représente un outil complémentaire très efficace en complément de TensorFlow. La bibliothèque a été développée pour être modulaire et conviviale, mais elle a d'abord commencé dans le cadre d'un projet de recherche pour le système d'exploitation intelligent neuro-electronique ouvert ou ONEIROS. L'auteur principal de Keras est François Chollet. Composée d'une bibliothèque de composants d'apprentissage automatique couramment utilisés, notamment des objectifs, des fonctions d'activation et des optimiseurs, la plate-forme open source de Keras prend également en charge les réseaux de neurones récurrents et convolutifs.

On a fait appel à cette bibliothèque à fin de vectoriser un corpus de texte, en transformant chaque texte en une séquence d'entiers, ainsi que s'assurer que toutes ces séquences ont la même taille. Cette bibliothèque nous permet aussi de tokeniser (tokenization) les tweets de dataset.

Pandas

Il permet la manipulation et l'analyse des données. Il propose en particulier des

structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles.

Nous avons utilisées cette bibiliotheque pour lire le document `csv(dataset)`.

Re

Les expressions régulières sont des schémas ou des motifs utilisés pour effectuer des recherches et des remplacements dans des chaines de caractères.

Les expressions régulières ou expressions rationnelles ne font pas partie du langage Python en soi mais constituent un langage à part. Python nous permet d'exploiter leur puissance et fournit un support pour les expressions régulières via son module standard `re`.

Nous le avons utilisées dans le pretraitement de `dataset`(les etapes de pretraitement se trouve dans le section ci-dessous).

Numpy

il est destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

Plus précisément, cette bibliothèque logicielle libre et open source fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices et polynômes.

NLTK

Natural Language Toolkit (NLTK) est une bibliothèque logicielle en Python permettant un traitement automatique des langues,Il fournit des interfaces faciles à utiliser à plus de 50 corpus et ressources lexicales telles que WordNet, ainsi qu'une suite de bibliothèques de traitement de texte pour la classification, la tokenisation... Nous avons utilisées cette bibiliotheque pour avoir la liste des stops words arabe.

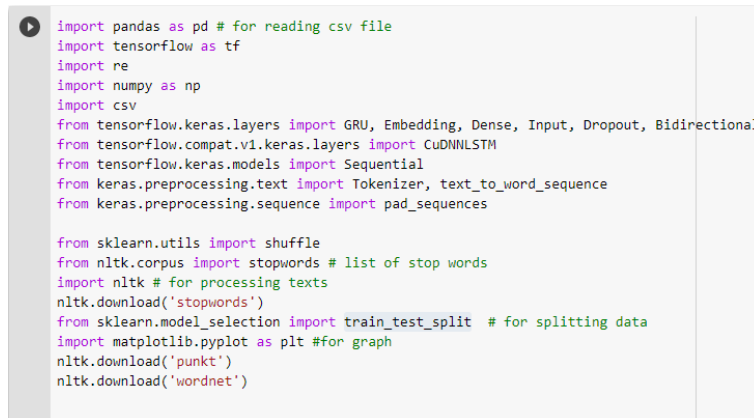
Scikit-learn

Scikit-learn est probablement la bibliothèque la plus utile pour l'apprentissage automatique en Python. La bibliothèque `sklearn` contient de nombreux outils efficaces pour l'apprentissage automatique et la modélisation statistique, notamment la classification, la régression, le regroupement et la réduction de la dimensionnalité. Cette bibliotheque nous a aidées lors de division de `dataset` :une tranche pour le

training, et l'autre pour testing.

Matplotlib

Matplotlib est une bibliothèque de tracage disponible pour le langage de programmation Python en tant que composant de NumPy, une ressource de gestion numérique du Big Data. Matplotlib utilise une API orientée objet pour incorporer des traces dans des applications Python.



```
import pandas as pd # for reading csv file
import tensorflow as tf
import re
import numpy as np
import csv
from tensorflow.keras.layers import GRU, Embedding, Dense, Input, Dropout, Bidirectional
from tensorflow.compat.v1.keras.layers import CuDNNLSTM
from tensorflow.keras.models import Sequential
from keras.preprocessing.text import Tokenizer, text_to_word_sequence
from keras.preprocessing.sequence import pad_sequences

from sklearn.utils import shuffle
from nltk.corpus import stopwords # list of stop words
import nltk # for processing texts
nltk.download('stopwords')
from sklearn.model_selection import train_test_split # for splitting data
import matplotlib.pyplot as plt #for graph
nltk.download('punkt')
nltk.download('wordnet')
```

FIGURE IV.3 – capture d'écran des bibliothèques utilisées

4 Prétraitement

En générale ,l'utilisateur de Twitter utilise des abréviations, des émoticons,des hashtag,les mention et des argots pour exprimer ses opinions et ses sentiments. Par conséquence une étape de prétraitement est indispensable.

