RépubliqueAlgerienne Démocratique et populaire

Ministére de l'enseignement superieur et recherche scientifique

Université Saad Dahlab Blida1

Faculté des sciences

Département d'informatique

Mémoire Présenté par :

- MOUHOUBI Azzeddine Mounir
- GHEFFARI Mohamed Abdelfattah

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique

Filière : Informatique

Spécialité : Informatique

Option : Traitement Automatique de la langue

Sujet :

Analyse de sentiments dans la langue arabe en utilisant différentes d'approches

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

-- Pr. Benblidia	Université de Blida 1	Présidente
-- Dr. Lahiani	Université de Blida 1	Examinatrice
-- Dr. ALIANE	CERIST	Encadrante

Année Universitaire 2019-2020

Dédicace

A ma chère mère pour son soutien moral,

A mon cher père que j'aurai aimé qu'il soit là pour voir le fruit de tout ce qu'il a fait pour moi.

A mon frère Rabah et ma sœur Chahinez

A mon neveu et ma nièce adorés Rayane et Ines

Encouragements et sacrifices

A ma chère famille

Pour votre affection et tendresse

A mon binôme Abdelfattah pour sa bonne collaboration

A tous mes amis

Au bonheur des plus chers

Je dédie cet humble travail

MOUHOUBI Azzeddine Mounir

Dédicace

A mes très chers parents

Pour leur soutien moral

A mes frère Adel et houssam et mes sœurs walha, imane et meriem

A mon binôme Mounir pour sa bonne collaboration

A tous mes amis

Au bonheur des plus chers

qui auraient voulu partager ma joie ...

Je dédie cet humble travail

GHEFFARI Mohamed Abdelfattah

Remerciements

On tient à remercier dieu tout puissant de nous avoir permis de mener à bien notre mission

On remercie également notre encadrante Madame ALIANE Hassina pour l'aide et les conseils concernant notre projet de fin d'étude, qu'elle nous a apporté lors des différentes suivies.

Nos remerciements à tous les membres des jurys qui nous ont fait l'honneur d'examiner notre travail.

Nous souhaitons finalement remercier nos famille, nos amis pour leur soutien physique et moral et pour leur patience dont ils ont fait preuve tout au long de cette période

Résumé

Aujourd'hui, la fouille des textes a une grande importance surtout dans des domaines sensibles comme la politique, les réseaux sociaux ont une grande importance dans tout ça, car ils contiennent pleins de textes sur des sujets divers c'est pour cela que l'analyse de sentiment est très importante parce que grâce à elle on obtient une bonne gestion des opinions et on aura l'opinion publique en un temps record.

Dans notre travail, nous allons essayer de classifier des opinions à l'aide d'un Dataset fournis par le centre de recherche CERIST en deux classes (Positif/Negatif) pour ce faire on a utilisé deux approches apprentissage automatique ou on a utilisé différents algorithmes comme la régression logistique, du côté apprentissage profond on a utilisé le Simple RNN , LSTM et LSTM Bidirectionnel . Ce qui nous mène à comparer ces différentes approches de classification.

Mots clés : Analyse des réseaux sociaux, Analyse des sentiments, Langue Arabe, Apprentissage automatique, Apprentissage profond.

Abstract

These days text analysis has a major importance in sensitive domains such as politics, Social media plays a big role to play in this, it is filled with texts and quotes on various subjects and text analysis provides us with a good prospective and insights on the public opinion in a record of time.

The following we are going to classify various opinion using dataset provided by the research center CERIST into two classes (positive/negative). We have used two approaches first one is automatic learning and different algorithm like the logistic regression and for second one who is the deep learning we used simple RNN , LSTM & Bidirectional LSTM. Which allows us to compare those different approaches of classification.

Keywords: Social media analysis, Sentiment analysis, Arabic language, Machine learning, Deep learning.

ملخص

اليوم ، يعد التنقيب عن النص ذو أهمية كبيرة خاصة في المجالات الحساسة مثل السياسة ، والشبكات الاجتماعية لها أهمية كبيرة في كل ذلك ، لأنها تحتوي على الكثير من النصوص حول مواضيع مختلفة وهذا هو سبب تحليل المشاعر مهم جدًا لأن بفضل ذلك نتحصل على إدارة الجيدة للآراء و كذا على الرأي العام في وقت قياسي.

في عملنا هذا، سنحاول تصنيف الآراء باستخدام مجموعة البيانات المقدمة من مركز أبحاث CERIST إلى فئتين (إيجابي / سلبي) للقيام بذلك ، استخدمنا طريقتين للتعلم الآلي أين استخدمنا خوارزميات مختلفة مثل la régression logistique أما في والتعلم العميق، استخدمنا Simple RNN و LSTM و LSTM ثنائي الاتجاه. مما يقودنا إلى مقارنة مناهج التصنيف المختلفة هذه.

الكلمات المفتاحية: تحليل الوسائط الاجتماعية، تحليل المشاعر ، اللغة العربية ، التعلم الآلي ، التعلم العميق.

Table des matières

Chapitre IV Test et validation de la solution

Liste des figures

Liste des Tableaux

Liste des abréviations

Abréviation	Description
TAL	Traitement Automatique De La Langue
ML	Machine Learning (Apprentissage Automatique)
DL	Deep Learning (Apprentissage Profond
MLP	Multi Layer Perceptron
SVM	Support Vector Machine (Machine A Vecteur de Support)
LSVC	Linear Support Vector Machine Classifier
LR	Logistic Regression(Régression Logistique)
MNB	Multinomial Naive Bayes
MSA	Marocain Sentiment Analysis
SA	Sentiment Analysis
OM	Opinion Mining
TF	Term Frequency
CBOW	Continuous Bag Of Words
IDF	Inverse Document Frequency
BNB	Bernoulli Naive Bayes
KNN	K-Nearest Neighbors(K-Plus-Porches-Voisins)
CNN	Convolution Neural Network
RNN	Recurrent Neural Network (Réseau De Neurones Récurrent)
LSTM	Long Short-Term Memory
B-LSTM	Bidirectional Long Short-Term Memory
UI	user interface
API	application programming interface

Introduction Générale

Internet est devenu un outil incontournable que ce soit dans le domaine professionnel ou dans la vie de tous les jours, notamment avec la croissance rapide et la généralisation des réseaux sociaux.

La popularité des médias sociaux est liée à la demande d'information qui est devenue plus importante dans notre société. En général, les gens s'expriment et aiment aussi regarder les réactions et les interactions des autres comme les opinions par exemple avant de prendre n'importe quelle décision. Cependant le grand volume d'information générée sur ces médias sociaux nécessite des outils adéquats pour être traitée et analysée.

L'analyse de sentiments ou fouille d'opinion est un domaine émergent du TALN qui a pour but d'analyser les commentaires des utilisateurs sur les réseaux sociaux pour la prise de décision et ce dans différents domaines : politique, marketing, santé, éducation... .

Nous avons expérimenté plusieurs algorithmes parmi dont les KNN, SVM et LR du coté Machine Learning et Simple RNN , LSTM et LSTM bidirectionnel du coté Deep Learning , les résultats obtenus coté Deep Learning sont les meilleurs plus précisément le LSTM Bidirectionnel.

Problématique

L'analyse de sentiment a pour but d'analyser les commentaires des utilisateurs afin d'avoir leurs opinions mais le grand volume d'information générée sur ces médias sociaux nécessite des outils adéquats pour être traitée et analysée.

Objectif

Pour résoudre ce problème nous allons réaliser plusieurs modèles capable d'analyser et d'extraire l'opinion d'un commentaire (Positif/Négatif) .

Afin d'atteindre cette objectif nous allons suivre la structure suivante :

Le Premier Chapitre : nous allons parler des notions générales dans le domaine comme les réseaux sociaux, l'analyse de sentiment et ensuite nous présenterons les travaux qui ont déjà été fait dans le domaine .

Le Deuxième Chapitre : parlera des différentes approches et méthodes de classification du coté machine learning et deep learning ainsi les techniques utilisé dans chaque approche **.**

Le Troisième Chapitre : est dédier à l'explication de notre architecture et ces différentes phases .

Le Quatrième Chapitre : enfin pour le dernier chapitre on fera l'évaluation de nos modèles et nous présenteront le meilleur modèle.

Chapitre I
Analyse des sentiments dans les réseaux sociaux

I.1. Introduction

Les réseaux sociaux en ligne représentent les sites Internet et applications mobiles qui permettent aux utilisateurs de se constituer un réseau d'amis ou de relations, et qui favorisent et facilitent les interactions sociales entre individus, groupes d'individus ou organisations. Il existe également des systèmes de communication indépendants tels que Skype, Yahoo ou Messenger. [1].

L'analyse des sentiments aussi appelée opinion mining est le domaine d'étude qui étudie les opinions, les sentiments, les attitudes et les émotions des gens. Il est la tendance des domaines de recherche dans le traitement du langage naturel et il est aussi considérablement étudié dans le data mining. L'importance croissante de l'analyse des sentiments coïncide avec la croissance des médias sociaux tels que les forums de discussion, les blogs, les micro-blogs, Twitter et les réseaux sociaux. C'est une première dans l'histoire humaine que nous avons une Base de données d'opinion considérable sous forme numérique afin d'effectuer l'analyse.

Dans ce chapitre nous allons parler de ce que c'est les réseaux sociaux et l'analyse de sentiment ainsi que les différente approches qu'il existe.

enfin nous allons présenter quelques travaux qui ont été réalisé dans les réseaux sociaux et donner notre point de vue sur .

I.2. Caractéristiques communes des réseaux sociaux

Tous les réseaux sociaux ont des caractéristique pour notre part nous avant choisie les plus communs parmi eux on site :

- Compte et profil utilisateur : tous les utilisateurs on leurs information personnalisé .

- Moyen de recherche parmi les utilisateurs : les utilisateurs peuvent se chercher entre eux .

- Moyen de mise en communication : les utilisateur peuvent communiqué entre eux dans un canal privé .

- Moyen de partage et de diffusion des données : les utilisateurs peuvent publié ou donnée leur avis sur divers informations (c'est celui-là qui nous intéresse le plus) .

I.3. Réseaux sociaux les plus connus

On pose toujours la question : quel est le réseau social le plus utilisé à l'échelle mondiale? Afin d'y répondre, nous devons disposer d'un outil informatique qui classe ces réseaux selon

des critères prédéfinies, Alexa se munit de l'un de ces outils. Cette dernière est une société Internet qui analyse le trafic des autres sites avec des outils et des algorithmes qui lui sont propres. Une fois ces analyses faites, elle établit un classement des sites les uns par rapport aux autres. Ce classement est plus ou moins contesté, cependant il est un indicateur utile. Il est possible que cette société améliore le classement de son site, en échange de quelques transformations. De ce fait, il suffit de se polariser sur trois aspects-clés : le trafic, l'optimisation et la visibilité de votre site. Selon les réseaux sociaux classifiés selon Heure quotidienne sur le site et les Pages vues quotidiennes par visiteur et d'après le % du trafic de la recherche et du Nombre total des sites liés. [2]

I.3.1. Facebook

Facebook est un réseau social des plus connu dans le monde qui permet aux utilisateurs de partager des images, des vidéos, des fichiers et de les plublier . il est aussi utilisé pour échanger des messages privé et de créer des groupes entre des personne spécifique .

I.3.2.LinkedIn

LinkedIn est très souvent utilisé pour la vie professionnelle : il constitue aujourd'hui un moyen efficace pour construire, développer et enrichir son capital social. Il vise à établir une relation de confiance entre des professionnels, des étudiants et des entreprises afin que chacun puisse mobiliser ces ressources en ligne pour acquérir ou développer de nouvelles idées, obtenir des opportunités d'emploi, bénéficier des communautés d'experts qui existent sur le réseau. [3]

I.3.3.Instagram

Instagram est une application, un réseau social et un service de partage de photos et de vidéos disponibles sur plates-formes mobiles de type iOS, Android et Windows Phone.
Instagram permet de partager ses photographies et ses vidéos avec son réseau d'amis, de fournir une appréciation positive (fonction « j'aime ») et de laisser des commentaires sur les clichés déposés par les autres utilisateurs. Elle permet aussi de dialoguer avec les membres via l'utilisation de la messagerie interne appelée « Instagram direct ». Les applications telles qu'Instagram contribuent à la pratique de la phonographie, ou photographie avec un téléphone mobile . [4]

I.3.4.Twitter

Twitter est un reseau social de microblogage Il permet à un utilisateur d'envoyer gratuitement de brefs messages, appelés tweets, les gens utilisent des acronymes, commissent des erreurs d'orthographe, utilisent des émoticônes et d'autres caractéristiques qui expriment des significations particulières.

Twitter est actuellement l'un des plates-formes de microblogage les plus populaires. Plusieurs célébrités utilisent Twitter, on y trouve même des chefs d'Etat. C'est pourquoi de nombreux développeurs et de data scientiste utilisent les corpus de tweets pour leurs diversité et leurs richesse d'information textuelles.

La structure de Tweet :

Un tweet est accompagné de plusieurs informations

- l'id du tweet.
- la date du tweet.
- le nom de l'utilisateur qui a posté le tweet.
- le texte du tweet.
- d'émojis.
- La photo du profil.
- L'emplacement de l'utilisation.

Figure 1 la structure d'un Tweet.

I.4. Avantages et inconvénients Des Réseaux Sociaux

Il faut dire que les réseaux sociaux jouent un rôle Très important dans notre vie quotidienne. Facebook, Twitter, Instagram et Snapchat sont partout, toute personne utilise, au minimum, un réseau social chaque jour. Ils sont devenus les principaux moyens de communication. Cependant, on retrouve toujours des avantages et des inconvénients à ces derniers.

Un des plus grand avantage est que ce sont des moyens gratuits et accessibles à tous. Nous pouvons communiquer avec des amis ou des membres de la famille de partout dans le monde. En outre, les réseaux sociaux, comme Facebook, sont un bon moyen de rester en contact avec d'anciens amis comme de nouveaux. Nous pouvons partager nos pensées, photos et vidéos en un seul clic.

Néanmoins, l'un des inconvénients est que ces réseaux sont addictifs. Il existe des personnes qui ont un besoin d'être sur ces sites à toute heure ! Cela peut créer de graves problèmes pour eux. De nombreux étudiants trouvent qu'il est difficile d'étudier et de se concentrer car ils sont trop distraits par leur téléphone. Leurs résultats d'examens sont médiocres à cause de cela. D'autres personnes prennent du retard au travail juste parce qu'ils sont trop obnubilés par leurs comptes sociaux.

Pour notre part, nous pensons que l'un des plus grands dangers des réseaux sociaux est qu'il arrive parfois qu'il y ait des cybers intimidations. C'est à ce moment que les utilisateurs disent des choses méchantes et blessantes sur d'autres utilisateurs et peuvent même les harceler. Ceci mène à de graves répercussions.

La vie privée, informer vos « amis » de certaines facettes de votre vie privée peut mener la personne dans de graves problèmes. Combien d'utilisateurs de Facebook ont été victimes de cambriolage après avoir annoncé, en grande pompe, un voyage dans les Caraïbes? Ou cette autre femme qui s'est vue retirer ses prestations d'assurance chômage après avoir publié des photos d'elle croquée lors d'un voyage dans le Sud. [5]

I.5. L’opinion

L’opinion est un jugement que l'on porte sur un individu, un être vivant, un phénomène, un fait, un objet ou une chose. Elle peut être considérée comme bonne ou mauvaise, tout dépend de la nature de l'individu en fonction de son caractère, ses émotions, son comportement. L'opinion peut influencer et peut donner de mauvaises ou de bonnes informations sur un sujet étudié au sein d'un groupe, d'une personne, d'un objet

I. 5.1. Différents types d'opinions

I.5.1.1. Opinion régulière

Une opinion régulière est souvent appelée simplement une opinion dans la littérature, Par exemple : « la nourriture de ce restaurant est excellente ».

I.5.1.2. Opinion comparative

Une opinion comparative exprime une relation de similitude ou de divergence entre deux ou plusieurs entités sur la base de certains aspects communs des entités. Par exemple, les phrases «سيارة مرسيدس أفضل من سيارة تويوتا» et « سيارة مرسيدس هي الأفضل» expriment deux opinions comparatives. Une opinion comparative est généralement traduite en utilisant la forme comparative ou superlative d'un adjectif ou d'un adverbe, mais pas toujours (par exemple, préférez). Les opinions comparatives ont également bon nombre de types.

I.6. L’analyse De Sentiment

I.6.1. Différents Niveaux D’analyse De Sentiments

I.6.1.1. Au niveau du document

Ce type d'analyse des sentiments est utilisé lorsqu'un document, doit être complètement analysé sous toutes ses formes, comme si une société avait besoin d'un avis sur leur produit. L’avis peut être positif ou négatif selon l'opinion du client . Il s’appuie sur un seul examen, c'est pourquoi ce type d'analyse des sentiments ne peut être utilisé que pour une seule entité et non pour plusieurs entités ou avis. On peut donc dire qu’un seul produit peut être analysé à la fois. Ainsi, plusieurs produits ne sont pas applicables au niveau du document. [6]

I.6.1.2. Au niveau de la phrase

A ce niveau, l'accent est mis sur la phrase plutôt que sur l'ensemble du document et une phrase peut être analysée de trois manières différentes : positive, négative ou neutre. L'orthographe, les fragments de phrases, les erreurs d'écriture, etc. sont quelques exemples à

analyser au niveau de la phrase. L'absence d'un sujet et d'un verbe ou les deux génère un fragment de phrase.

