Ministere De L'enseignement Superieur Et De La recherche
Scientifique
Université de Mohamed Khider - BISKRA
Faculté des sciences exactes,des sciences naturelles et de la vie
Département d'informatique

Mémoire

Présenté pour l'obtention du diplôme de Master académique en

Informatique

Option: Génie Logiciel et Systèmes Distribués

Analyse des sentiments en utilisant l'apprentissage Profond: Cas de la langue Arabe

---

Réalisé par :
Chaima KIHEL

Soutenu le ../../.... devant le jury :

MCA	Président	
Okba TIBERMACINE	MCB	Encadrant
MAA	Examinateur	

Année universitaire: 2019/2020

dédicace

Je dédie ce travail à moi-même.

Remerciement

Tous d'abord, je tiens à remercier le bon Dieu de nous avoir accordé toute la détermination, la volonté et la force pour qu'on puisse réaliser ce modeste travail.

Je voudrais également être reconnaissante infiniment à mon encadreur **Dr.TIBERMACINE Okba** pour ses conseils, sa patience, sa disponibilité et son soutien tout au long de cette période.

Je voudrais exprimer ma reconnaissance envers mes parents **Abd ELmadjid** et **Rania** ma sœur et mes frères, qui m'ont apporté leur support moral et les discussions animées tout au long de ma démarche ,et ma grand-mère **Rachida** qui m'a toujours accompagné de ses prières.

Nous tenons à exprimer notre profonde gratitude et nos sincères remerciements aux membres de jury d'avoir accepté de juger notre travail et de l'avoir enrichi.

Résumé

L'analyse de la subjectivité et des sentiments a récemment attiré une attention considérable, mais la plupart des ressources et des systèmes construits jusqu'à présent sont adaptés à l'anglais et à d'autres langues indo-européennes. Le besoin de concevoir des systèmes pour d'autres langues augmente, d'autant plus que les sites de blogs deviennent populaires dans le monde entier. Pourtant, les recherches dans le domaine de l'analyse du sentiment arabe progressent à un rythme très lent par rapport à celles menées en anglais et dans d'autres langues. Ce travail présente également une étude de cas dont le but est d'étudier la possibilité de déterminer l'orientation sémantique des expressions et des commentaires arabes compte tenu des ressources arabes limitées.

- Mots clés : Analyse des sentiment, opinion mining, traitement automatique de la langue, apprentissage profond ...

Abstract

The analysis of subjectivity and Senti,ents has gained a considerable attention in last decade. Unfortunately, most of the resources and systems constructed so far are suitable for English and other Indo-European languages. The need to design systems for other languages is in increase demande in respect to the amount of bloggind and social media sites available online. However, research in the area of Arab sentiment analysis is advancing at a very slow pace compared to that conducted to the English and other languages. This work also presents a case study aimed at investigating the possibility of determining the semantic orientation of Arabic expressions and commentaries given the limited Arabic resources.

- Keywords: Sentiment analysis, Opinion mining, Natural Language Processing, Deep Learning ...

Contents

List of Figures

Introduction générale

Contexte

Avec la croissance rapide des applications Web et des médias sociaux, il y a eu des critiques, des commentaires, des évaluations et des commentaires générés par les utilisateurs. Ces opinions peuvent concerner pratiquement n'importe quoi, y compris les produits, la politique, les actualités, les personnes, les services et les événements. Tout cela doit être traité et analysé pour obtenir une bonne estimation de ce que pense et ressent l'utilisateur. Avant la disponibilité d'outils automatiques d'analyse des sentiments, le processus d'obtention des avis des clients était une tâche extrêmement lourde et longue. Ceci explique probablement le grand intérêt de ce domaine de recherche [1].

l'analyse des sentiments peut aider à créer un portrait plus riche et à éclairer vos futures stratégies de contenu et de médias sociaux. L'analyse des sentiments, d'autre part, peut être considérée comme un ensemble d'algorithmes mis en œuvre dans un logiciel informatique qui détecte et exploite les opinions et les émotions dans les ressources des médias sociaux en ligne. C'est un domaine interdisciplinaire qui emprunte des techniques de traitement du langage naturel, d'analyse de texte et de linguistique informatique pour extraire des informations subjectives [2].