Dans ce qui suit nous allons présenter la procédure de prétraitement suivie dans notre travail, dont le but est de nettoyer les tweets et les rendre le plus proche possible d'un langage formel

Nous avons procédé à un prétraitement qui suit les étapes suivantes :

- 1- Supprimer les caractère latin car on est entrain de traiter des tweets en arabe
- 2- Supprimer les nombre,les liens,les ponctuation puisqu'ils n'ont pas un impact sur le classement
- 3- Enlever les émoticones
- 4- Supprimer les noms des utilisateur (@user)
- 5- Supprimer les hashtag (#)
- 6- Supprimer les espaces supplémentaire

- 7- Enlever(harakat)qui interfèrent avec les manipulations informatiques avec les textes arabes.
- 8- Enlever les stop word.
- 9- Eliminer les caractères répétés que l'utilisateur l'utilise pour affirmer le sens.

Tweets avant prétraitement	Tweets après pretraitement
man live و ده لما تبقى مش عارف تلم ليلة الامتحان ف تعيش بمبدأ only once 😊 !	ده تبقى مش عارف تلم ليلة الامتحان تعيش بمبدأ
الخطاء هي عبارة عن دروس نتلقاها وكل درس يحدث لنا يجعلنا أفضل ❤️	الخطاء عبارة دروس نتلقاها وكل درس يحدث يجعلنا أفضل
عايز تكون عضو في حملة المشير السيسي سهل جداً ساعدنا اننا نعمل مليون شيبير #سأنتخب_السيسي	عايز تكون عضو حملة المشير السيسي سهل جدا ساعدنا اننا نعمل مليون شيبير

FIGURE IV.4 – Exemples des Tweets avant et après pretraitement

```

arabic_punctuations = '""÷×-“”‘’|+!~{}`,.$%:/~][%^&*()_<>:!' # define arabic punctuations
tashkeel = re.compile(r'[\u0617-\u061A\u064B-\u0652]')

def process_review(review):
    out = re.sub(r"^\w\s]", '', review)
    out = re.sub(r"[a-zA-Z]", '', out) # remove english letters
    out = re.sub(r"\n", '', out) # remove \n from text
    out = re.sub(r'\d+', '', out) # remove number
    out = re.sub(r'http\S+', '', out) # remove links
    out = re.sub(r"["
        u"\U0001F600-\U0001F64F" # emoticons
        u"\U0001F300-\U0001F5FF" # symbols & pictographs
        u"\U0001F680-\U0001F6FF" # transport & map symbols
        u"\U0001F1E0-\U0001F1FF" # flags (iOS)
        "]", '', out) #remove emojis
    out = out.translate(str.maketrans('', '', arabic_punctuations)) # remove punctuation
    #remove tashkeel
    out = re.sub(tashkeel, "", out)
    out = re.sub(r"@[\s]+[\s]?", '', out) # Remove username
    p_longation = re.compile(r'(\.)\1+')
    subst = r"\1\1"
    out = re.sub(p_longation, subst, out) # remove repetitive caractere
    out = ' '.join([word for word in out.split() if word not in stopwords.words("arabic")]) # remove stop word
    out = re.sub(r"\s+", ' ', out)
    out = re.sub(' +', ' ', out) # remove extra space
    return out.strip()

```

FIGURE IV.5 – capture d’écran de fonction créer pour le nettoyage de dataset

5 Extraction et présentation des descripteurs

Le processus du classement du texte par des modèles d’apprentissage automatique est essentiellement le même que celui utilisé pour le classement d’un autre type de données. La principale différence, est constituée par le processus de transformation de données pour que celles-ci puissent être passées à l’algorithme de classement comme une représentation vectorielle numérique.

Dans cette transformation, il est nécessaire de passer les données du texte pur à une représentation dans laquelle les documents de texte sont numériquement représentés dans une matrice que le classifieur peut interpréter. On s’est basé sur la description de tâches du processus du classement illustré par la figure 4.6, on explique ci-dessous les différentes étapes de transformation du texte.