I.6.1.3. Au niveau de l'aspect

Comme on voit, les deux niveaux ci-dessus ne se soucient pas des opinions des gens. Cependant, si on parle du niveau d'aspect, il repose totalement sur des gens qui aiment et n'aiment pas. Le niveau d'aspect se concentre directement sur les opinions plutôt que sur les constructions de langage. Aspect le niveau prend toujours un objectif comme opinion. Un avis prend toujours deux aspects, l'un est positif et l'autre négatif. Par exemple, la phrase « Le moteur du scooter Honda est bon mais la durée de vie de sa batterie est courte », analyse deux aspects fondamentaux, la qualité du moteur et la durée de vie de la batterie. Le sentiment émis sur le moteur du scooter est positif en revanche, celui sur la durée de vie de la batterie est négatif. Cet exemple montre comment l'aspect des niveaux se concentre sur des opinions dissemblables [7]

I.6.2. Les approches d'analyse des sentiments

Il existe deux approches principales :

- Approche basée sur l'apprentissage automatique
- Approche basée sur le TALN

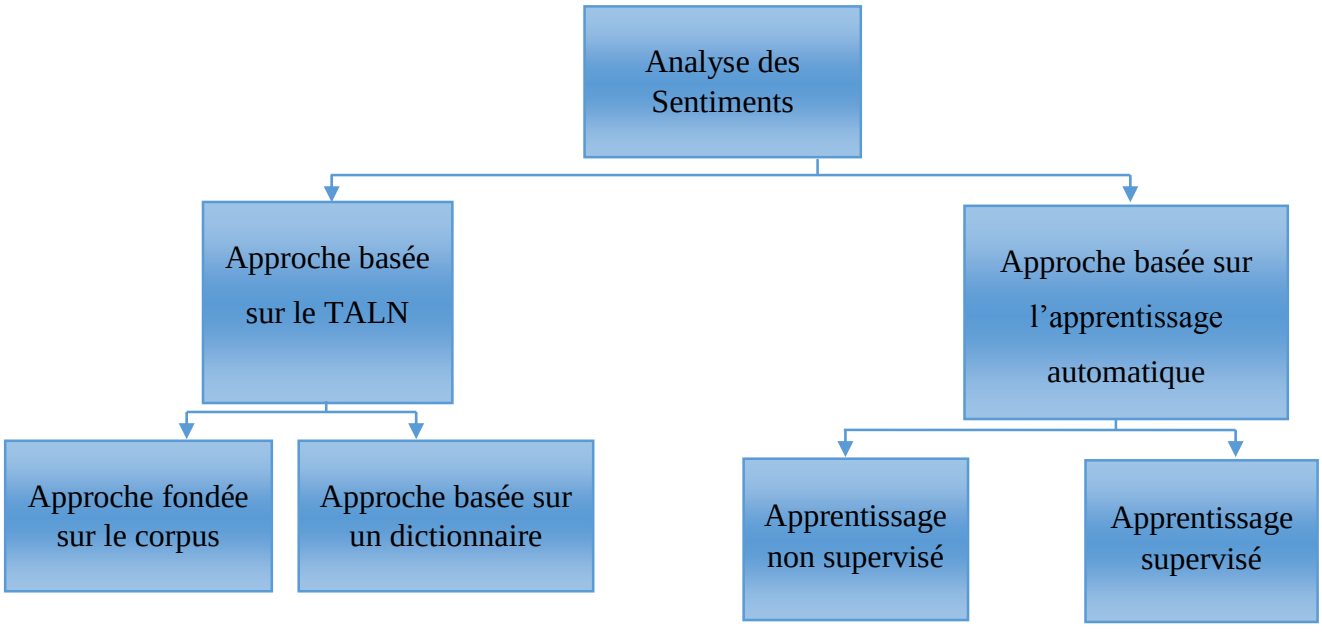


Figure 2: méthodes de classification du sentiment.

I.6.2.1. Approche basée sur le TALN :

Il existe deux approche dans le TALN qui sont :

I.6.2.1.1. Approche basée sur un dictionnaire

Un ensemble de mots d'opinions est collecté manuellement avec des orientations connues. Ensuite, cet ensemble est cultivé par la recherche dans les corpus bien connus comme Word Net pour leurs synonymes et antonymes. Les mots nouvellement trouvés sont ajoutés à la liste de départ, de là, l'itération suivante commence. L'itératif du processus s'arrête quand aucun nouveau mot n'est trouvé. Après que le processus soit terminé, une inspection manuelle peut être effectuée par un expert pour supprimer ou corriger les erreurs.

I.6.2.1.2.Approche basée sur le corpus

L'approche basée sur le corpus contribue à résoudre le problème de trouver des mots d'opinion avec des orientations précises au contexte. Ces méthodes dépendent des modèles syntaxiques ou des modèles qui se produisent ensembles avec un corpus de mots d'opinion pour trouver d'autres mots d'opinion dans le même contexte. Une de ces méthodes était représentée par Hatzivassiloglou et McKeown. Ils ont commencé avec une liste d'adjectifs d'opinion de semences, et les ont utilisés avec un ensemble de contraintes linguistiques pour identifier d'autres mots d'opinion adjectifs et leurs orientations. Les contraintes sont pour des connecteurs comme ET, OU, MAIS, SOIT OU-OU... . La conjonction ET par exemple dit que les adjectifs conjoints ont généralement la même orientation. Cette idée est appelée la cohérence du sentiment, qui n'est pas toujours cohérente dans la pratique. [8]

I.6.2.2. Approche basée sur l’apprentissage automatique :

appelé Machine Learning , il y a deux types d’apprentissages beaucoup utilisé : supervisés et non supervisés. La plus importante différence entre les deux types réside dans le fait que l’apprentissage supervisé se fait sur la base d’une vérité. En d’autres termes, nous avons une connaissance préalable de ce que devraient être les valeurs de sorties de nos échantillons. Par conséquent, l’objectif de l’apprentissage supervisé est d’apprendre une fonction qui, à partir d’un échantillon de données et de résultats souhaités, se rapproche le mieux de la relation entre entrée et sortie observable dans les données. En revanche, l’apprentissage non supervisé n’a pas de résultats étiquetés. Son objectif est donc de déduire la structure naturelle présente dans un ensemble de points de données. [9]

I.6.2.2.1. Apprentissage supervisé

L'apprentissage supervisé, dans le contexte de l'intelligence artificielle (IA) et de l'apprentissage automatique, est un système qui procure à la fois les données en entrée et les données attendues en sortie. Les données en entrée et en sortie sont étiquetées en vue de leur classification, afin d'établir une base d'apprentissage pour le traitement ultérieur des données.

Voici quelques exemples populaires d'algorithmes d'apprentissage automatique supervisé:

- Arbres de décision
- K Nearest Neighbours
- SVC linéaire (classificateur de vecteur de support)
- Régression logistique
- Naive Bayes
- Les réseaux de neurones
- Régression linéaire

I.6.2.2.2. Apprentissage non supervisé

Les données d'entrées ne sont pas annotées. Comment cela peut-il fonctionner diriez-vous ? Eh bien, l'algorithme d'entraînement s'applique dans ce cas à trouver seul les corrélations et les différenciations présentes dans ces données, et à regrouper ensembles celles qui partagent des caractéristiques communes. Dans notre exemple, les photos similaires seraient ainsi regroupées automatiquement au sein d'une même catégorie. [9]

Voici quelques exemples populaires d'algorithmes d'apprentissage automatique non supervisé :

- K-means clustering (K-moyenne)
- Neural networks (Réseaux de neurones) / Deep Learning
- Principal Component Analysis (Analyse des composants principaux)
- Singular Value Decomposition (Décomposition en valeur singulière)

I.7. Travaux réalisés en langue arabe

I.7.1. Cas de dialecte algérien

Ce travail est celui de M'hamed Mataoui et al. (2016), ils ont travaillé sur le dialecte algérien (ALGD) [10]. Leur approche est basé lexique, pour faire leur modèle, ils ont créé trois lexique, lexique de mots clés, lexique de mots de négation, lexique de mots d'intensité et ils ont utilisé deux autres ressources, une liste d'émoticônes avec les polarités qui leur ont été attribuées

et un dictionnaire d'expressions courantes de l'ALGD. Le lexique de mots clés contient 3093 mots (713 mots positifs et 2380 mots négatifs).

Ils ont collecté et annoté leur propre Dataset qui contient 7698 commentaires Facebook. Le tableau 1 présente la distribution des données collectées selon leurs thèmes.

Tableau 1 Distribution des données collectées selon leurs thèmes.

Thèmes	Nombre de commentaires
Economie	1705
Politique	2422
Société	1263
littérature et arts	1215
Divers	1093

Le tableau 2 montre les résultats obtenus par les deux configurations liées au module de calcul de similarité de phrases courantes [11].

Tableau 2 Résultats obtenus par les deux configurations liées au "module de calcul de similarité de phrases courantes".

	Sans utiliser "le module de calcul de similarité de phrases courantes"	En utilisant "le module de calcul de similarité de phrases courantes"
Accuracy	76.68 %	**79.13 %**

D'après le tableau 2, nous avons remarqué que la meilleure configuration de leurs expériences est liée à l'utilisation de module de calcul de similarité de phrases courantes.

I.7.2. Cas de dialecte tunisien

Sur ce projet, on va voir le travail de Salima mdhaffer et al. (2017). Leurs principales contributions sont : Une enquête sur les ressources disponibles pour la langue arabe SA (Sentiment Analysis) MSA et dialectique. La création d'un corpus de formation disponible gratuitement pour le dialecte tunisien et L'évaluation des performances du système Dialecte tunisien SA [12].

Leur corpus appelé TSAC, il est constitué de commentaires écrits sur les pages officielles des radios et des chaînes de télévision tunisiennes, à savoir Mosaïque FM, Jawhra FM, Shemes FM, HiwarElttounsi TV et Nessma TV au cours d'une période allant de janvier 2015 à juin 2016, voir le tableau 3

Tableau 3 Statistiques de corpus TSAC.

	Positive	Négative	Total
Commentaires	8215	8845	17060

Tableau 4 Résultats d'expériences d'Analyse de Sentiment tunisien en utilisant divers classificateurs .

Classificateur	Positive		Negative		Taux d'erreur
	Précision	Rappel	Précision	rappel	0.52
SVM	0.71	0.81	0.78	0.66	0.26
BNB	0.54	0.82	0.62	0.30	0.44
MLP	0.74	0.77	0.77	0.73	**0.25**

Avec le tableau 4 on remarque que :

Il y a un taux d'erreur de 0,25 avec MLP (Multi-Layer Perceptron), et 0,26 avec SVM (Multi-Layer Perceptron) et 0,44 avec BNB.

SVM et MLP obtenaient des résultats similaires pour toutes les configurations expérimentales, cependant, des résultats plus faibles étaient obtenus avec le classificateur BNB, aucune.

I.7.3. Cas de dialecte marocain

Passons maintenant au dialecte marocain, qui est fait par Abdeljalil Elouardighi et al. (2018). Ils ont fait leur travail à partir de l'analyse des sentiments des commentaires Facebook [13]. Les principales contributions dans ce travail sont :

- Citer les propriétés de la langue l'ADM (Arabe Dialectal Marocain) et leurs défis pour l'AS (Analyse de Sentiment),

- Proposer un ensemble de techniques de prétraitement des commentaires Facebook écrits en ADM pour l'AS.

La Figure 3 présente le processus proposé pour l'analyse des sentiments. Ce processus est composé de quatre phases :

Figure 3 Etapes du processus proposé pour l'analyse des sentiments.

a) Collection des commentaires Facebook :

Ils ont ciblé des journaux marocains qui publiaient en ligne des commentaires sur les élections législatives marocaines ayant eu lieu le 07 octobre 2016, la collecte a été effectuée, en utilisant "Facebook Graph API" [14]. Pendant 70 jours ils ont sélectionné 10254 commentaires.

b) Traitement des commentaires:

L'objectif étant d'extraire des variables (sous forme de mots ou de séquences de mots) pour les utiliser dans la classification, nettoyage et une normalisation du texte : suppression des signes, des symboles, des lettres répétées, des mots vides ou des mots qui ne fournissent aucune information sur le sujet étudié, puis l'opération de tokénisation par laquelle le texte du commentaire est divisé en unités lexicales (tokens), Le tableau 5 présente un exemple de prétraitement d'un commentaire.

Tableau 5 Exemple de prétraitement d'un commentaire.

Tache	Résultat
Texte initial	الكلام الي كي يقولو هاذ السياسي ماشي معقوووووول ! هههههههه # السياسة المغربية Les discours de ce policien ne sont pas raisonnables ! hahahaha # politique marocaine
Nettoyage	الكلام الي كي يقولو هاذ السياسي ماشي معقوووووول هههههههه السياسة المغربية
Normalisation	الكلام الي كي يقولو هاذ السياسي ماشي معقول هه السياسة المغربية
Tokénisation	'الكلام' , 'الي' , 'كي' , 'يقولو' , 'هاذ' , 'السياسي' , 'ماشي' , 'معقول' , 'هه' , 'السياسة' , 'المغربية'
Suppression des mots vides	'الكلام' , 'يقولو' , 'السياسي' , 'ماشي' , 'معقول' , 'هه' , 'السياسة' , 'المغربية'
Désuffixation	'كلام' , 'قول' , 'سياس' , 'ماش' , 'معقول' , 'هه' , 'سياس' , 'مغرب'

c) Extraction et sélection de variables :

En fin, 6581 commentaires annotés 2908 négatifs et 3673 positifs, les variables d'entrée sont automatiquement extraites depuis le corpus formé à partir des commentaires prétraités, en utilisant les schémas d'extraction n-gram et de pondération TF/TF-IDF [15].

d) Classification des commentaires :

Par l'application de trois algorithmes, Naïve Bayes (NB), les Forêts Aléatoires (FA) et les Machines à Vecteurs Support (SVM), 50% de Dataset pour l'apprentissage, 25% pour la validation et 25% pour le test

Tableau 6 diffèrent résultat des classificateur avec plusieurs configuration.

Configuration	Classificateur	Accuracy
(Unigram+Bigram)/TF	SVM	0.76
	NB	0.39
	FA	0.71
(Unigram+Bigram)/ TF-IDF	SVM	**0.78**
	NB	0.56
	FA	0.73

On remarque dans le Tableau ci-dessus qu'avec la configuration TF l'accuracy été un peu faible par contre en utilisant le TF-IDF le résultat été bien meilleur

I.7.4. Analyse de sentiments et extraction des opinions pour les sites e-commerce : application sur la langue arabe

Ce travail du quadri nôme Mohamed Ali Sghaier, Houssem Abdellaoui, Rami Ayadi et Mounir Zrigui qui on proposé l'implémentation d'un outil d'analyse des sentiments qui a pour rôle de détecter la polarité des opinions à partir des revues extraites des sites web qui se spécialisent dans le commerce électronique ou les revues des produits en langue arabe.

Pour cela ils ont collecté un corpus par eux-mêmes manuellement à partir de plusieurs ressources web, à titre d'exemples reviewzat , jawal123 , jumia … Il est sous forme d'un ensemble de documents textes, chaque document représente un produit dont chaque produit est représenté par son type, son nom et les revues (les commentaires) sur ce dernier. On a sélectionné cinq types de produits qui forment ce corpus, les types sont Caméra, PC portable, Téléphone portable, Tablet, Télévision. Le corpus contient 250 documents, 2812 phrases et 15466 mots.

Leurs prototype consiste à le collecte des revues sur Internet à partir des sites web spécialisée dans le commerce électronique, ensuite le prétraitement des données, puis la phase de racination

(stemming) ou se fait la réduction des mots à leurs racines, et finalement la détection de la polarité des opinions selon le classificateur SVM soit positive soit négative.

La figure ci-dessous représente l'architecture générale de leurs prototype pour l'extraction des opinions et l'analyse de sentiments.