De nombreux outils d'analyse des sentiments ont été développés pour l'anglais, mais nous essayons de percer un nouveau terrain dans ce domaine et de proposer un outil d'analyse des sentiments basé sur la langue Arabe qui n'est pas affecté par l'utilisation des dialectes; un outil qui donne aux utilisateurs arabes le pouvoir d'analyser les médias sociaux, leur donnant la possibilité de connaître le sentiment général sur les sujets d'actualité en cours de discussion. La langue arabe a de nombreux dialectes qui doivent être pris en compte, où dans chaque dialecte les significations des mots peuvent être totalement différentes.

Objectif

Dans ce projet, notre visons à explorer le domaine de l'analyse des sentiments dans les textes Arabes en utilisant particulièrement les techniques d'apprentissage Profond (Deep Learning), Car ces dernières ont données des résultats remarquables dans la langue Anglaise. Par conséquent, On vise à développer un modèle de Deep learning et de l'appliquer sur un corpus de données Arabes collectés à partir des réseaux sociaux. On utilise dans notre proposition le modèle Flaire pour présenter la sémantique de chaque mot lors de processus d'apprentissage.

Organisation

Ce manuscrit est organisé quatre chapitre à savoir :

Chapitre 1: Le premier chapitre présente le problème de l'analyse du sentiment, ses différents niveaux, ses approches et les travaux connexes.

Chapitre 2: Le second chapitre sera une description de l'apprentissage approfondi, de ses applications et de ses fondements.

Chapitre 3: Le troisième chapitre sera une description de nos modèles proposés, les outils utilisés et les résultats obtenus.

Chapitre 4: Le quatrième chapitre est destiné à l'évaluation du modèle proposé en Conception. Nous commençons par la description des outils utilisés, ainsi que les expérimentations réalisées et les résultats obtenus.

Conclusion : Le présent manuscrit terminera par une conclusion générale.

Chapter 1

Traitement Automatique de la Langage Arabe et l'analyse des Sentiments

1.1 Langue Arabe

L'Arabe est une langue officielle de 22 pays, parlée par plus de 300 millions. Elle est la langue qui connaît la croissance la plus rapide sur le Web. Il y a environ 65 millions Utilisateurs arabophones en ligne, soit environ 18,8 % de la population Internet mondiale [4]. L'arabe est une langue sémitique [5] et se compose de nombreux dialectes régionaux différents. Bien que ces dialectes soient de véritables formes de langue maternelle, ils ne sont généralement utilisés la communication quotidienne informelle et ne sont pas standardisés ou enseignés dans les écoles [6].
)Il existe une norme écrite formelle qui est couramment utilisée dans les médias écrits et l'éducation dans le monde arabe, appelée Arabe Standard Moderne (ASM). Il y a une grande différence entre ASM et la plupart des dialectes arabes et, fait intéressant, ASM n'est en fait la langue maternelle d'aucun pays ou groupe arabe. ASM est syntaxiquement, morphologiquement et phonologiquement basé sur l'Arabe Classique (AC) [6], qui est la langue du Coran (le livre saint de l'Islam). L'arabe a un système d'inflexion très riche et est considéré comme l'une des langues les plus riches en termes de morphologie [7]. Dans la langue arabes il y a 2 constructions : nominales et verbales [8]. Dans le domaine verbal, l'arabe a deux modèles d'ordre des mots (c'est-à-dire, sujet-verbe-objet et verbe-sujet-objet). Dans le domaine nominal, un le modèle normal serait composé de deux mots consécutifs, un nom (c'est-à-dire, sujet) puis un adjectif .