5.1 L’extraction de termes (Tokenization)

La tokenisation est la première étape de l’analyse de texte. c’est processus de décomposition d’un paragraphe de texte en petits morceaux tels que des mots ou des phrases. Le jeton est une entité unique qui constitue les blocs de construction d’une phrase ou d’un paragraphe. Ces jetons aident à comprendre le contexte ou à

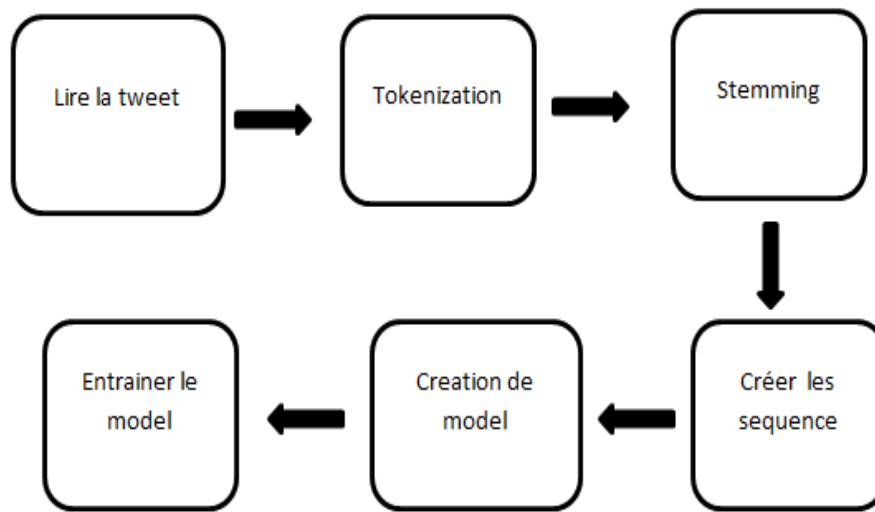


FIGURE IV.6 – Processus de prétraitement et transformation du texte.

développer le modèle du PNL. La tokenisation aide à interpréter le sens du texte en analysant la séquence des mots.

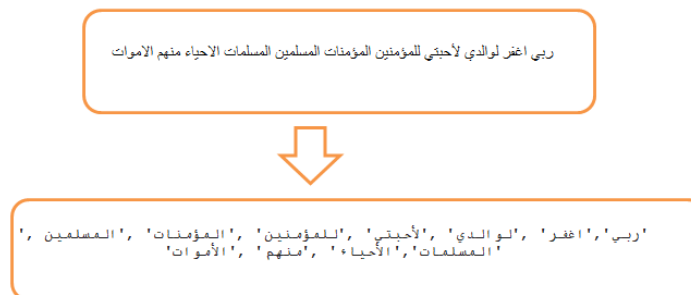


FIGURE IV.7 – exemple de tokenization d'un tweet de notre dataset

5.2 stemming ou réduction à la tige

Cette étape consiste à rendre l'ensemble des mots, qui ont déjà subi la tokenization, à son origine on utilise la fonction `stemmer.stem()`. Il facilite le prétraitement des données en normalisant tout le vocabulaire.

5.3 création des séquences ou présentation vectorielle

Le texte original peut être vu comme une séquence de mots. Ce type de représentation est actuellement incompréhensible pour les algorithmes d'apprentissage.



FIGURE IV.8 – exemple de stemming d’un tweet qui a déjà subit tokenization de notre dataset

automatique qui ont besoin de recevoir des représentations vectorielles numériques des entités à classer. La représentation vectorielle consiste à transformer chaque document en une séquence de nombres, dans laquelle chaque nombre correspond à un mot du vocabulaire de l’ensemble des documents ou corpus.

Pour transformer les documents de texte en vecteurs, on a utiliser la fonction `texts to sequences()`, et la fonction `pad sequences` pour assurer que toutes les entités on la meme longueur.

```
[ ] #making sequences
X = tknzsr.texts_to_sequences(data['text'])
X = pad_sequences(X, padding='post', value=0)
```

FIGURE IV.9 – capture d’écrans de fonction utiliser pour la presentation vectoriel

5.4 creation de model

Notre modele utiliser lors du projet est RNN .sa structure est composée de :1 couche Embedding,1 couche Bidirectional,2 couche dense et 1 couche Dropout. 294,209 paramètres au total doivent être entraînés.

Comme premiere etape de creation de reaseau de neurone on va choisir quelle type de modele on va creer(sequential ou functional). Pour notre cas on a choisi un model sequenciel pour permettre a chaque couche d’admet un poids et une connexion a la couche située juste apres dans notre diagramme.

la premier couche a creer est "Embdding layer",qui est est l’une des couches disponibles dans Keras.

Embedding layer prend comme parametre la taille des vocabulaires(dans notre cas c'est la longueur `tokenizer.word-index` qui vaut 8732) et la Longueur du vecteur pour chaque mot(pour ce cas c'est 32).

De manière générale, nous utilisons l'embedding layer pour compresser l'espace des caractéristiques d'entrée en un espace plus petit.