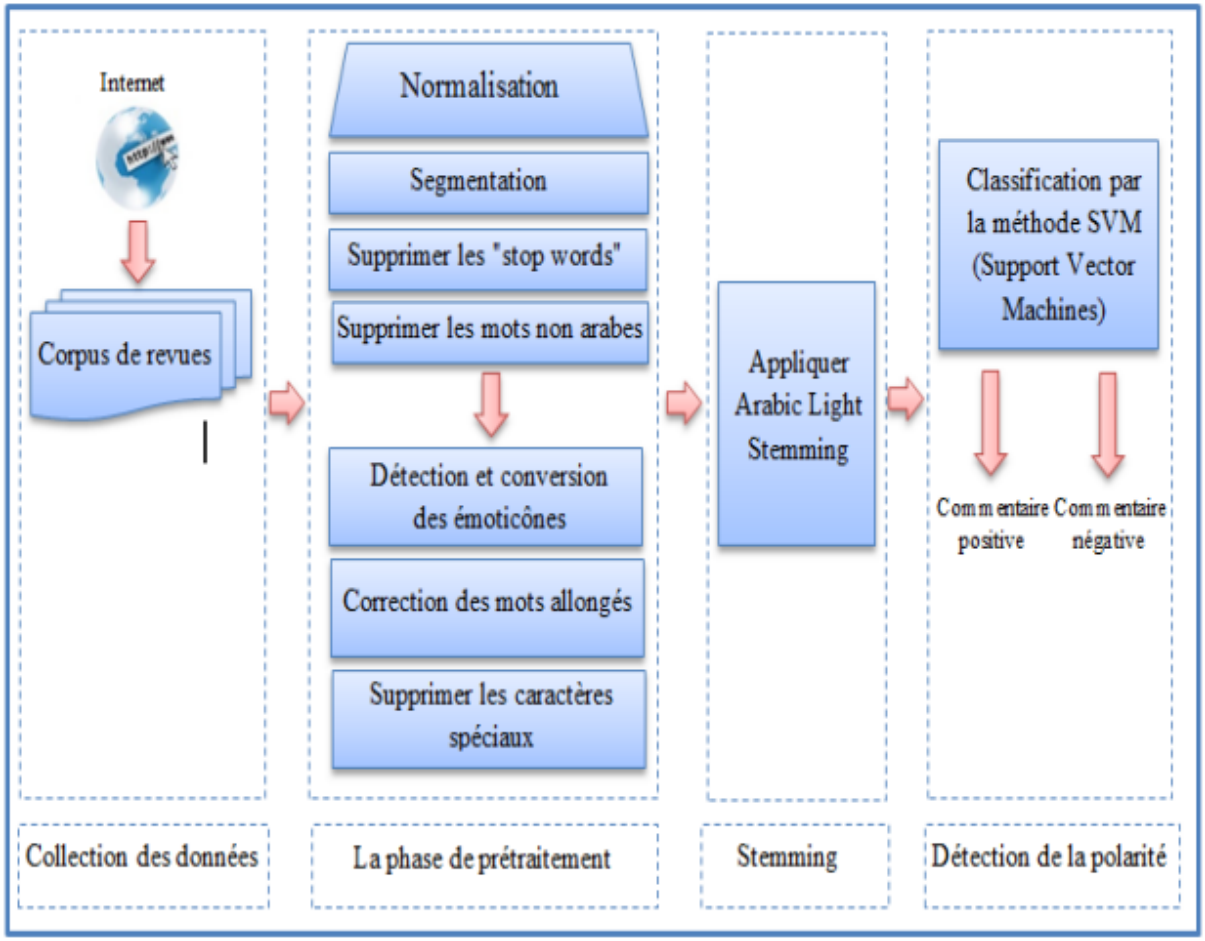


Figure 4 l'architecture générale du système d'analyse de sentiments. *[16]*

Ils ont fait leurs tests avec plusieurs types d'algorithmes de classification. On cite à titre d'exemple le Support Vector Machines (SVM), Naïve Bayes (NB), K-plus proche voisin (KNN). Et ces derniers sont appliqués sur différent combinaison de données prétraitées. La phase de test de la performance des classificateurs appliqué sur le corpus est faite à l'aide de l'outil libre weka, en fait c'est une suite populaire de logiciels d'apprentissage automatique. Écrite en Java, développée à l'université de Waikato, Nouvelle-Zélande.

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

Corpus/ Classificateur	K-plus proches voisins	Support Vector Machines	Naïve Bayes
Corpus à l'état brut	0.712	0.886	0.834
Corpus + Light stemmer	0.76	0.904	0.861
Corpus + Khoha stemmer	0.76	0.904	0.861
Corpus + Normalisation	0.618	0.885	0.871
Corpus +Normalisation + Khoja stemmer	0.58	0.912	0.876
Copus + Normalisation + Light stemmer	0.58	0.912	0.876

On remarque dans le tableau ci-dessus qu'en utilisant une combinaison complète (corpus + normalisation + khouja ou light stemmer) le résultat été bien meilleurs que qu'on utisant une configuration qui manque de normalisation ou de stemmer on remarque que SVM et NB donne un bon résultat mais SVM est beaucoup mieux.

I.7.5. Une étude qualitative pour l'analyse d'opinions en arabe [17]

C'est le travail de Amira Barhoumi, Nathalie Camelin et Yannick Estève qui ont fait une méthodes à base de réseaux de neurones pour la langue arabe, qui ont effectué leur experience sur le corpus Large-scale Arabic Book Review (LABR) qui est un corpus de critique de livre en langue arabe . ils ont implémenté deux systèmes un CNN et un autre LSTM .

Pour leur premier système ils ont fait une implémentation CNN similaire à celle de (Dahou et al . 2016) En plus des embeddings de (Dahou et al., 2016),ils ont également testé les embeddings de (Soliman et al., 2017). Pour leur second système il s'appuie sur LSTM et a été testé avec les différents embeddings.

Tableau 8 Exactitudes des architectures CNN et LSTM sur LABR avec différents embeddings.

	(Dahou et al., 2016)	(Soliman et al., 2017)					
	Web	Twitter		Wikipédia		Web	
	CBOW	CBOW	Skip-gram	CBOW	Skip-gram	CBOW	Skip-gram
CNN	77,39%	77,41%	77,55%	77,51%	77,43%	77,56%	77,47%
LSTM	75,03%	74,87%	74,65%	74,92%	74,58%	74,74%	74,95%

Leurs expériences ont montré que l'architecture CNN est plus performante que l'architecture LSTM, quelque soit le modèle d'embeddings utilisé. leur meilleur système (CNN_Soliman_CBOW_W eb) obtient une exactitude de 77,56% améliorant légèrement le meilleur système publié qui n'utilise pas de connaissances a priori (77,39% pour (Dahou et al., 2016) appliqué sur la répartition officielle)

I.8. Synthèse des travaux présentés

D'après les travaux présentés on remarque que pour faire un système d'analyse de sentiment il faut se basé sur plusieurs critère et plusieurs configuration, mais on peut pas dire qu'une configuration est meilleurs a une autre, car une configuration peux donner un résultat excellent à certaines environnement tandis que si on change l'environnement elle peut ne pas être performante.

Comme celui du premier cas ils se sont basé sur le module de calcul de similarité de phrases courantes et on a remarqué que les résultats été mieux, on voie aussi dans le 3éme cas ils ont comparé différente configuration parmi eux il ont vu que la meilleurs configuration est celle de bi gram et uni gram avec Tf-Idf avec le classificateur SVM , Pour le dernier cas est plutôt basé sur le Deep Learning, ils ont fait deux modèles CNN et LSTM et ils les ont testé sur diffèrent embeding

I.9. Conclusion

Enfin on peut dire à la fin de ce chapitre que l'analyse de sentiment peut être faite avec plusieurs approches et différente méthodes avec plusieurs configurations et on peut conclure qu'il y a des classificateurs mieux que d'autre comme le SVM dans le machine Learning et que ce dernier peux ne pas être formant si on lui choisit une mauvaise configuration pareille pour Deep learning.

C'est pourquoi on va voir dans le chapitre suivant les différente technique qu'on va utiliser ainsi les meilleures configurations qu'ils leurs convient.

Chapitre II Apprentissage Automatique et apprentissage Profond

II.1. Introduction

Notre travail de recherche est de apporter à traiter les tweets arabes pour l'analyse des sentiments et de faire la comparaison de différentes approches et voir laquelle de cela est la meilleur.

Le prétraitement des données est parmi les phases les plus importants dans le cadre d'analyse des sentiments, Bien que le prétraitement des données soit un outil puissant permettant de traiter des données complexes. Il existe de nombreux types de données à partir desquels des informations pertinentes pour une tâche donnée peuvent être extraites. Cependant, quel que soit le type de données, il est nécessaire de prétraiter les données brutes afin de pouvoir ensuite les traiter avec des processus unifiés et non une multitude de processus adaptés à tous les cas possibles.

Dans ce chapitre on va se basé sur les différentes techniques qu'on va utiliser leurs fonctionnements leurs configurations sur les deux approches ainsi que les techniques qui nous aide à avoir une bonne prédiction.

II.2. Machine Learning et Deep Learning

Dans notre projet on a fait deux principaux modèles un qui est basé sur le machine Learning et un autre sur le deep Learning pour faire la comparaison.

Si on voie en terme pratique le Deep Learning n'est qu'un sous ensemble du Machine Learning car il s'appuie sur un apprentissage sans surveillance.

II.2.1. Machine Learning

Le machine learning ou « apprentissage automatique » en français est un concept qui fait de plus en plus parler de lui dans le monde de l'informatique, et qui se rapporte au domaine de l'intelligence artificielle. Encore appelé « apprentissage statistique », ce terme renvoie à un processus de développement, d'analyse et d'implémentation conduisant à la mise en place de procédés systématiques. Pour faire simple, il s'agit d'une sorte de programme permettant à un ordinateur ou à une machine un apprentissage automatisé, de façon à pouvoir réaliser un certain nombre d'opérations très complexes.

L'objectif visé est de rendre la machine ou l'ordinateur capable d'apporter des solutions à des problèmes compliqués, par le traitement d'une quantité astronomique d'informations. Cela offre ainsi une possibilité d'analyser et de mettre en évidence les corrélations qui existent entre deux ou plusieurs situations données, et de prédire leurs différentes implications. [18]

II.2.1.1. Les différentes méthodes utilisées

II.2.1.1.1. Régression logistique

La régression logistique est une approche statistique qui peut être employée pour évaluer et caractériser les relations entre une variable réponse de type binaire (par exemple : Vivant / Mort, Malade / Non malade), et une, ou plusieurs, variables explicatives, qui peuvent être de type catégoriel (le sexe par exemple), ou numérique continu (l'âge par exemple).

La régression logistique appartient aux modèles linéaires généralisés. Pour rappel, il s'agit de modèles de régression qui sont des extensions du modèle linéaire, et qui reposent sur trois éléments :

1- un prédicteur linéaire
2- une fonction de lien
3- une structure des erreurs

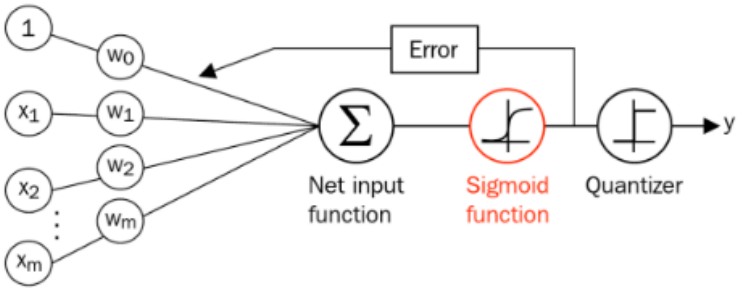


Figure 5 déroulement de regression logistic. *[19]*

Le but du jeu c'est qu'on trouve une ligne (BoundaryDecision) séparant les deux groupes (les cercles et les crois).

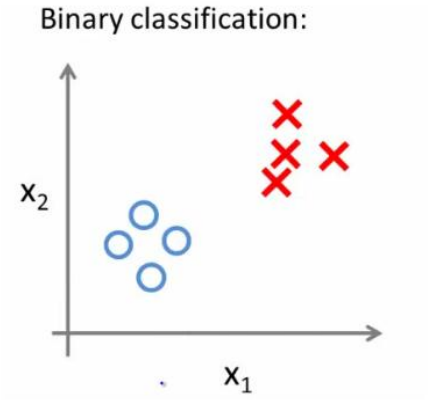


Figure 6 Régression logistique. *[20]*

Logistic Regression est un modèle de classification linéaire qui est le pendant de la régression linéaire , quand Y ne doit prendre que deux valeurs possibles (0 ou 1). Comme le modèle est linéaire, la fonction hypothèse pourra s'écrire comme suit :

$$S\left(X^{(i)}\right) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \ldots + \theta_n x_n$$

avec :

$X^{(i)}$: une observation (que ce soit du *Training Set* ou du *Test Set*), cette variable est un vecteur contenant

x_i: est une variable prédictive (feature) qui servira dans le calcul du modèle prédictif

θ_i : est un poids/paramètre de la fonction hypothèse. Ce sont ces qu'on cherche à calculer pour obtenir notre fonction de prédiction

θ_0 : est une constante nommée le bias (biais) [21]

II.2.1.1.2. Machine à vecteurs de support

Les **machines à vecteurs de support** ou **séparateurs à vaste marge** (en anglais support vector machine, SVM) sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination et de régression. Les SVM sont une généralisation des classificateurs linéaires.

La logique en classification de document par SVM est d'avoir un classifier par classe. Pour deux classes d'exemples donnés, le but de SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est un classificateur linéaire appelé hyperplan. Dans la figure qui suit, on détermine un hyperplan qui sépare les deux ensembles de points.

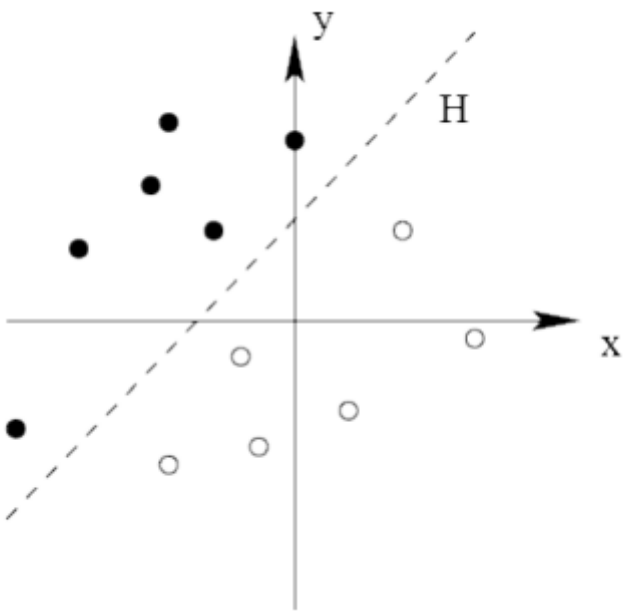


Figure 7 schéma qui sépare les deux ensembles de points. *[22]*

Les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support [23]

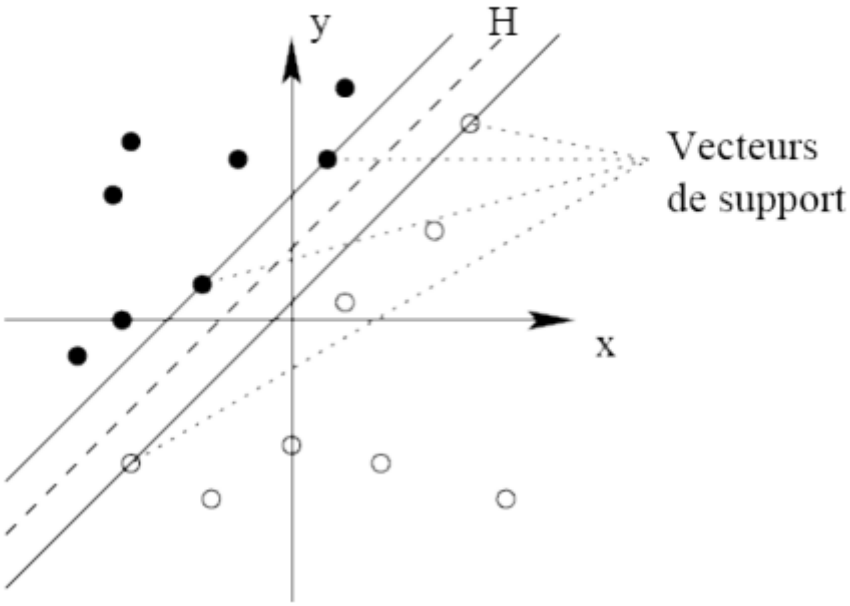


Figure 8 Schéma montrant les vecteurs support.

II.2.1.1.3. k-Nearest Neighbors (KNN)

L'algorithme k-plus proche voisins, ou KNN, est l'un des algorithmes d'apprentissage automatique les plus simples. Habituellement, k est un petit nombre impair - parfois seulement

1. Plus k est grand, plus la classification sera précise, mais plus la classification prendra du temps.

Les algorithmes KNN utilisent des données et classifient les nouveaux points de données en fonction de mesures de similarité (par exemple, fonction de distance). Le classement se fait à la majorité des voix de ses voisins. Les données sont affectées à la classe qui a les voisins les plus proches.

Comme on vient de le dire, KNN a besoin d'une fonction de calcul de distance entre deux observations. Plus deux points sont proches l'un de l'autre, plus ils sont similaires et vice versa.

Il existe plusieurs fonctions de calcul de distance, notamment, la distance euclidienne, la distance de Manhattan etc … . On choisit la fonction de distance en fonction des types de données qu'on manipule. Ainsi pour les données quantitatives (exemple : poids, salaires, taille, montant de panier électronique etc…) et du même type, la distance euclidienne est un bon candidat. Quant à la distance de Manhattan, elle est une bonne mesure à utiliser quand les données (input variables) ne sont pas du même type (exemple :age, sexe, longueur, poids etc…).

- La distance euclidienne :

Distance qui calcule la racine carrée de la somme des différences carrées entre les coordonnées de deux points :

$$D_e(x, y) = \sqrt{\sum_{j=1}^{n} (x_j - y_j)^2}$$

- Distance Manhattan :

la distance de Manhattan: calcule la somme des valeurs absolues des différences entre les coordonnées de deux points :

$$D_m(x, y) = \sum_{i=1}^{k} |x_i - y_i|$$

II.2.2. Le Deep Learning

L'apprentissage profond ou apprentissage en profondeur (en anglais : deep learning) est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires. Ces techniques ont permis des progrès importants et rapides dans les domaines de l'analyse du signal sonore ou visuel et notamment de la reconnaissance faciale, de la reconnaissance vocale, de la vision par ordinateur, du traitement automatisé du langage mais pour notre cas nous allons nous intéresser beaucoup plus sur le traitement automatisé du langage. Dans les années 2000, ces progrès ont suscité des investissements privés, universitaires et publics importants, notamment de la part des GAFAM (Google, Apple, Facebook, Amazon, Microsoft). Elle fait partie d'une famille de méthodes d'apprentissage automatique fondées sur l'apprentissage de modèles de données.

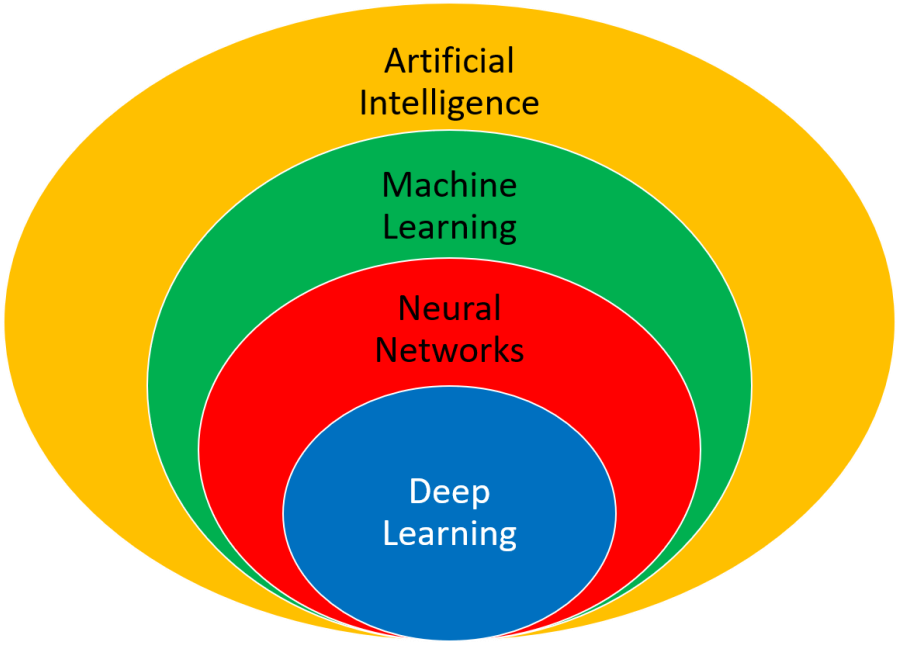


Figure 9 Apprentissage profond | Optimisation dynamique.

L'apprentissage en profondeur peut alors être défini comme des réseaux de neurones avec un grand nombre de paramètres et de couches dans l'une des quatre types d'architectures de réseau fondamentales :

Unsupervised Pre-trained Networks

Convolutional Neural Networks

Recurrent Neural Networks

Recursive Neural Networks

II.2.2.1. Réseau de neurones artificiels

Un réseau de neurones artificiels ou artificial neurones network , est un système dont la conception est à l'origine schématiquement inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques.

Les réseaux de neurones sont généralement optimisés par des méthodes d'apprentissage de type probabiliste, en particulier bayésien. Ils sont placés d'une part dans la famille des applications statistiques, qu'ils enrichissent avec un ensemble de paradigmes permettant de créer des classifications rapides et d'autre part dans la famille des méthodes de l'intelligence artificielle auxquelles ils fournissent un mécanisme perceptif indépendant des idées propres de l'implémenter, et des informations d'entrée au raisonnement logique formel

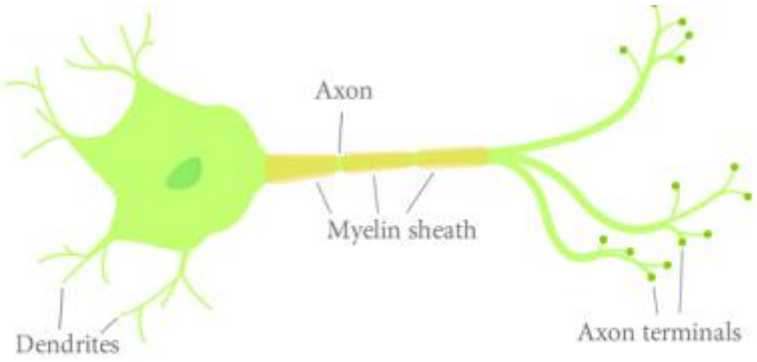


Figure 10 Un neurone réel.

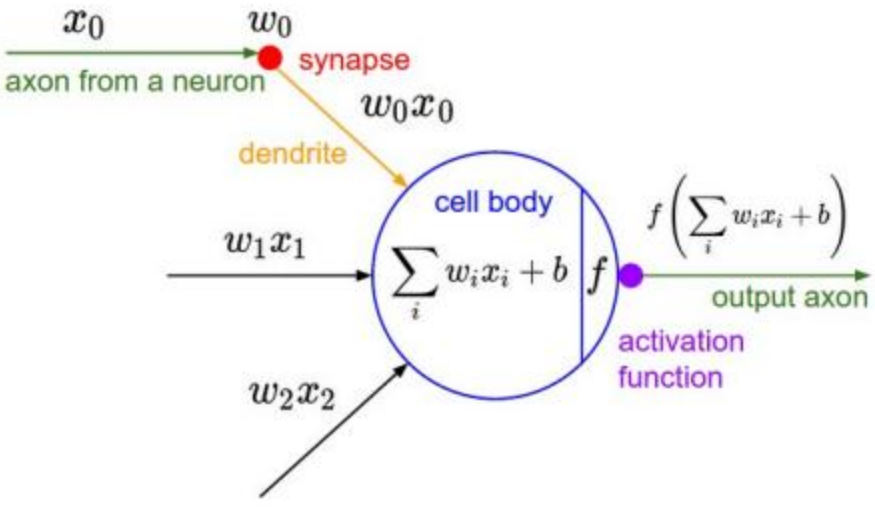


Figure 11 Un neurone artificiel. *[24]*

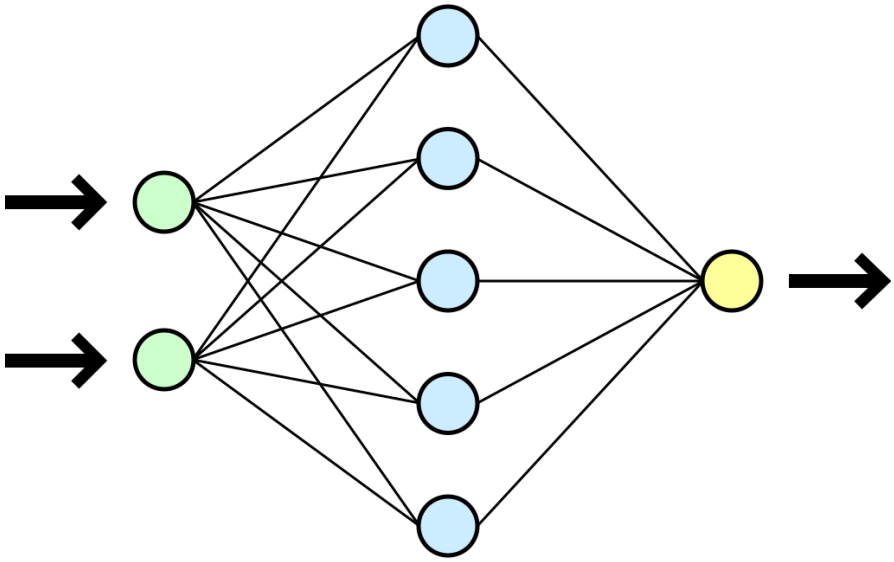


Figure 12 un réseau artificiel de neurones simplifiés.

II. 2.2.2. Fonction d'Activation

Dans le domaine des réseaux de neurones artificiels, la fonction d'activation est une fonction mathématique appliquée à un signal en sortie d'un neurone artificiel c'est l'une des plus importante tache dans ce domaine. Le terme de "fonction d'activation" vient de l'équivalent biologique "potentiel d'activation", seuil de stimulation qui, une fois atteint entraîne une réponse du neurone. La fonction d'activation est souvent une fonction non-linéaire. Sans une fonction d'activation, un réseau de neurones est juste un modèle de régression linéaire. Un exemple de fonction d'activation est la fonction de Heaviside, qui renvoie tout le temps 1 si le signal en entrée est positif, ou 0 s'il est négatif. Il existe plusieurs types de c'est fonction parmi lesquelles nous trouvons

II. 2.2.2.1. Fonction Sigmoïde

Cette fonction est comme une fonction pas à pas mais de nature non linéaire. Sa sortie est entre 0 et 1 et a une courbe de forme S. Si on remarque bien on voie que les valeur X entre -2 et 2 leur valeur Y sont très rapide ce qui signifie qu'un petite modification de X dans cette intervalle entrainera un changement significatif des valeur Y .

Sa fonction est : $$\sigma(z) = \frac{1}{1+e^{-z}}$$

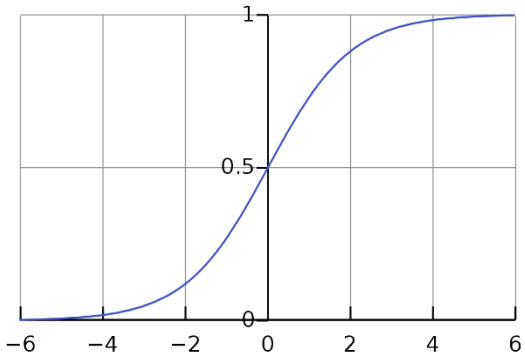


Figure 13 Représentation du graphe de fonction sigmoïde. *[25]*

II. 2.2.2.2. Fonction Tanh

La fonction Tanh est également appelée "tangente hyperbolique".

Cette fonction ressemble à la fonction Sigmoïde. La différence avec la fonction Sigmoïde est que la fonction Tanh produit un résultat compris entre -1 et 1. La fonction Tanh est en terme général préférable à la fonction Sigmoïde car elle est centrée sur zéro. Les grandes entrées négatives tendent vers -1 et les grandes entrées positives tendent vers 1.

$$F(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

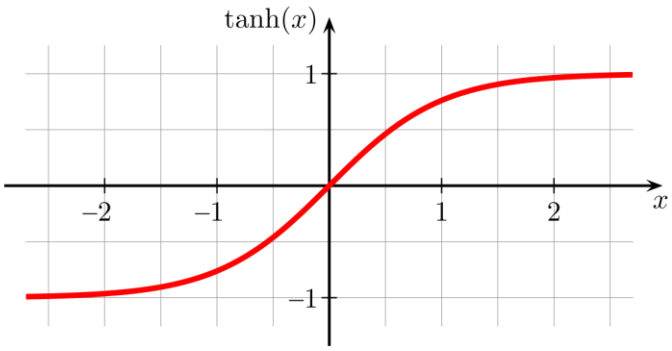


Figure 14 La fonction Tanh. *[25]*

II.2.2.2.3. Fonction ReLu

c'est une fonction simple qui rend le X comme sortie si il est supérieur a 0 sinon sa sortie sera un 0 , sa fonction est :

$$\text{Fonction (X)} = \max(0, X)$$

Dans la figure si dessous on remarque une ligne droite qui nous fais croire qu'elle est une fonction linéaire mais en réalité ReLu est non-linéaire La plage de ReLu est [0, infini]. Calculer ReLu est moins cher que d'autres fonctions d'activation, car il a un fonctionnement mathématique simple

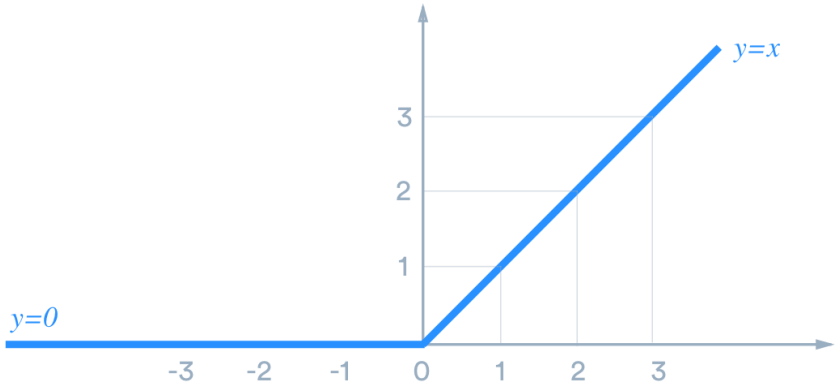


Figure 15 Représentation graphique de la fonction ReLu. *[25]*

II.2.2.2.4. Fonction Softmax

appelée Régression Softmax c'est une généralisation de la régression logistique que nous pouvons utiliser pour la classification multi-classes. Contrairement à d'autres types de fonction, la sortie d'un neurone d'une couche utilisant la fonction softmax dépend des sorties de tous les autres neurones de sa couche. Cela s'explique par le fait qu'il nécessite que la somme de toutes les sorties soit égale à 1.

$$F(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{pour tout} \quad i \in \{1, \dots, K\}$$

II. 2.2.3. Fonction d'erreur

La fonction d'erreur est utilisé pour déterminer la bonne combinaison des poids pour ce faire elle calcule la différence entre la sortie réelle du réseau et la sortie attendu après qu'un cas a circulé à travers le réseau.

II. 2.2.4. La Régularisation

La régularisation est faite pour éviter le problème du sur-apprentissage. En d'autre terme la complexité du modèle augmente de telle sorte que l'erreur d'apprentissage diminue mais pas l'erreur de test. Il existe plusieurs méthodes de régler ce problème parmi elle on a :

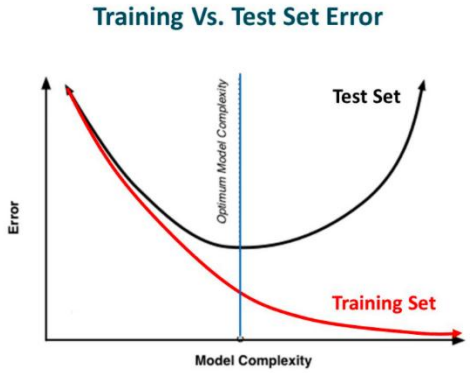


Figure 16 Probleme du surapprentissage. *[26]*

II.2.2.4.1. Le Dropout

Il y a une chose dans l'apprentissage automatique qu'on appelle le « overfitting » qui veut dire le sur-apprentissage qui dégrade la performance des algorithme c'est pourquoi il a été crée le Dropout qui a pour but de réduire l'erreur de généralisation , il s'agit d'une technique pour éviter le overfitting.

Il est généralement utilisé à la sortie de certaines des couches du réseau. Il enlève aléatoirement certains des neurones (ainsi que leurs connexions d'entrée et de sortie).

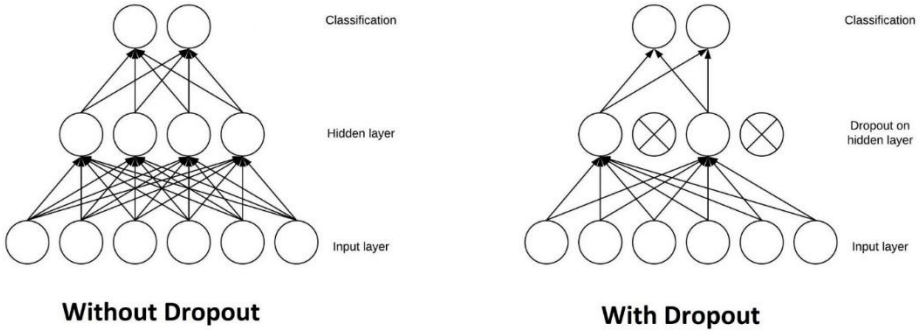


Figure 17 Explication du fonctionnement du Dropout. *[27]*

Quand les neurones sont effacés aléatoirement du réseau au cours de l'apprentissage, les neurones restant sont obligé de répondre et de contrôler la représentation requise pour faire des prédictions pour les neurones manquants. Cette méthode améliore la généralisation parce que elle oblige les couche à apprendre le même concept avec diffèrent neurone .

II.2.2.4.2. Early-stopping

L'arrêt précoce est une sorte de stratégie de validation croisée où nous conservons une partie de l'ensemble de formation comme ensemble de validation. Lorsque nous constatons que les performances sur l'ensemble de validation se détériorent, nous arrêtons immédiatement la formation sur le modèle. C'est ce qu'on appelle l'arrêt prématuré.

Figure 18 Fonctionnement early stopping. *[26]*

II. 2.2.5. Les réseaux neuronaux récurrents

Un réseau de neurones récurrent (Récurrent Neural Network - RNN) est un réseau de neurones dont le graphe de connexion contient au minimum un cycle. Les couches du RNN sont des entités primitives qui permettent aux réseaux d'apprendre à partir de séquence d'entrées.