La deuxième chose à ajouter à notre réseau de neurones est bidirectionnel. Il permet de connecter deux couches cachées de directions opposées à la même sortie. la couche utilisée comme parametre de bidirectionnel fonction est GRU(Gated Recurrent Unit). GRU permet de conserver juste les mots qui permet de prédire le type de sentiment de tweets.

La troisième chose à ajouter est "Dense", est une couche profondément connectée à sa couche précédente, ce qui signifie que les neurones de la couche sont connectés à chaque neurone de sa couche précédente. Le neurone de cette couche reçoit la sortie de chaque neurone de sa couche précédente, où les neurones de la couche dense effectuent une multiplication matrice-vecteur.

la couche Dense prend comme parametre 'units' et 'activation function'. Les unités définit la taille de la sortie de la couche dense. Il doit s'agir d'un entier positif puisqu'il représente la dimensionnalité du vecteur de sortie, dans notre modèle on a choisie 32 comme taille de la sortie.

le deuxième parametre est la fonction d'activation qui est utilisée pour la transformation des valeurs d'entrée des neurones. Fondamentalement, il introduit la non-linéarité dans les réseaux de neurones afin que les réseaux puissent apprendre la relation entre les valeurs d'entrée et de sortie. la fonction d'activation utilisé dans notre modèle est 'tanh', sa formule est :

$$\tanh(x) = (\exp(x) - \exp(-x)) / (\exp(x) + \exp(-x))$$

Cette fonction est comme Sigmoid, utilisé dans la classification binaire. Par exemple pour notre classification des tweets arabes, plus la valeur retournée par tanh est proche de 1 plus le modèle considère que le tweet est positif, plus elle est proche de -1, plus elle est considérée comme négative.

la quatrième fonctionnalité à ajouter dans ce réseau est 'Dropout'. Le Dropout est une technique permettant de réduire l'overfitting lors de l'entraînement du modèle. il fait référence à la suppression de neurones dans les couches d'un modèle. En fait, on désactive temporairement certains neurones dans le réseau, ainsi que toutes ses connexions entrantes et sortantes, Le choix des neurones à désactiver est aléa-

toire.À chaque epoch, on applique cette désactivation aléatoire. C'est-à-dire qu'à chaque passe (forward propagation) le modèle apprendra avec une configuration de neurones différentes, les neurones s'activant et se désactivant aléatoirement.

le Dropout prend comme parametre un nombre entre 0 et 1 qui design la probabillite de desactivation de chaque neurones(0,3 pour notre cas).

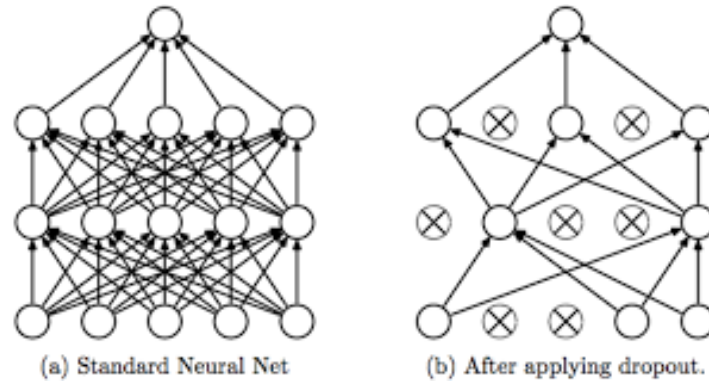


FIGURE IV.10 – exemple de Réseau de Neurones Standard avant et après l'application de Dropout

La dernier couche a ajouter est une couche Dense avec sigmoide comme fonction d'activation ,sa formule mathématique est :

$$\text{sigmoid}(x) = 1/(1 + \exp(-x))$$

Ci-dessous, capture d'écrans de réseau de neurone créer. Après la creation de toutes les couche necssaires, on fait compiler le model a l'aide de 'model.compile'. Model compile permet de choisir 'optimizer' et 'loss function'. Dans notre cas on doit clasifier les tweets en deux categorie(positif ou negative) c'est pour cela on a utiliser 'binary-crossentropy' comme loss function ,puisque cette dernier est utiliser lors de la clasfication binaire.Et 'adam' comme optimizer puisque les résultats de l'optimiseur Adam sont généralement meilleurs que tous les autres algorithmes d'optimisation, ont un temps de calcul plus rapide et nécessitent moins de paramètres pour le réglage.

```
model = Sequential() #creer un sequence
model.add(Embedding(len(tknzr.word_index), 32))
model.add(Bidirectional(GRU(units = 32)))
model.add(Dense(32, activation = 'tanh'))
model.add(Dropout(0.3))
model.add(Dense(1, activation = 'sigmoid'))
model.compile(optimizer = 'Adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
```