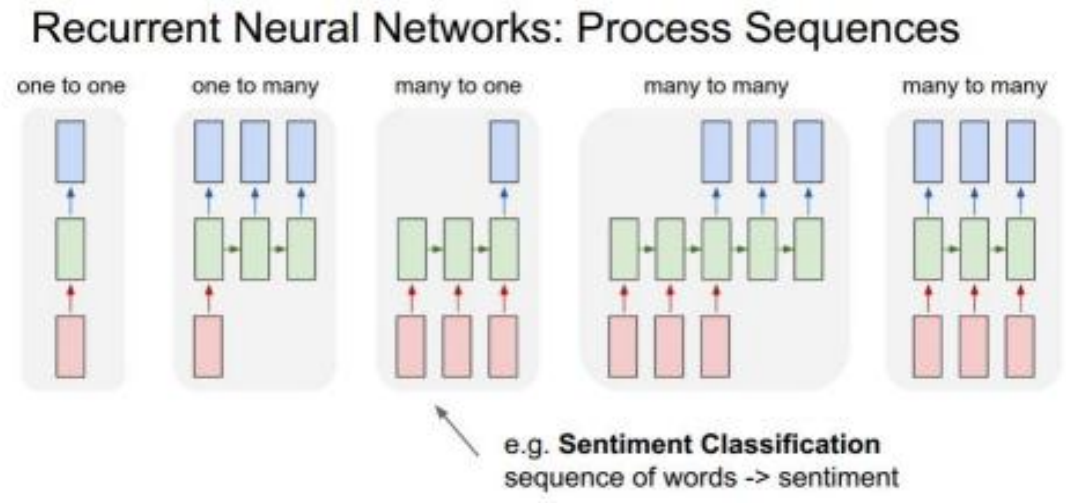


Figure 19 Les types de séquences d'entrée pour un réseau récurrent.

Ces dernières années, un type de RNN est sorti du lot grâce a ses excellentes performance sur des taches aussi nombreuses que variées : les réseaux de neurones à base de cellules Long Short-Terme Memory (LSTM).

Tout comme les systèmes de contrôle, Les réseaux neuronaux récurrents fonctionnent selon le principe du bouclage et du chaînage, comme représenté dans la figure 18

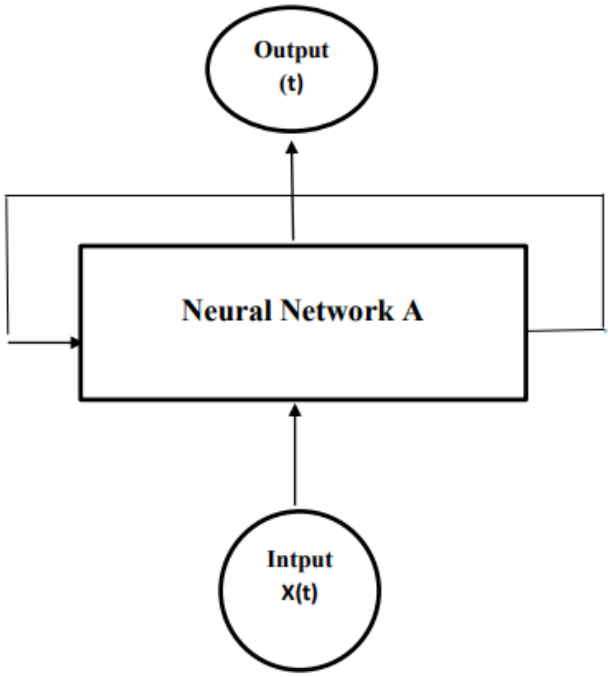


Figure 20 RNN Looping.

a partir de cette figure , disons que A est une partie du réseau neuronal, cette partie a besoin d'une valeur d'entrée pour démarrer le traitement et produire une valeur de sortie, qui est x(t) et h(t) dans ce cas. Neural Network A, fais une boucle qui est fondamentalement une copie multiple de lui-même et permet de partager des connaissances entre les prochaine étapes du réseau c'est ce qui met en valeur ce type des autres techniques de réseau neuronal.

II. 2.2.5.1 Les Réseaux Long Short-Term Memory (LSTM)

Les réseaux de mémoire à long terme à court terme généralement appelés simplement «LSTM» - sont un type spécial de RNN, capable d'apprendre les dépendances à long terme, Ils travaillent très bien sur une grande variété de problèmes et sont maintenant largement utilisés

Les LSTM sont explicitement conçus pour éviter le problème de dépendance à long terme.

Ils sont capables de mémoriser une information pendant une longue période c'est leur comportement par default

Tous les réseaux neuronaux récurrents ont la forme d'une chaîne de modules répétitifs de réseau neuronal. Dans les RNN standard, ce module répétitif aura une structure très simple, telle qu'une couche de tanh unique.

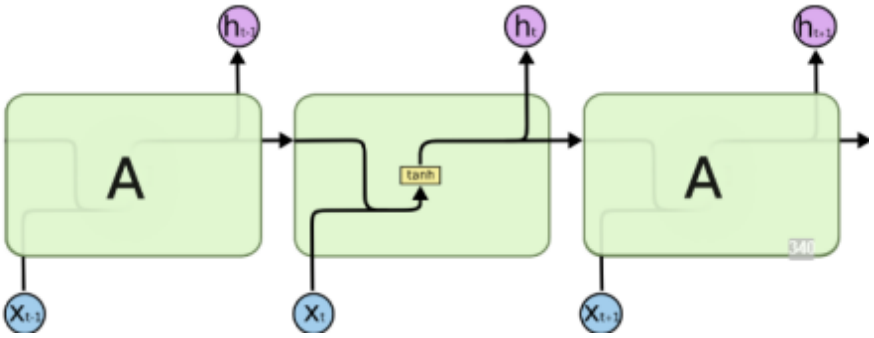


Figure 21 le module de répétition dans le réseau neuronal récurrent.

Les LSTM ont également cette structure en forme de chaîne, mais le module répétitif à une structure différente. Au lieu d'avoir une seule couche de réseau neuronal, il y en a quatre, interagissant d'une manière très spéciale

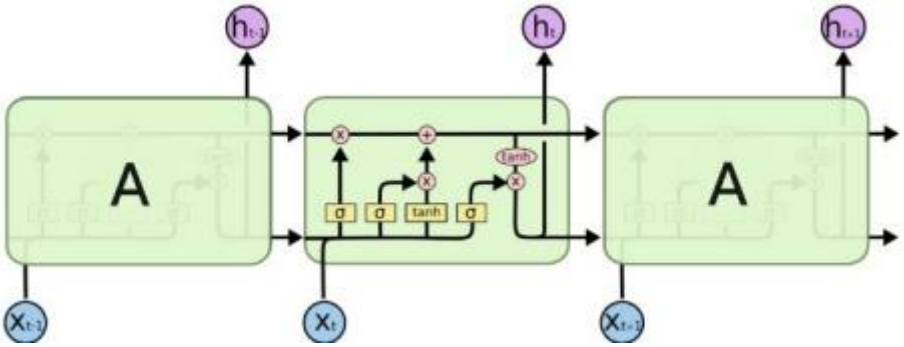
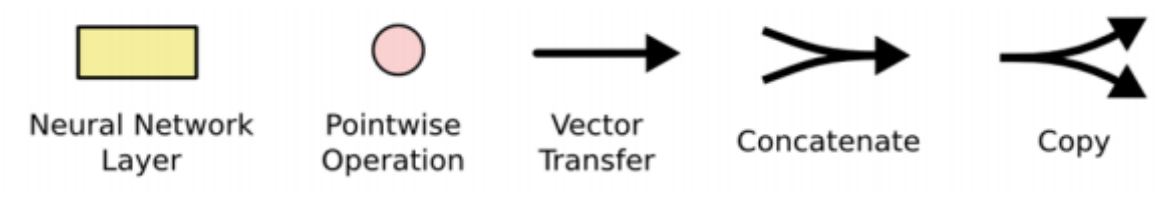


Figure 22 Le module de répétition dans un LSTM.

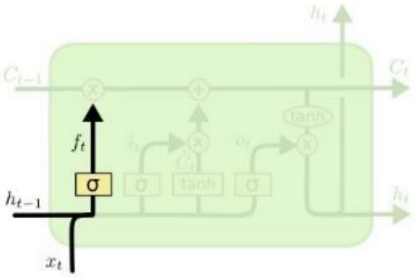


Sur la Figure , chaque ligne contient le mot vecteur, généralement de la sortie du bloc à la suivante en entrée. Les opérations ponctuelles comme la multiplication ou l'addition de vecteurs sont effectuées par des cercles roses. Les rectangles jaunes sont une couche de réseaux de neurones appris. Les lignes fusionnant les unes avec les autres effectuent une concaténation. Le contenu est copié et va à différents endroits avec la ligne de Branchement où Xt est l'entrée à l'étape t et ht est la sortie à l'instant t.

Description d'une cellule de Lstm

Une cellule Lstm agit comme un convoyeur d'information modulée à l'aide de 3 portes configurables (par entraînement)

1.1.Porte d'oubli : Décide quoi garder du contexte précédent pour mettre à jour de l'état de la cellule, basé sur l'entrée courante :



$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f)$$

Avec :

ht−1 : La sortie a l'instant t-1

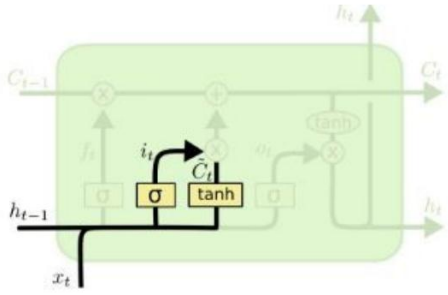
xt : L'entrée courant a l'instant t

b_f : C'est le bais

w_f : C'est le poids

σ : C'est la fonction sigmoïde

1.2 Porte entrée et circuit de mémoire : i décide quoi contribuer de la mémoire ($\tilde{C}$t est équivalent à l'état caché dans un RNN standard) pour mettre à jour de l'état de la cellule, basé sur x_t



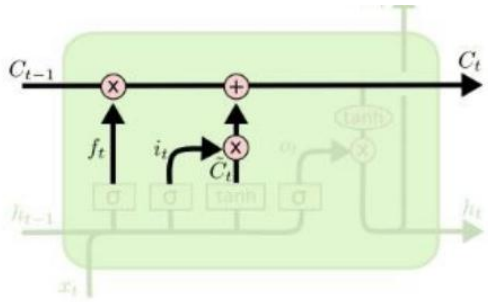
$$i_t = \sigma\,(W_i.[h_{t-1}, x_t] + b_i)$$
$$\check{C}_t = tanh(W_C.[h_{t-1}, x_t] + b_C)$$

Avec :

tanh : C'est la fonction d'activation tangente hyperbolique

C_t : Une valeur candidate

1.3 Mise à jour de la cellule : On combine ce qui a été retenu de l'état précédent de la cellule avec celui retenu de la mémoire

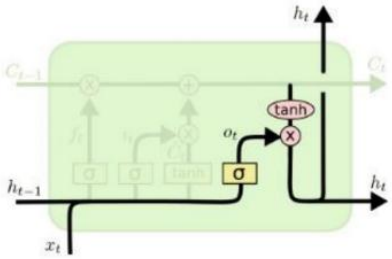


$$C_t = f_t * C_{t-1} + i_t * \check{C}_t$$

Avec :

C_t : État interne

1.4 Porte de sortie : Décide quoi rendre publique du nouvel état de la cellule :



$$o_t = \sigma\,(W_0[h_{t-1}, x_t] + b_0)$$
$$h_t = o_t * tanh(C_t)$$

Avec :

h_t : La sortie

II. 2.2.5.2 LSTM Bidirectionnel

Celui si dispose de deux réseaux (deux LSTM), d'une information d'accès dans le sens direct et d'un autre accès dans le sens inverse (comme le la figure ci-dessous). Ces réseaux on accès aux informations passé et futur, par contre, le résultat est fait à partir du contexte passé et futur.

$$\hat{y}^{<t>} = g\left(W_y[\vec{a}^{<t>}, \overleftarrow{a}^{<t>}] + b_y\right)$$

$\vec{a}^{<t>}$: la sortie le Lstm direct (gauche à droite)

$\overleftarrow{a}^{<t>}$:la sortie le Lstm inverse (droit à gauche)

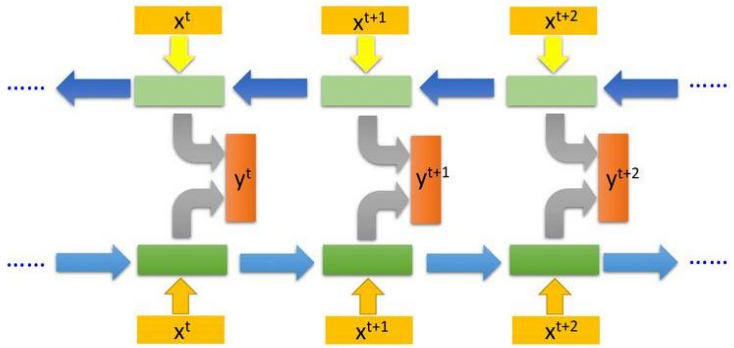


Figure 23 Le modele LSTM Bidirectionnel.

II.3. Vectorisation

Pour ce point il concerne le machine learning elle se devise en deux principaux étapes

II.3.1 Bag of words

Les algorithmes d'apprentissage automatique ne peuvent pas travailler directement avec du texte brut ; le texte doit être converti en nombres. Plus précisément, des vecteurs de nombres.

C'est ce qu'on appelle l'extraction de caractéristiques ou l'encodage de caractéristiques.

Un sac de mots est une représentation de texte qui décrit l'occurrence de mots dans un document.

Exemple :

Tweet1 =”من دون التفاؤل ما في حياة”

Tweet2 = “ما اضيق العيش لولا فسحة الامل،...الحمد لله”

Tableau 9 Representation d'un tweet dans BOW.

	لله	الحمد	الامل	فسحة	العيش	اضيق	التفاؤل	حياة
Tweet1	0	0	0	0	0	0	1	1
Tweet2	1	1	1	1	1	1	0	0

II.3.2. TF-IDF

La formule TF-IDF n'a créé aucune nouvelle règle pour l'optimisation des textes, mais elle a plutôt permis la redécouverte de la pondération des mots dans un contenu.

Calcul de TF :

TF(t) = Nombre d'apparition du terme t dans le document / Nombre total de termes dans le document

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

Calcul de IDF :

Le terme « qui » n'apparaît pas dans le deuxième document. Ainsi :

$$idf_i = log \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

Poids final:

$$tf - idf = \frac{n_{i,j}}{\sum_k n_{k,j}} * log \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

Ainsi le résultat le plus grand apparait comme « le plus pertinent » [28]

II.4. Word Embedding

Word Embedding est un type de représentation de mots qui permet aux mots ayant une signification similaire d'avoir une représentation similaire.

C'est une représentation distribuée pour le texte qui est peut-être l'une des avancées clés pour les performances impressionnantes des méthodes d'apprentissage en profondeur sur des problèmes complexes de traitement du langage naturel. [29]

Word2Vec une l'une des méthodes statiques pour apprendre efficacement un mot autonome incorporé à partir d'un corpus de texte.

De plus, les travaux impliquaient l'analyse des vecteurs appris et l'exploration des mathématiques vectorielles sur les représentations des mots. Par exemple, le fait de soustraire «l'homme» de «Roi» et d'ajouter «la femme» aboutit au mot «Reine», capturant l'analogie «le roi est à la reine ce que l'homme est à la femme».

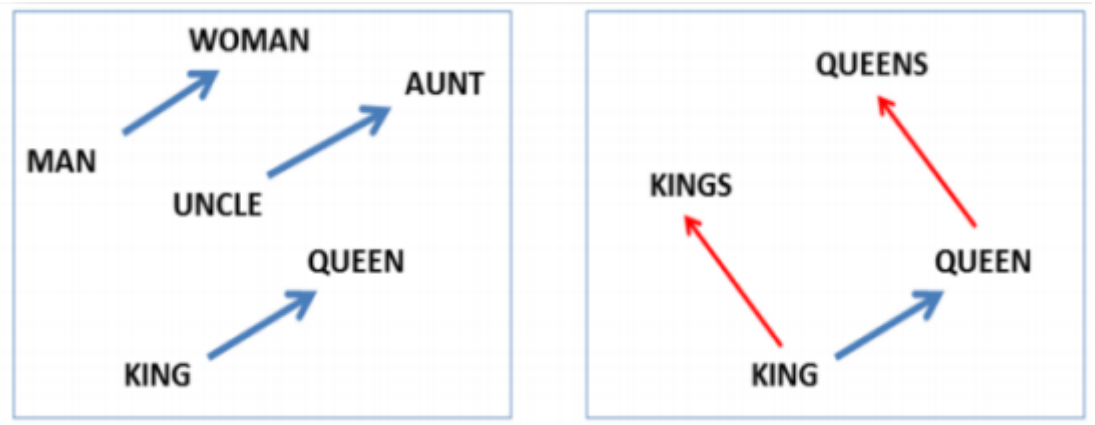


Figure 24 : représentation de word embeding.

Deux modèles d'apprentissage différents ont été introduits qui peuvent être utilisés dans le cadre de l'approche Prédefinie de Keras pour apprendre le mot incorporant; elles sont:

- Continuous Bag-of-Words (CBOW).
- Continuous Skip-Gram .