FIGURE IV.11 – capture d'écran de model RNN réalisé

Voici une capture d'écran de summary de model creer .il indique la force de la relation entre le modèle et la variable dépendante. Chaque couche a une sortie et sa forme est indiquée dans la colonne "Output shape". La sortie de chaque couche devient l'entrée de la couche suivante. La colonne "Param #" vous montre le nombre de paramètres qui sont formés pour chaque couche.

model.summary()

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, None, 32)	281248
bidirectional (Bidirectional)	(None, 64)	12672
dense (Dense)	(None, 32)	2080
dropout (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 1)	33

=====

Total params: 296,033
Trainable params: 296,033
Non-trainable params: 0

FIGURE IV.12 – capture d'écrans de summary de model RNN

5.5 Entraîner le model

Avant d'entraîner le model, on a diviser dataset on deux :25% pour le test et 75% pour l'entraînement. Avec cette fonction, nous n'avons pas besoin de diviser l'ensemble de données manuellement.

```
[ ] # Split data into training and testing
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size =0.25, random_state=42)
print("Training:", len(X_train))
print("Testing: ", len(X_test))
```

FIGURE IV.13 – capture d'écrans de fonction de division de dataset

Après avoir diviser dataset ,on va entraîner le model a l'aide de la fonction 'model.fit'. Il prend comme parametre les donnés d'entrée(x-train), données ciblées(y-train), epochs : qui est nombre d'iteration sur l'ensemble x-train et y-train fournies,

validation-data : ce sont les données sur les quelles on va évaluer la perte et toute métrique de modèle à la fin de chaque époque, batch size : c'est le nombre d'échantillons par mise à jour du gradient.

Notre entraînement a donné 0,78 qui est une valeur moyennement bonne ,comme valeur de testing accuracy.

Ci-dessous,un graph qui illustre le deroulement de training

```
history = model.fit(X_train, y_train, epochs=4, verbose=True, validation_data=(X_test, y_test), batch_size=64)
loss, accuracy = model.evaluate(X_train, y_train, verbose=True)
print("Training Accuracy: {:.4f}".format(accuracy))
loss_val, accuracy_val = model.evaluate(X_test, y_test, verbose=True)
print("Testing Accuracy: {:.4f}".format(accuracy_val))
```



```
Epoch 1/4
76/76 [=====] - 8s 22ms/step - loss: 0.6406 - accuracy: 0.6200 - val_loss: 0.4932 - val_accuracy: 0.7631
Epoch 2/4
76/76 [=====] - 1s 13ms/step - loss: 0.3544 - accuracy: 0.8514 - val_loss: 0.4360 - val_accuracy: 0.7896
Epoch 3/4
76/76 [=====] - 1s 14ms/step - loss: 0.1946 - accuracy: 0.9335 - val_loss: 0.5002 - val_accuracy: 0.7964
Epoch 4/4
76/76 [=====] - 1s 13ms/step - loss: 0.1125 - accuracy: 0.9644 - val_loss: 0.6088 - val_accuracy: 0.7890
152/152 [=====] - 1s 5ms/step - loss: 0.0622 - accuracy: 0.9842
Training Accuracy: 0.9842
51/51 [=====] - 0s 5ms/step - loss: 0.6088 - accuracy: 0.7890
Testing Accuracy: 0.7890
```

FIGURE IV.14 – capture d'écrans de training de model

6 Résultat obtenue par le model

Comme indiqué dans la section precedente ,l'entraînement d'algorithme avec la dataset proposé a donné 0,78 comme taux d'accuracy ,ainsi on a calculer le taux de precision,recall et f1-score :

- La précision est calculée comme le rapport entre le nombre d' échantillons positifs correctement classés et le nombre total d'échantillons classés comme positifs (correctement ou incorrectement). La précision mesure l'exactitude du modèle à classer un échantillon comme positif.

- Le rappel est calculé comme le rapport entre le nombre d' échantillons positifs correctement classés comme positifs et le nombre total d' échantillons positifs . Le rappel mesure la capacité du modèle à détecter les échantillons positifs . Plus le rappel est élevé, plus les échantillons positifs détectés sont nombreux.

- Le f1-score combine la précision et le rappel d'un classificateur en une seule métrique en prenant leur moyenne harmonique. Il est principalement utilisé pour comparer les performances de deux classificateurs.

Ci-dessous,voici les resultat de precision ,recall et f1-score obtenu de notre model.

```
[ ] from sklearn.metrics import classification_report, confusion_matrix
    target_names=["negative","positive"]
    print(classification_report(y_test, y_pred, target_names=target_names))
```

	precision	recall	f1-score	support
negative	0.80	0.82	0.81	894
positive	0.77	0.75	0.76	727
accuracy			0.79	1621
macro avg	0.79	0.79	0.79	1621
weighted avg	0.79	0.79	0.79	1621

FIGURE IV.15 – valeur de precision,recall et f1-score obtenu de notre model

La visualisation des données est l'un des meilleurs moyens d'humaniser les données pour faciliter leur compréhension et en tirer les tendances pertinentes. Cette activité peut être cruciale lorsque l'utilisateur essaie encore d'optimiser le modèle et de le rendre prêt pour la production.c'est pour cela on a essayer de visualiser notre graph afin de savoir si l'entrainement et le test on bien etait passé.

Comme vous voyez sur le graph ci-dessous ,on a obtenue des bonnes résultat en ceux qui concerne le training et testing.

D'après la Figure IV.16, L'accuracy de l'apprentissage augmente avec le nombre d'époque, ceci reflète qu'à chaque époque le modèle apprenne plus d'informations.Par contre validation loss dans la figure V.17 elle est presque stable entre [0.3 .06].

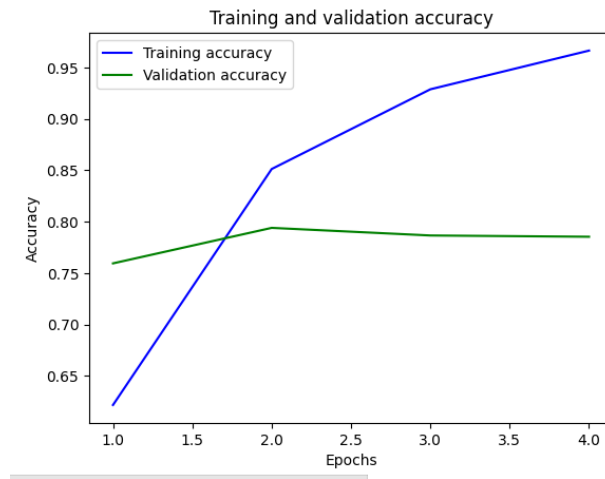


FIGURE IV.16 – graph illustre les résultats moyens de performance des classifieurs BRNN pour training et validation accuracy



FIGURE IV.17 – graph illustre les résultats moyens de performance des classifieurs BRNN pour trainnig et validation loss

7 Comparaison et discussion

Résultat de l'approche proposée par rapport à d'autre approches existantes).

classificateur	precisions		recall		f1-score		accuracy
	pos	neg	pos	neg	pos	neg	
Multilayer Perceptrons (mlp)	0.92	0.96	0.99	0.73	0.96	0.83	0.93
CNNs et LSTM	0.96	0.95	0.966	0.94	0.95	0.96	0.94
approche réalisé(BRNN)	0.77	0.80	0.75	0.82	0.76	0.81	0,79

Dans le tableau ci-dessus, en a fait la comparaissent entre 3 modèles (mlp,CNN et LSTM, BRNN). D'après ce tableau nous remarquons que le model CNN LSTM nous donne des meilleurs résultats de test par apport au BRNN.

Notre approche proposée est limitée par l'utilisation de données de binaire classification .Le texte peut être très bien traité par le deep learning, mais les méthodes d'apprentissage en profondeur nécessitent souvent une trop grande quantité de données pour la formation, ce qui est pas disponible. Peut-être conviendrait-il d'explorer à l'avenir les possibilités de apprendre le modèle à partir de données mixtes textuelles et non textuelles. Une autre façon peut être d'utiliser méta-apprentissage ou apprentissage par un ensemble de méthodes différentes. Une autre limite de notre approche est lié au prétraitement. Le prétraitement des données textuelles doit être simple pour éviter perte d'informations spécifiques typiques des trolls (erreurs intentionnelles, majuscules, mots d'argot,dictionnaire abusif, etc.). D'autre part, les données de structure et d'utilisation ont besoin de plus pré-traitement sophistiqué.