II.4.1.Continuous Bag-of-Words (CBOW)

Le modèle CBOW est le modèle inverse du model Skip-Gram et le contraire est juste, surement parce que les entrées sont plus riches que dans le modèle Skip-Gram. Il y a plusieurs entrées pour une sortie unique. Et il plusieurs fois plus rapide pour s'entraîner que le skip-gramme, une précision légèrement meilleure pour les mots fréquents

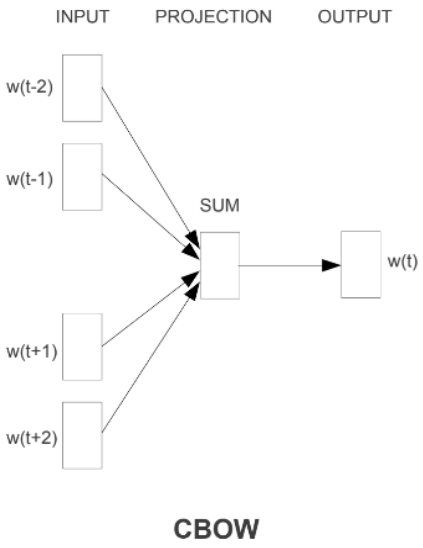


Figure 25 Le modèle CBOW.

II.4.2.Skip-Gram

Le modèle Skip-Gram inverse l'utilisation des mots cible et contexte. Dans ce cas, le mot cible est alimenté à l'entrée, la couche cachée reste la même et la couche de sortie du réseau de neurones est répétée plusieurs fois pour s'adapter au nombre choisi de mots de contexte.

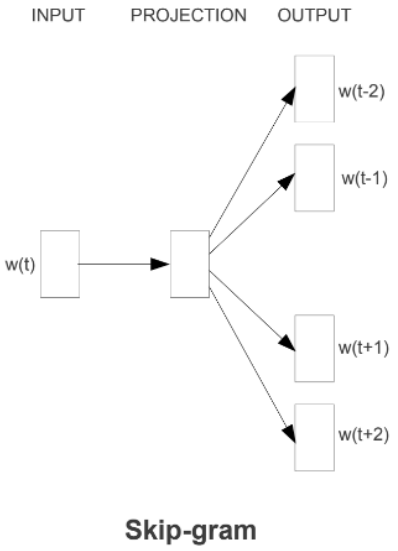


Figure 26 Le modèle Skip-gram.

II.5. Mesure de performance des modéles

II.5.1. Justesse (Accuracy)

il indique le pourcentage de bonnes prédictions. C'est un très bon indicateur parce qu'il est très simple à comprendre. Pour une classification binaire, la justesse peut être calculée en termes de positifs et de négatifs comme

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN}$$

II.5.2. Précision

La précision permet de répondre à la question suivante « **Quelle proportion d'identifications positives était effectivement correcte ? »** elle se calcule comme suit :

$$Précision = \frac{VP}{VP + FP}$$

II.5.3. Rappel

Le rappel permet de répondre à la question suivante **« Quelle proportion de résultats positifs réels a été identifiée correctement ? »** elle se calcule comme suit :

$$Rappel = \frac{VP}{VP + FN}$$

II.5.4. F1 Score

C'est la moyenne harmonique de la précision et du rappel

$$F1 - score = 2.\frac{Précision\,.Rappel}{Précision + Rappel}$$

II.6. Conclusion

En conclusion nous remarquons que malgré que le deep Learning fais partie du machine Learning sa méthode de faire est totalement différente à celle du machine Learning, et que chaque approche a différente configuration qu'il faut choisir minutieusement comme l'exemple du deep learning qui lui utilise différente fonction d'activation.

Chapitre III
Modélisation de la solution proposée

III.1. Introduction

Dans ce chapitre, nous allons présenter, le corpus sur lequel nous avons travaillé, les expérimentations que nous avons effectuées sur et les résultats obtenus. Nous avons traité des tweets en langue arabe standard moderne (MSA) en expérimentant différentes approches deep Learning et machine learning.

En ce qui concerne le Machine learning, nous avons utilisé trois algorithmes qui sont l KNN, LR et le SVM. Du coté Deep learning, nous avons travaillé avec les LSTM vu les avantage qu'ils offrent et nous avons enfin implémenté un LSTM bidirectionnel.

III.2. L'architecture du système

Avant de détailler notre travail, nous représentons dans la figure si dessous l'architecture globale de notre approche d'analyse de sentiments.

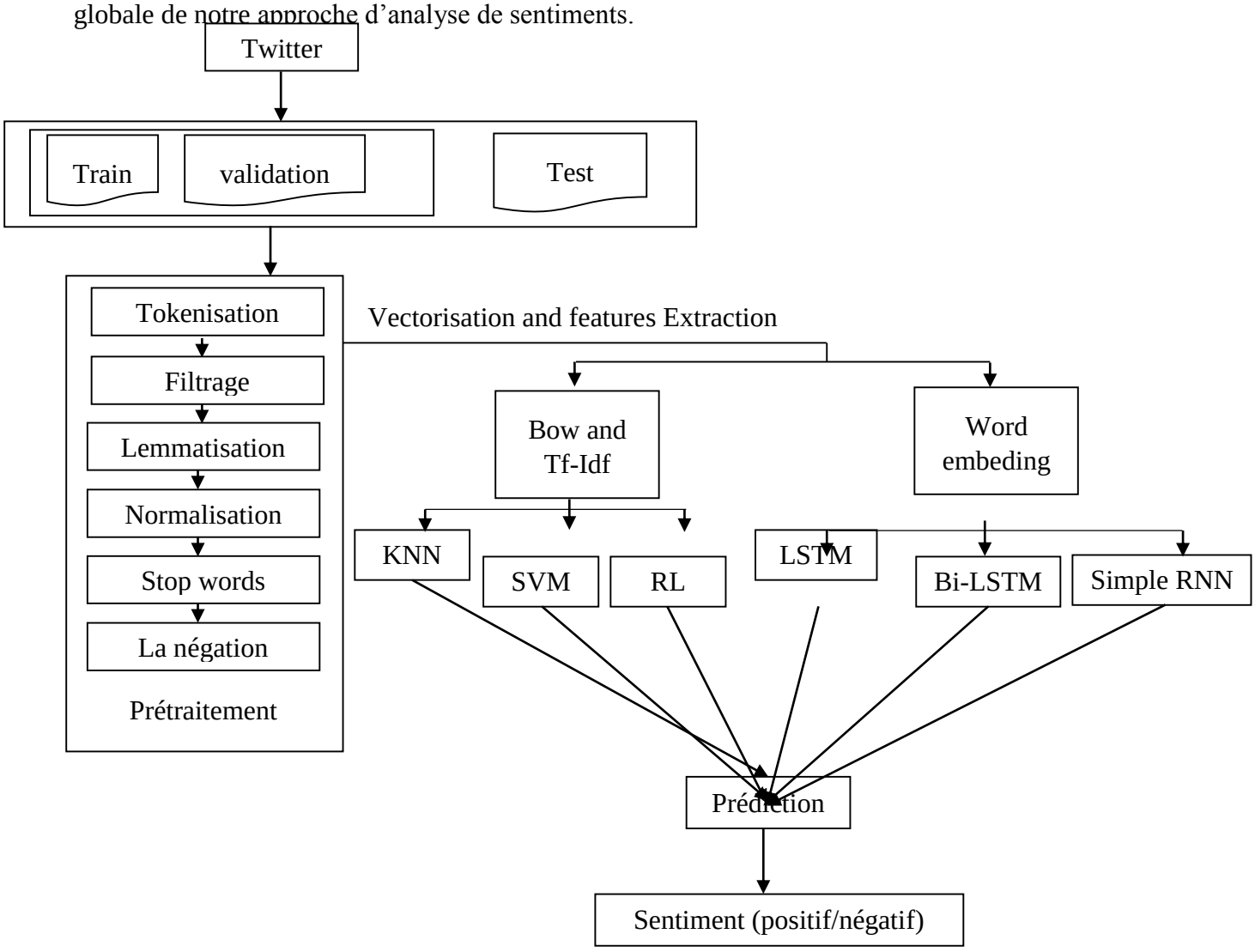


Figure 27 L'architecture de notre système d'AS.

Comme le montre la figure 25, nous commençons par le nettoyage et la préparation du Dataset (Twitter), ensuite l'extraction des caractéristiques et la vectorisation des données avant de les passer aux différents algorithmes d'apprentissage que nous allons expérimenter.

III.3. Dataset

Nous avons utilisé un Dataset de tweets annoté pour l'analyse de sentiments qui est composé de Tweets à sentiment positifs et négatifs sous forme de deux dossiers associés aux deux polarités, chaque dossier contient 1000 commentaire sous forme de fichier (.txt).

Figure 28 La distribution des tweets.

La figure 26 représente la distribution des tweets de notre Dataset qui est de 1000 positifs et 1000 négatifs.

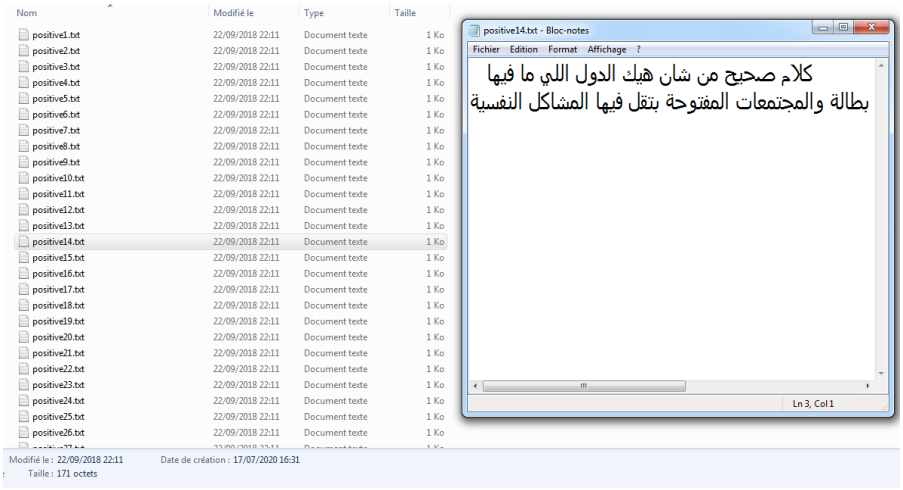


Figure 29 Exemple d'un tweet positif.

La figure 27 montre un exemple de tweet positif .

- **1ere approche (Machine Learning)**

Pour utiliser les algorithmes de Machine Learning, nous avons divisé le dataset en 2 ensembles:

- Train_set
- Test_set

Tableau 10 répartition du Dataset Pour le Machine Learning

Ensemble	Dataset(2000 comments)
Train_set (80%)	1600 Comments
Test_set (20%)	400 Comments

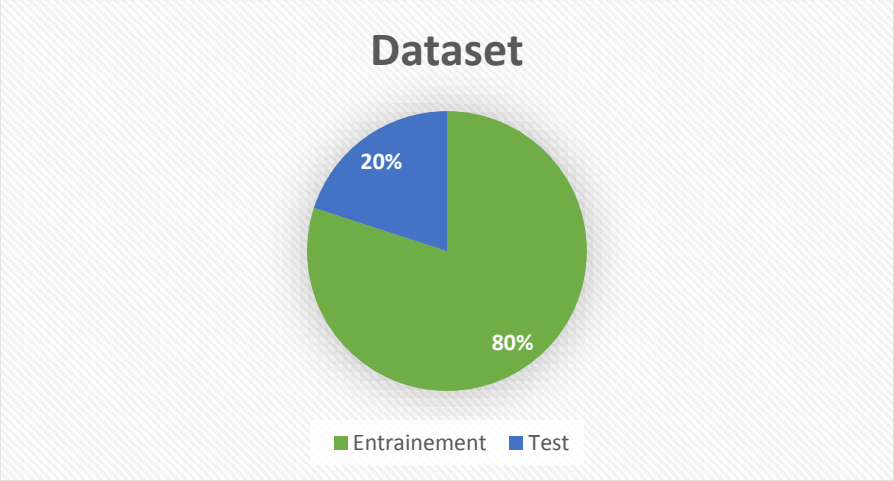


Figure 30 répartition du Dataset pour le Machine Learning

Le Tableau 10 et la figure 28 décrit la division du Dataset avec 400 commentaires pour le test, 1600 commentaires pour l'entrainement.

- **2eme approche (Deep Learning)**

Nous avons divisé le Dataset en 3 ensembles: Test_set, Train_set et Validation_set.

Tableau 11 répartition du Dataset pour le deep Learning .

Ensemble	Dataset (2000 comments)
Test_set (20%)	400 Comments
Train_set (64%)	1280 Comments
Validation_set (16%)	320 Comment

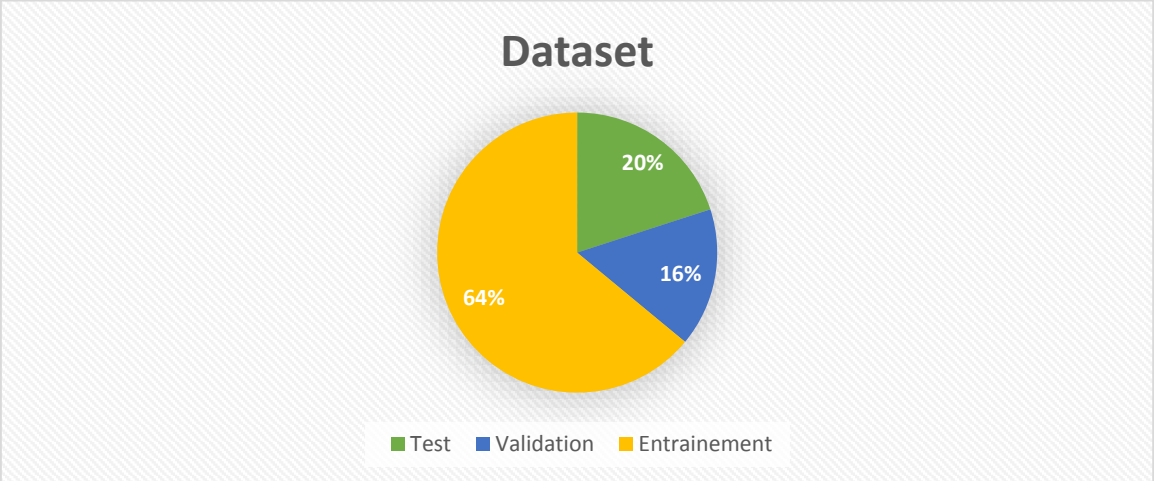


Figure 31 répartition du Dataset pour le deep Learning.

Le Tableau 11 et la figure 29 décrit la division du Dataset en trois parties, 400 commentaires pour le test, 1280 commentaires pour l'entrainement et 320 commentaires pour la validation.

III.4.Prétraitement

III.4.1 Tokenisation

Cette phase est la première étape de notre prétraitement consiste à transformer un texte en une série de tokens individuels. Dans l'idée, chaque token représente un mot.

```
tok = Tokenizer(num_words=max_features )
tok.fit_on_texts(MyDataSet.data)
```

Figure 32 code source de tokenisation.

III.4.2. Filtrage

Nous passons par la suite au filtrage de nos données ce qui veut dire on enleve tous ce qui est texte non arabe, nombre, caractéres spéciaux, diacritique etc ..

```
def remove_urls(text):
    text = re.sub(r"https?:\/\/t.co\/[A-Za-z0-9]+",'',text)
    return text

def remove_spec(text):
    text = re.sub('<.*?>+', '', text)
    return text

def remove_diacritics(text):
    regex = re.compile(r'[\u064B\u064C\u064D\u064E\u064F\u0650\u0651\u0652]')
    return re.sub(regex, '', text)

def remove_numbers(text):
    regex = re.compile(r"(\d|[\u0660\u0661\u0662\u0663\u0664\u0665\u0666\u0667\u0668\u0669])+")
    return re.sub(regex, ' ', text)
```

Figure 33 Code source du filtrage des données.

III.4.3. Normalisation

Cette phase permet de rendre tous les mots à la forme normal ce qui veut dire que si il y a un mot écris avec des caractères doublons par exemple il sera réduit a une seul lettre et les doublons seront enlevés

```
def normalizeArabic(text):
    text = re.sub("[إأآا]", "ا", text)
    text = re.sub("ى", "ي", text)
    text = re.sub("ؤ", "ء", text)
    text = re.sub("ئ", "ء", text)
    return(text)
```

Figure 34 code source de la normalisation.

III.4.4. Elimination des mots vides

En recherche d'information, un mot vide (ou stop word, en anglais) est un mot qui est tellement commun qu'il est inutile de l'indexer ou de l'utiliser dans une recherche ou comme dans notre cas l'analyse de sentiment. En arabe les mots vides pourraient être « و » , « إلى », « على » etc..

```
for word in tweet_split:
      word=word.replace(u'\ufeff', '')
      if word not in stop_words:
          tweet.append(word)
data_clean.append(tweet)
```

Figure 35 code source de l'élimination des mots vides.

III.4.5. La négation

En ce qui concerne la négation c'est un des problèmes majeur de l'analyse de sentiment car à cause de lui un sentiment positif peut devenir négatif de ce fait vu que les mots de négation sont considéré comme des mots vides comme « لا » , « ليس » etc .., pour notre part on les a pris en considération.

Tableau 12 exemple des différentes étapes de prétraitement

Etape	Phrase
Normal	"اللهم نجحنا في دراستنا و إجعل الضحكة في وجوووووه من نحب" 3 #ماستر_2 ☺ www.google.com
Filtrage	"اللهم نجحنا في دراستنا و إجعل الضحكة في وجوووووه من نحب"
Normalisation	"اللهم نجحنا في دراستنا و اجعل الضحكة في وجوه من نحب"
Elimination des mots vides	"اللهم نجحنا دراستنا اجعل الضحكة وجوه نحب"
Phrase finale	"اللهم نجحنا دراستنا اجعل الضحكة وجوه نحب"

III.5. Vectorisation et Features Extraction

III.5.1. BOW et Tf-Idf

III.5.1.1. Bag of Word

Pour commencer nous devons représenter les tweets par un vecteur, pour ce faire on fait correspondre chaque composante du vecteur-document à un mot du dictionnaire du corpus. Il s'agit d'une approche dite de bag of Word ou (sac des mots). La figure suivante illustre la fonction utilisée :

```
# # Calculating BOW

count_vector = CountVectorizer()
word_counts=count_vector.fit_transform(MyDataSet.data)
```

Figure 36 vectorisation des tweets.

III.5.1.2. Calcul du Tf-Idf

Nous avons utilisé le Tf-Idf cela correspond à un poids calculé et affecté pour chaque mot de tweets du corpus. Il se décompose en deux parties :

- La fréquence d'apparition d'un mot dans un tweet
- Le nombre de tweets dans lequel le mot apparaît une fois ou plus par rapport au nombre de tweets total du notre corpus.

```
# Calculating TFIDF
tf_transformer = TfidfTransformer(use_idf=True).fit(word_counts)
X = tf_transformer.transform(word_counts)
```

Figure 37 Calcule du Tf-Idf.

III.5.2 Word Embeding

Coté deep learning nous avons utilisé le word embedding de Keras. Cela nécessite que les données d'entrée soient codées en entier, de sorte que chaque mot soit représenté par un entier unique. Cette étape de préparation des données peut être effectuée à l'aide de l' API Tokenizer également fournie avec Keras. La couche Embedding est initialisée avec des pondérations aléatoires et apprendra une incorporation pour tous les mots de l'ensemble de données d'apprentissage. [30]

III.6. Prédiction et évaluation des modèles

Après avoir entrainé nos modèles on fais évaluer nos modèles pour ce faire on utilise différentes technique comme la justesse, la précision, et le f1-score

```
print('\n\n********************Logistic regression CLASSIFIER*************\n\n')
#LR
y_pred_LR=LR.predict(X_test)
print("\nResults pour LR...")
score_LR=accuracy_score(target_test, y_pred_LR)
print("accuracy:",score_LR*100,"%")
print("Reports:",classification_report(target_test, y_pred_LR))
```

Figure 38 Evaluation du modèle.

III.7. Conclusion

Enfin Dans ce chapitre nous avons expérimenté les deux approches (apprentissage automatique et apprentissage profond) avec les différentes étapes à suivre les méthodes que nous avons utilisées pour avoir une bonne prédiction.

Chapitre IV
Test et validation de la solution

IV.1. Introduction

Dans ce chapitre, Nous allons présenter les outils et librairie utilisés avant de décrire les différentes expérimentations que nous avons menées ainsi que les résultats obtenus.

IV.2. Les outils et librairies Utilisés

IV.2.1. Software

IV.2.1.1. Python

Python est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Parce que sa syntaxe est élégante, que son typage est dynamique et qu'il est interprété, Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes. [31]

IV.2.1.2. Anaconda

Anaconda est un gestionnaire de packages, un gestionnaire d'environnement, une distribution de science des données Python et une collection de plus de 7500 packages open source . Anaconda est gratuit et facile à installer, et il offre une assistance gratuite à la communauté .

IV.2.1.3. Spyder

Spyder est un environnement scientifique puissant écrit en Python, pour Python, et conçu par et pour des scientifiques, des ingénieurs et des analystes de données. Il présente une combinaison unique des fonctionnalités avancées d'édition, d'analyse, de débogage et de profilage d'un outil de développement complet avec l'exploration de données, l'exécution interactive, l'inspection approfondie et les superbes capacités de visualisation d'un package scientifique. [32]

IV.2.1.4. Google Colab

offert par Google on l'utilise google colab pour faciliter et économisé le matériel physique (hardware) , Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud., pour notre cas on la utilisé pour faire les épochès dans le deep learning pour minimisé le temps de traitement .

IV.2.1.5. Keras

Keras est une bibliothèque open-source de composants de réseaux neuronaux écrits en Python. 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-électronique 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 . [33]

IV.2.1.6. Scikit-Learn

Scikit-learn est une bibliothèque de clés pour le langage de programmation Python qui est généralement utilisé dans les projets d'apprentissage automatique. Scikit-learn se concentre sur les outils d'apprentissage automatique, y compris les algorithmes mathématiques, statistiques et à usage général qui constituent la base de nombreuses technologies d'apprentissage automatique. En tant qu'outil gratuit, Scikit-learn est extrêmement important dans de nombreux types de développement d'algorithmes pour l'apprentissage automatique et les technologies associées.

IV.2.1.7. Matplotlib

Matplotlib est une bibliothèque de traçage 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 tracés dans des applications Python.

IV.2.1.8. Regular expression

Ce module fournit des opérations de correspondance d'expressions régulières similaires à celles trouvées dans Perl.

IV.2.1.9. Angular

Angular est un Framework open source écrit en JavaScript qui permet la création d’applications Web et plus particulièrement de ce qu’on appelle des « Single Page Applications ».

IV.2.1.10. interface utilisateur(UI)

nous avant utilise Mterial.angular framework qui est une structure de composant d'interface utilisateur qui vous permet de produire une application à une seule page en utilisant un ensemble de composants et de directives prédéfinis.

IV.2.1.11. Postman

Postman est une extension de Google Chrome qui est un navigateur propriétaire fonctionnant sous Windows, Mac, Linux, Android et IOS. Il est conçu pour tester les web services, est gratuit, facile et rapide à manipuler. Il possède également une ergonomie simple et convivial.

IV.3. Hardware

La configuration du matériel utilisé dans notre implémentation ce devise on deux parties :

Google Colab et un PC portable avec les configuration suivante :

- Un PC portable Lenovo AMD CPU
- RAM de taille 4 Go DDR4
- Disque Dur 500 Go SSD
- Système d'exploitation Windows 10 64 bits

IV.4.Machine Learning

IV.4.1 extraction des features

IV.4.1.1. Tf-idf Vectorizer

Pour notre programme nous avons utilisé une fonction qui s'appelle « TfidfVectorizer » Cette fonction transforme la collection de documents brut en une matrice de fonctionnalités TF-IDF qui est équivalent a faire le « countvectorizer » suivi de « tfidftransformer »

IV.4.2. Algorithmes et résultats

Dans la partie machine learning nous avons utilisé 3 algorithme.

Pour cella on a utilisé la bibliothèque « Sklearn » qui contient ces algorithme que nous avons appliqué sur notre Dataset .

- **KNN :**

Dans la bibliothèque « Sklearn » on appelle la fonction **« from sklearn.neighbors import kNeighborsClassifier »**

Tableau 13 résultat obtenue avec KNN.

		Précision	Rappel	F-score	Support
Uni-gram	Négative	0.55	1.00	0.71	203
	Positive	0.97	0.17	0.29	197
			Accuracy	**0.59**	400
Bi-gram	Négative	0.54	0.86	0.66	194
	Positive	0.70	0.31	0.43	206
			Accuracy	**0.58**	400

Le Tableau précèdent montre la performance de l'algorithme K proche voisin entre les deux méthodes de n-gram testé sur 400 commentaires les résultats de l'accuracy sont plutôt pas performant .

- **Régression logistique :**

Dans la bibliothèque « Sklearn » on appelle la fonction **« from sklearn.Linear_model import LogisticRegression »**

Tableau 14 Résultat obtenue avec LR.

		Précision	Rappel	F-score	Support
Uni-gram	Négative	0.81	0.91	0.86	195
	Positive	0.91	0.80	0.85	205
			Accuracy	**0.85**	400
Bi-gram	Négative	0.81	0.92	0.86	194
	Positive	0.91	0.80	0.85	206
			Accuracy	**0.86**	400

Le Tableau précèdent montre la performance de l'algorithme LR entre les deux méthodes de n-gram testé sur 400 commentaires les résultats de l'accuracy . .

- **LSVC :**

Dans la bibliothèque « Sklearn » on appelle la fonction **« from sklearn.svm import LinearSVC »**

Tableau 15 Résultat Obtenue Avec LSVC.

		Précision	Rappel	F-score	Support
Uni-gram	Négative	0.87	0.89	0.88	203
	Positive	0.89	0.87	0.88	197
			Accuracy	**0.88**	400
Bi-gram	Négative	0.83	0.92	0.87	194
	Positive	0.91	0.82	0.86	206
			Accuracy	**0.87**	400

Le Tableau précèdent montre la performance de l'algorithme LSVC entre les deux méthodes de n-gram testé sur 400 commentaires les résultats de l'accuracy .

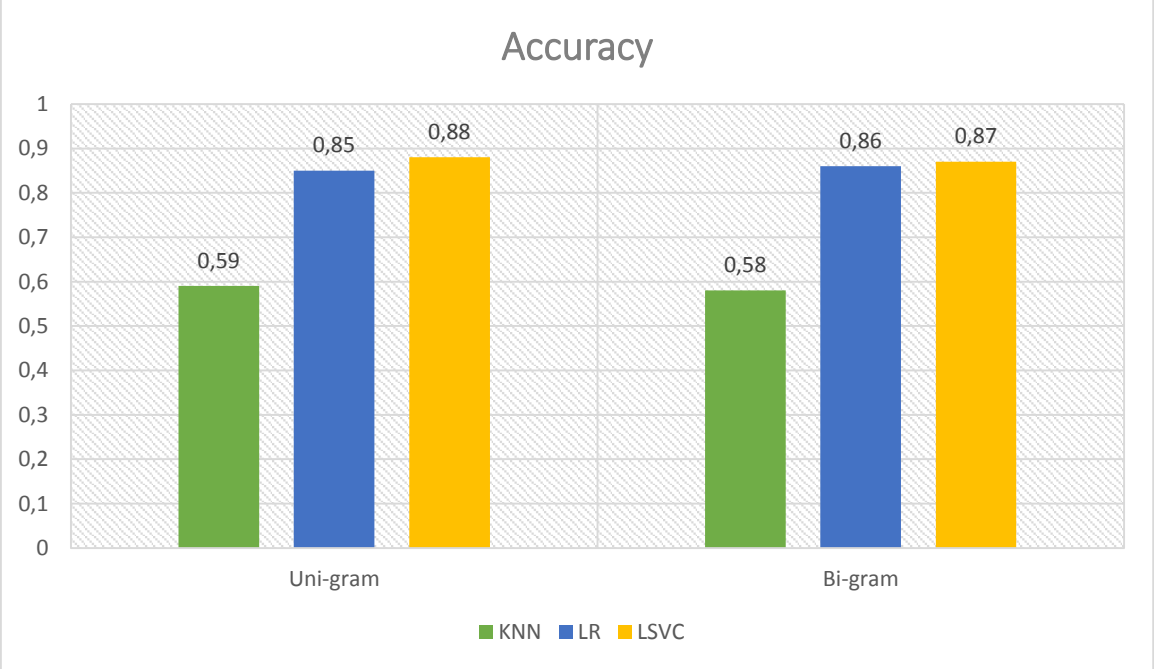


Figure 39 La comparaison des différents algorithmes uni-gram et bi-gram .

D’après ce schéma on remarque que les résultats entre bi-gram et uni-gram sont presque les même

Et que le classificateur KNN n’est pas vraiment performant et le meilleurs classificateur est le LSVC .

IV.5. Deep Learning

IV.5.1. Explication des méthodes utilisées

Tableau 16 Hyper paramètre utilisé et explication.

Hyper paramètre	Valeur	Explication
max_features	5000	Le maximum de mots entrés
num_classes	1	Nombre de classe (sigmoid), Output
max_length	50	Maximum de mots en entrée
batch_size	64	définit le nombre d'échantillons qui vont être propagés à travers le réseau.
embedding_size	16	indique la taille du vecteur d'entités (le modèle utilise des mots incorporés comme entrée)
dropout_rate	0.5	C'est une technique de régularisation (pour combattre l'overfitting).
num_epochs	40	c'est le nombre maximum d'entrainement

IV.5.2. extraction des features

IV.5.2.1.Word Embedding

Nous avons utilisé le embedding de Keras pour le mappage des mots à des vecteur de nombre réels dans un espace dimensionnel réduit , avec ça il est capable de savoir le contexte d'un mot dans un document.

IV.5.3. Résultats obtenus et discussions

IV.5.3.1. Modèle 1

pour notre premier modèle on a utilisé le model Simple RNN sur notre Dataset et aussi on a fait une comparaison une fois avec l'utilisation du Dropout et une sans l'utilisé.

Les résultats obtenue dans le model simple RNN

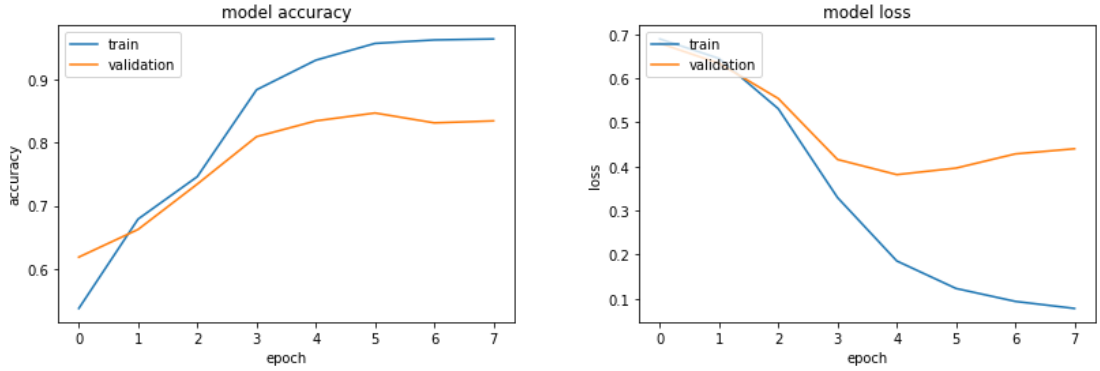


Figure 40 accuracy and loss pour le model 1 sans Dropout.

D'après la figure on remarque que la précision de l'apprentissage augmente avec le nombre d'époque jusqu'à ce stabilisé à l'époque 7, ce qui reflète qu'à chaque époque le modèle apprend de plus en plus.

La même chose pour l'erreur, l'erreur d'apprentissage diminue, d'autre part la validation augment avec le nombre d'époque.

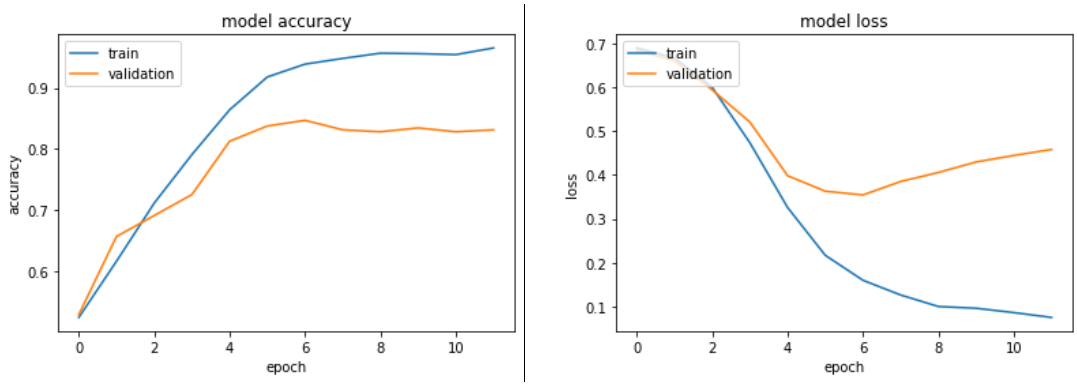


Figure 41 accuracy and loss pour le model 1 avec Dropout 0,3.

Dans la figure on remarque que l'accuracy avec le Dropout est meilleurs que sans Dropout

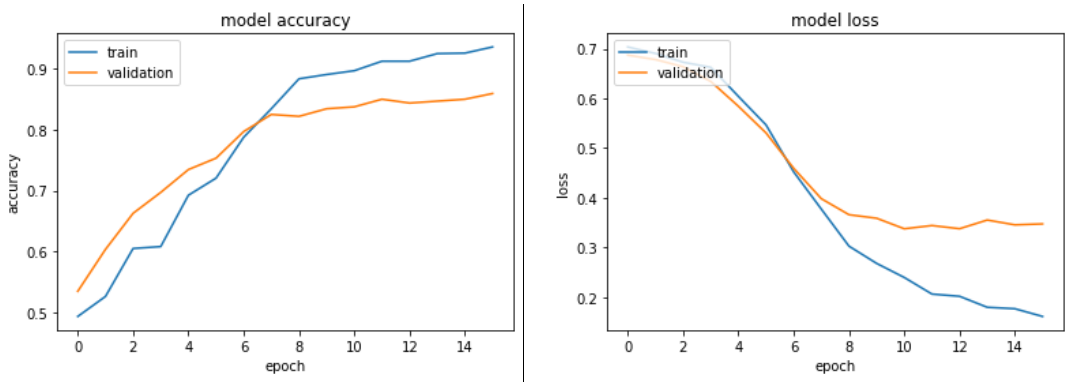


Figure 42 accuracy and loss pour le model 1 avec Dropout 0,7.