Chapitre V

Conclusion et perspective

1 Perspective

Réseau de neurones artificiels et réseau de neurones convolutifs principalement utilisés pour le traitement d'images. Dans ce projet, nous utilisons le réseau de neurones récurrent qui est principalement utilisé pour les tâches de traitement du langage naturel, donc si vous pensez à l'apprentissage en profondeur dans son ensemble, les CNN sont principalement pour les images, les RNN sont principalement pour le NLP. Il existe également d'autres cas d'utilisation, nous comprendrons donc le fonctionnement du réseau de neurones récurrent et nous examinerons différentes applications de RNN dans le domaine de la NLP ainsi que dans d'autres domaines. Nous examinerons certains cas d'utilisation réels où les modèles de séquence sont utiles. Vous devez avoir utilisé Google mail-Gmail. Ici, lorsque vous tapez une phrase, elle se complète automatiquement. Alors voyez, quand je tape "pas intéressé pour le moment", c'est quelque chose qui s'est automatiquement terminé. Ainsi, Google a intégré ce RRN ou réseau neuronal récurrent dans lequel, lorsque vous tapez une phrase "pas intéressé par", il se complétera automatiquement par "cette fois". Si vous dites "nous vous informerons si cela change", il sera également indiqué "à l'avenir", ce qui vous fera gagner du temps. Il écrira la phrase pour vous. Un autre cas d'utilisation est la traduction. Vous devez avoir utilisé Google Translate où vous pouvez facilement traduire une phrase d'une langue à une autre. Le troisième cas d'utilisation est la reconnaissance d'entité nommée où, dans le X, vous savez que vous donnez une déclaration au réseau de neurones et dans le réseau de neurones Y, vous indiquera le nom de la personne, l'entreprise et l'heure. "Rudolph Smith doit être millionnaire avec les prix de Tesla qui montent en flèche". Il s'agit donc de divers cas d'utilisation où l'utilisation de modèles de séquence ou de réseaux de neurones récurrents RNN est utile. Le quatrième cas d'utilisation est l'analyse des sentiments où vous avez un paragraphe et il vous dira le sentiment si cette critique de produit est une étoile, deux étoiles et ainsi de suite. Maintenant, vous penseriez - Pourquoi ne pouvons-nous

pas utiliser un simple réseau de neurones pour résoudre ce problème ? Voir tous ces problèmes, ils sont appelés problème de modélisation de séquence car la suite est importante. En ce qui concerne le langage humain, la séquence est très importante. Par exemple, lorsque vous dites "comment allez-vous ?" versus "tu es comment" n'a pas de sens, n'est-ce pas ? Donc, la séquence est importante ici et vous penseriez - Pourquoi n'utilisons-nous pas un simple réseau de neurones pour cela ? Eh bien, essayons. Donc, pour la traduction linguistique, que diriez-vous de construire ce type de réseau de neurones, nous savons où l'entrée est la déclaration en anglais et la sortie pourrait être la déclaration en arabe. Une fois que j'ai construit ce réseau, que se passe-t-il si la taille de ma phrase change ? Donc, je pourrais entrer une taille de phrase différente et avec une architecture de réseau neuronal fixe, cela ne fonctionnera pas car vous devez décider du nombre de neurones dans la couche d'entrée et de sortie. Ainsi, avec la traduction linguistique, le nombre de neurones devient un problème. Comme quoi décidez-vous comme taille de neurones ? Maintenant, on pourrait dire d'accord, je déciderais, disons, d'une taille énorme, disons 100 neurones et le reste si je dis, avez-vous mangé du riz ? Il occupera donc 4 neurones. Restant 96, je dirai simplement 0 ou vous savez déclaration vide. Cela pourrait fonctionner mais ce n'est toujours pas idéal. Le deuxième problème est trop de calcul. Vous savez tous que les réseaux de neurones fonctionnent sur des nombres, ils ne fonctionnent pas sur des chaînes. Vous devez donc convertir votre mot en vecteur. Donc, l'une des façons de convertir cela en un vecteur. L'utilisation d'un réseau neuronal simple pose davantage de problèmes, de sorte que nous choisissons de travailler avec le réseau neuronal récurrent (RNN). Ensuite, l'analyse sentimentale est confrontée à plusieurs défis qui sont ces manières subtiles d'exprimer des états privés comme si on a un exemple ou il n'y a aucun mots négatifs mais le sens est totalement négatif. Un autre cas par exemple quand on ne peut pas faire la différence entre cela étant un Fait ou une opinion ? Ainsi, lorsque nous sommes ironiques, nous ne le comprenons que par le ton de la voix, donc s'il est écrit, l'algorithme serait confus, ce qui entraînerait une diminution de la précision. Enfin et surtout, l'utilisation de la langue informelle la majorité de la langue utilisée sur certaines plateformes sociales, en particulier sur Twitter, s'écarte de l'arabe standard et utilisera d'autres dialectes qui diffèrent totalement de l'arabe standard.