Dans la figure on remarque que l'accuracy avec le Dropout 0,7 est meilleurs que les deux cas précédent .

IV.5.3.2. Modèle 2

Le deuxième model on a fait un model LSTM et on l'a implémenter sur notre Dataset tous comme le premier model on a utilisé le dropout mais on a ajouté une autre fonction qui s'appelle earlystoping cette fonction s'arrête au nombre d'époque idéal .

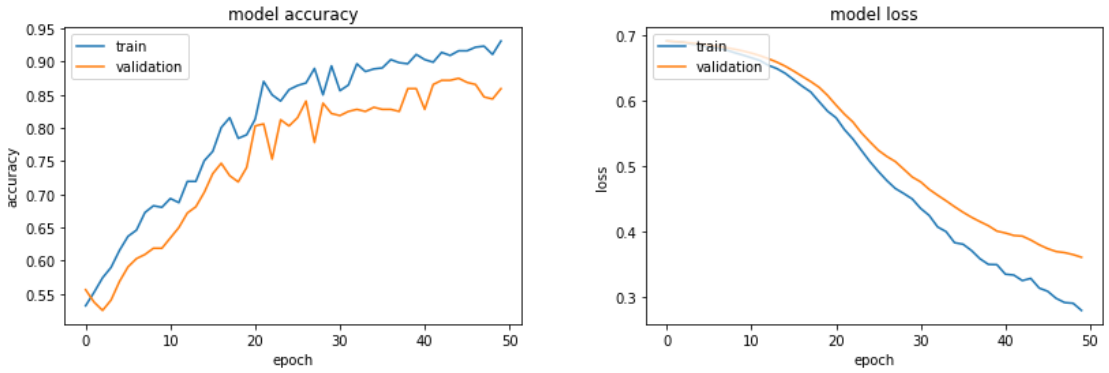


Figure 43 le model LSTM avec 50 époque sans Dropout.

D'après la figure on remarque que la précision de l'apprentissage augmente avec le nombre d'époque (à chaque époque la précision augmente), ce qui reflète qu'à chaque époque le modèle apprend de plus en plus.

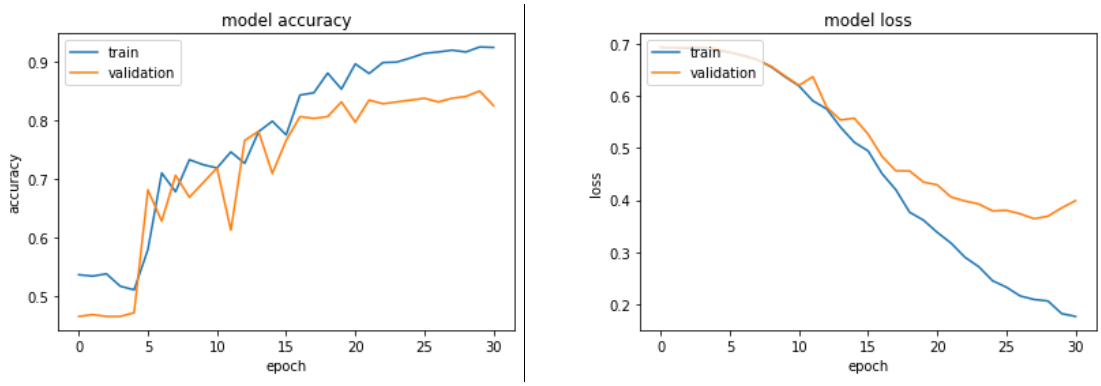


Figure 44 le model LSTM avec 50 époque avec Dropout 0,5 et early-stoping.

D'après la figure on remarque que le programme c'est arrête a l'époques 31 même si on la programmé pour faire 50 époques ce qui explique le travail de early-stoping .

IV.5.3.1. Modèle 3

Nous avons amelioré le model 2 afin d'obtenir un model LSTM bidirectionnel ce qui nous donne les résultats suivants :

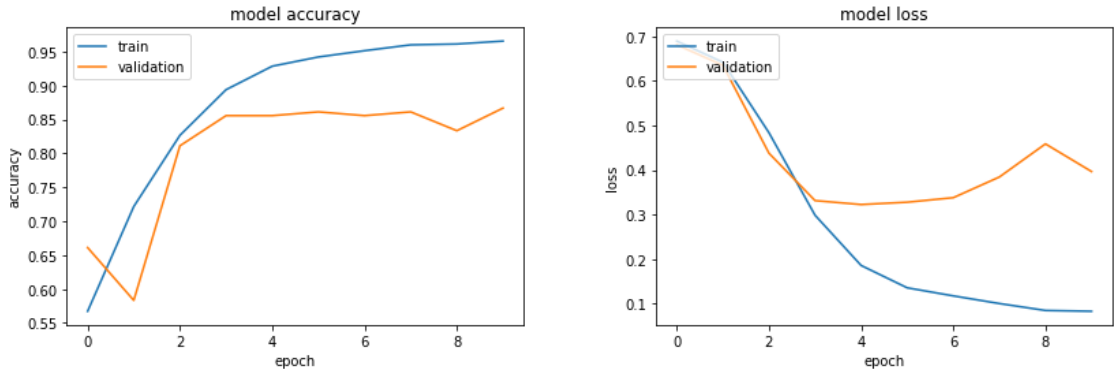


Figure 45 accuracy and loss pour le model 3 sans Dropout .

D'après l'analyse des résultats obtenus, On remarque que :

La précision de l'apprentissage et validation augmente avec le nombre d'époque, ceci reflète qu'à chaque époque le modèle apprenne plus d'informations, et de même pour l'erreur.

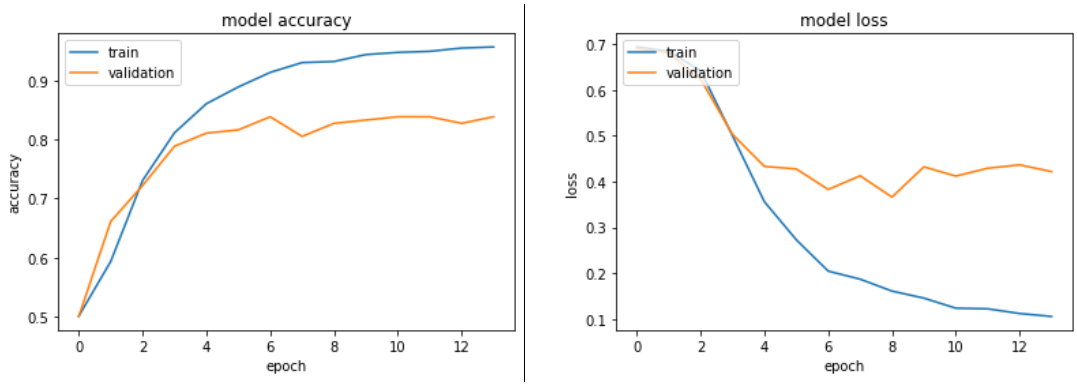


Figure 46 accuracy and loss pour le model 3 Avec Dropout 0.5.

IV.5.4. Comparaison des modèles

Tableau 17 Résultat de l'ensemble de test des diffèrents modèles.

Modèle	Option	Loss	Accuracy	F1_Score	Recall	Precision
Simple_RNN	**Avec dropout**	33%	86 %	85%	82%	88%
	Sans dropout	41%	84%	86%	87%	86%
LSTM	**Avec dropout**	32%	88%	85%	83%	87%
	sans dropout	48%	81%	76%	77%	76%
Bidirectionnel LSTM	**Avec dropout**	28%	**90 %**	89%	89%	88%
	Sans dropout	41%	84%	84%	90%	79%

On remarque dans ce tableau que le dropout a bel et bien un effet sur le modèle l'accuracy augmente toujours en utilisant le Dropout, d'après c'est résultats le meilleur modèle est celui de LSTM bidirectionnel avec un résultat excellent qui est de 90% avec un taux d'erreur de 28% seulement .

IV.5.5. Résultat final

Tableau 18 Comparaison entre les differentes approches.

Approche	Méthode	Accuracy
Machine Learning	LR	0.85
	KNN	0.59
	LSVC	0.88
Deep Learning	Simple RNN (avec Dropout)	0.86
	LSTM (avec Dropout)	0.88
	LSTM Bidirectionnel (avec Dropout)	**0.90**

D'après le tableau 18 on remarque que le meilleur model est le model LSTM Bidirectionnel avec Dropout avec une accuracy de 0.90.

IV.6. Conclusion

Dans ce chapitre nous avons donné les résultats obtenues des différentes approches avec le Dataset qui contient 2000 tweets arabe (1000 positifs et 1000 negatifs) , nous avons exploité trois classificateurs dans le machine Learning qui sont linear support vector classification (LSVC), K nearest neighbor (KNN) et logistics regression (LR) ou l'evaluation se fait par 20% du Dataset, coté Deep Learning on a fait trois modèle Simple recurrent neurone network (Simple RNN) , Long Short term Memory(LSTM) et LSTM bidirectionnel .

Nous avons utilisé plusieurs fonctionnalité par exemple le Dropout toute en faisant une comparaison avec sans les fonctionnalités et l'impact qu'ils font.

Conclusion Générale

Conclusion Générale

La croissance dans le domaine de l'analyse des sentiments a été très rapide et vise a exploiter les opinions ou textes présent sur les diffèrent plateforme de media a partir des technique d'apprentissage automatique et apprentissage profond

Tout d'abord nous avons vu les différente technique déjà existante qui ont déjà été implémenté ainsi que leurs résultat et leurs configurations varié.

Ensuite nous avons discuté les notions fondamentales des approches utilisé pour ce qui est de l'apprentissage automatique nous avons vu trois algorithmes ainsi que leurs fonctionnement , coté apprentissage profond nous avons discuté les notions des réseaux de neurones en générale et réseaux LSTM en particulier , nous avons aussi parlé aussi des méthodes varié de régularisation .

Pour après présenté la modélisation de notre system d'analyse des sentiments qui passe par plusieurs étapes de prétraitement pour après transformer notre dataset en une représentation numérique cette phase s'appelé features extractions

Enfin pour clôturer notre mémoire nous avons implémenté c'est différente méthode pour enfin les faire comparer, nous avons obtenu un excellent résultat grâce au LSTM bidirectionnel qui donne une bonne accuracy .

Comme perspectives nous pouvons citer :

- Augmenter le nombre de donnée pour minimiser l'erreur
- Développer le modèle pour être plus précis dans la détection de la négation.
- Développer d'autre modèle comme Tranfer learning.
- Développer le modèle pour traiter le problème du sarcasme.

Bibliographie

Bibliographie

[1] **Olivier Ezratty**

«mercator-publicitor,» [En ligne] https://www.mercator-publicitor.fr/lexique-marketing-definition-reseaux-sociaux#:~:text=R%C3%A9seaux%20sociaux%20%2D%20Social%20media,groupes%20d'individus%20ou%20organisations.. [Accès le 15 08 2020].

[2] **Steve Dawson**

«top site,» 13 avril 2020. [En ligne]. https://www.alexa.com/topsites. [Accès le 15 08 2020].

[3] **Palo Alto, Calif**

«Press Releases: LinkedIn Premium Services Finding Rapid Adoption,» 7 mars, 2006 [En ligne]. [Accès le 16 08 2020].

[4] **F. Ropars,**

«Instagram Direct : une messagerie pour envoyer des photos privées,» 29 decembre 2017. [En ligne]. https://www.blogdumoderateur.com/instagram-annonce-messagerie/. [Accès le 16 08 2020].

[5] **Marc Granovetter**

http://nmstpe.over-blog.com . 24 janvier 2016. [En ligne]. [Accès le 16 08 2020].

[6] **V. G. Amandeep Kaur**

«A Survey on Sentiment Analysis and Opinion Mining Techniques,» *journal of emerging technologies in web intelligence*, pp: 367-371. vol. 5, no. 4, 2013.

[7] **Asad Bukhari**

«critical review of sentiment analysis techniques,» *proceeding of the international Conference on Artificial Intelligence and Computer Science*, (AICS 2014), e-ISBN 978-967-11768-3-2, September 2014.

[8] **Walaa Medhat , Ahmed Hassan , Hoda Korashy**

«Sentiment analysis algorithms and applications,» *Ain Shams Engineering Journal*, pp. 1093-1113, 2014.

[9] **A. Cornuéjols, L. Miclet, Y.Kodratoff**

«Apprentissage Artificiel, Concepts et algorithmes,» *ISBN*, pp. 2-212-11020-0, 2002.

[10] **M'hamed Mataoui et Omar Zelmati et Madiha Boumechache**

A Proposed Lexicon-Based Sentiment Analysis Approach for the Vernacular Algerian Arabic, 2016, pp. 55-68.

[11] **A. Sieg**

«Text Similarities: Estimate the degree of similarity between two texts,» 04 juillet 2018 [En ligne]
https://medium.com/@adriensieg/text-similarities-da019229c894.
[Accès le 25 3 2020].

[12] **Salima Mdhaffar, Fethi Bougares, Yannick Esteve, Lamia Hadrich-Belguith**

«Sentiment Analysis of Tunisian Dialect: Linguistic Resources and Experiments, Proceedings of The Third Arabic Natural Language Processing Workshop (WANLP),» Valencia, Spain, 2017.

[13] **Abdeljalil Elouardighi, Mohcine Maghfour, Hafdalla Hammia, Fatima-Zahra Aazi**

«Analyse des sentiments à partir des commentaires Facebook publiés en Arabe standard ou dialectal marocain par une approche d'apprentissage,,» chez *18ème édition de la conférence Internationale sur l'Extraction et la Gestion des Connaissances,*, Paris, France, 2018.

[14] **J. Littman**

«Collecte de données Facebook avec l'API Graph,» 25 3 2020. [En ligne].
https://gwu-libraries.github.io/sfm-ui/posts/2018-01-02-facebook.
[Accès le 25 7 2020].

[15] **Shahzad Qaiser, Ramsha Ali**

«Use of TF-IDF to Examine the Relevance of Words to Documents,» *International Journal of Computer Applications*, pp. 25-29, 2018.

[16] **Sghaier, Mohamed & Abdellaoui, Houssem & Ayadi, Rami & Zrigui, Mounir**

Analyse de sentiments et extraction des opinions pour les sites e-commerce. application sur la langue arabe., 2014.

[17] **Amira Barhoumi, Nathalie Camelin, Chafik Aloulou, Yannick Estève, Lamia Hadrich Belguith**

«Toward Qualitative Evaluation of Embeddings for Arabic Sentiment Analysis. International,» chez *Language Resources and Evaluation (LREC2020)*, Marseille, France., May 2020.

[18] **Oliver Theobald**

« Machine Learning For Absolute Beginners: A Plain English Introduction (Second Edition) »

[19] **S. Raschka**

Python Machine Learning, Birmingham: packet publishing , 2015 p 55.

[20] **A. Mohapatra**

«Logistic Regression from Scratch: Multi classification with OneVsAll,» 25 janvier 2020. [En ligne]. https://medium.com/analytics-vidhya/logistic-regression-from-scratch-multi-classification-with-onevsall-d5c2acf0c37c. [Accès le 15 aout 2020].

[21] **S. raschka**

«python machine learning,» chez *unlock deeper insight into machine learning with this vital guide to cutting edge predictive analytics* , birmingham, packet publishing, 2016, p. 56.

[22] **W. S. Noble**

«What is a support vector machine?,» *Nature biotechnology*, vol. 12, n° %11565-1567, p. 24, 2006.

[23] **Mohamadally Hasan, Fomani Boris**

«SVM : Machines à Vecteurs de Support,» *BD Web*, *ISTY3*, 16 janvier 2006.

[24] **S. C. Wang**

«Artificial neural network.,» chez *Interdisciplinary computing in java programming (pp. 81-100)*, Springer, Boston, MA., 2003.

[25] **Ramachandran, P., Zoph, B., & Le, Q. V**

« Searching for activation functions,» p. arXiv preprint arXiv:1710.05941., 2017.

[26] **S. JAIN**

«Un aperçu des techniques de régularisation dans le Deep Learning,» 19 avril 2018. [En ligne]. https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/. [Accès le 20 aout 2020].

[27] **Zaremba, W., Sutskever, I., & Vinyals, O**

«Recurrent neural network regularization,» p. arXiv preprint arXiv:1409.2329., 2014.

[28] **Shahzad qaisier, Ramsha Ali**

«text minig : use of TF IDF to examine the revelance of words to documents,» *international journal of computer applications* , pp. 25-29, 2018.

[29] **Jason Brownlee**

«What Are Word Embeddings for Text?,» 11 octobre 2017. https://machinelearningmastery.com/what-are-word-embeddings/. [Accès le 15 juillet 2020].

[30] **J. Brownlee**

«Deep Learning for Natural Language Processing,» 04 octobre 2017. [En ligne]. https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/. [Accès le 15 aout 2020].

[31] **Python Software Foundation**

«Le tutoriel Python,» [En ligne]. https://docs.python.org/fr/3/tutorial/. [Accès le 02 septembre 2020].

[32] **MIT License**

«Bienvenue dans la documentation de Spyder,» [En ligne]. https://docs.spyder-ide.org/current/index.html. [Accès le 02 septembre 2020].

[33] **Chollet, François et Autres**

«Qu'est-ce que Keras?,» [En ligne]. https://deepai.org/machine-learning-glossary-and-terms/keras. [Accès le 02 septembre 2020].