2 Conclusion générale :

L'analyse des sentiments a de nombreuses applications tendances dans divers domaines. En affaires, il permet aux entreprises de recueillir automatiquement les avis de leurs clients sur leurs produits ou services. En politique, cela peut aider à déduire l'orientation et la réaction du public envers les événements politiques, ce qui aide

à la prise de décision. L'analyse des sentiments peut être effectuée à plusieurs niveaux, au niveau du document, au niveau de la phrase et au niveau du sujet. Dans ce travail, nous sommes intéressé par l'analyse des sentiments au niveau de la phrase des tweets arabes, pour déterminer la polarité du tweet, s'il est positif, négatif ou neutre. Malgré l'importance de l'analyse des sentiments, la recherche en langue arabe est s'améliore à un rythme un peu lent. La langue arabe a une nature complexe, en raison de son ambiguïté et de son système morphologique riche. Cette nature ainsi que le manque de ses ressources et les différents dialectes imposent des défis au avancées dans la recherche sur l'analyse des sentiments arabes. Dans ce rapport, nous avons montré les résultats de l'utilisation d'un modèle d'apprentissage en profondeur sur la performance de l'analyse des sentiments de tweets arabes. Notre modèle repose uniquement sur une représentation vectorielle de mots pré-entraînés. Malgré la complexité de la langue arabe et la simplicité du système utilisé, le modèle améliore considérablement le score et la précision par rapport aux modèles existants. En tant que travaux futurs, nous envisagerons d'utiliser différentes architectures CNN et d'essayer des ensembles différents et plus complexes. modèles pour améliorer les résultats. De plus, nous vérifierons le modèle sur de plus grands ensembles de données autres .

Bibographie

- <https://www.lebigdata.fr/tensorflow-definition-tout-savoir>(consulté le 15/04)
- <https://www.tensorflow.org/>(consulté le 26/04)
- <https://medium.com/analytics-vidhya/understanding-embedding-layer-in-keras-bbe3ff1327ce>(consulté le 6/05)
- <https://gdcoder.com/what-is-an-embedding-layer/>(consulté le 01/02)
- <https://keras.io/>(consulté le 6/03)
- <https://penseeartificielle.fr/comprendre-lstm-gru-fonctionnement-schema/>(consulté le 15/05)
- <https://analyticsindiamag.com/a-complete-understanding-of-dense-layers-in-neural-networks/>(consulté le 5/03)
- <https://inside-machinelearning.com/fonction-dactivation-comment-ca-marche-une-explication-simple/tanh>(consulté le 25/04)
- <https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/>(consulté le 15/04)
- <https://inside-machinelearning.com/le-dropout-cest-quoi-deep-learning-explication-rapide/>(consulté le 13/04)
- <https://machinelearningmastery.com/visualize-deep-learning-neural-network-model-keras/>(consulté le 8/04)

- <http://wps.fep.up.pt/wps/wp489.pdf>(consulté le 20/02)
- [http://www2.agroparistech.fr/ufr-info/membres/cornuejols/Teaching/Master-AIC/PROJETS-M2-AIC/PROJETS-2016-2017/analyse-de-sentiments\(lambert-bellard-lorre-kouki\).pdf](http://www2.agroparistech.fr/ufr-info/membres/cornuejols/Teaching/Master-AIC/PROJETS-M2-AIC/PROJETS-2016-2017/analyse-de-sentiments(lambert-bellard-lorre-kouki).pdf)(consulté le 20/02)
- <https://www.math.univ-toulouse.fr/besse/Wikistat/pdf/st-m-app-rn.pdf>(consulté le 5/04)
- <https://www.ionos.fr/digitalguide/web-marketing/search-engine-marketing/quest-ce-quun-reseau-neuronal-artificiel/>(consulté le 5/04)
- <https://github.com/Mohabyoussef09/Arabic-Sentiment-Analysis/blob/master/Data/DS3.csv>(consulté le 4/04)
- <https://github.com/iamaziz/ar-embeddings/blob/master/datasets/tweets/ArTwitter.csv>(consulté le 4/04)
- https://github.com/zakaria-aabbou/arabic_tweet_sentiment_analysis/tree/main/SentimentAnalysisArabic/data(consulte le 4/04)
- https://github.com/motazsaad/arabic_tweet_sentiment_analysis/tree/main/master(consulté le 6/04)
- <https://www.bmc.com/blogs/nlu-vs-nlp-natural-language-understanding-processing/>(consulté le 15/04)
- https://machinelearningknowledge.ai/keras-tokenizer-tutorial-with-examples-for-fit_on_texts-texts_to_sequences-texts_to_matrix-sequences_to_matrix/(consulté le 10/04)
- https://raw.githubusercontent.com/ZarahShibli/sentiment_analysis/master/data/bbn_(le 6/04)
- <http://www.mohamedaly.info/datasets/astd>(consulté le 12/04)

- <https://towardsdatascience.com/a-beginners-guide-on-sentiment-analysis-with-rnn-9e100627c02e>(consulté le 26/03)
- <https://www.i2tutorials.com/what-is-the-difference-between-bidirectional-rnn-and-rnn/>
- <https://cedar.buffalo.edu/~srihari/CSE676/10.3>
- <https://blog.paperspace.com/bidirectional-rnn-keras/>