1.2 l'analyse des sentiments

L'analyse du sentiment, aussi appelée opinion mining, est un sous-domaine de l'informatique, il est considéré comme une partie du traitement automatique du langage naturel et a pour but de classifier les sentiments exprimés dans des textes.Il existe de nombreux autres noms apparentés se référant légèrement à différentes tâches telles que l'extraction d'opinion, le sentiment mining, l'analyse de subjectivité, l'analyse d'affect, l'analyse d'émotion, les avis mining, etc.
L'analyse des sentiments est généralement effectuée en utilisant l'une des deux approches de base: les classificateurs basés sur des règles, dans lesquels les règles dérivées de l'étude linguistique d'une langue sont appliquées à l'analyse des sentiments [9, 10, 11] et les classificateurs d'apprentissage automatique dans lesquels la machine statistique des algorithmes d'apprentissage sont utilisés pour apprendre automatiquement des signaux de sentiment [12, 13].

Nous devons donc définir les deux termes sentiment et opinion, car dans la littérature et dans le domaine actif de la recherche, il existe une confusion entre ces deux mots. Dans le dictionnaire Oxford[14], le sentiment est défini comme un point de vue ou une opinion qui est détenu ou exprimé et comme une émotion. Pour le mot opinion, il s'agit d'une croyance ou d'un jugement formé sur quelque chose, qui n'est pas nécessairement fondé sur des faits ou des connaissances, mais sur les croyances ou les opinions d'un groupe ou de la majorité des gens.On peut dire que le terme sentiment est plus l'émotion d'une personne à propos quelque chose, où le terme opinion représente ou montre le point de vue d'une personne. Selon Bing Liu[1] :

"L'analyse du sentiment, aussi appelée opinion mining, est le domaine d'étude qui analyse les opinions, les sentiments, les évaluations, les attitudes et les émotions des gens envers les entités et leurs attributs exprimés dans un texte écrit."

L'analyse des sentiments est une démarche principalement basée sur le text mining donc elle peut porter sur des verbatims issus des réseaux sociaux, des avis, forums etc. Les sentiments sont généralement classés en trois types : négatifs, neutres et positifs.il peut s'agir d'une classification multiple, où le sentiment peut avoir une étiquette neutre ou même une étiquette différente comme très positive, positive, neutre, négative, très négative, les étiquettes peuvent aussi être associées à des émotions comme la tristesse, la colère, le bonheur, etc.

1.3 Catégorisation des sentiments

Les phrases peuvent être soit objectives ou subjectives. Lorsqu'une phrase est objective, aucune autre tâche fondamentale n'est requise. Quant à une phrase subjective, on peut étudier sa polarité (positive, négative ou neutre) comme montré dans la figue 1.1 .

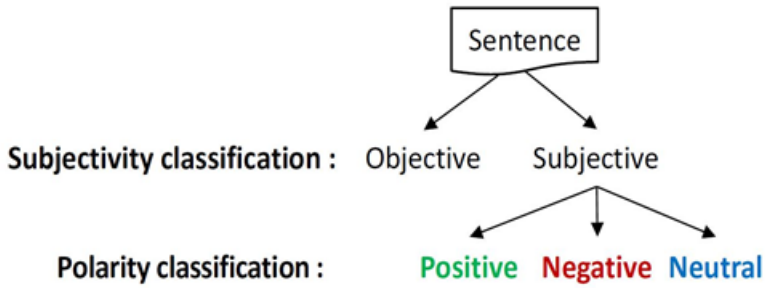

Figure 1.1: La classification des polarités

- La classification de subjectivité est la tâche qui distingue les phrases exprimant des informations objectives des phrases exprimant des vues et opinions subjectives.
- La classification de polarité est la tâche qui distingue les phrases qui expriment des polarités positives, négatives ou neutres.

exemple :
L'iphone est un smart-phone - *phrase objective*
L'iphone est génial - *phrase subjective*

1.4 Niveaux d'analyse du sentiment

Le premier choix lorsqu'on applique l'analyse des sentiments est de définir le texte qui va être analysé dans le cas d'une étude. En général, il existe trois niveaux d'analyse : le niveau du document, le niveau de la phrase et le niveau des aspects [15] :

1.4.1 Niveau du document

Détermine la polarité d'un texte entier. L'hypothèse est que le texte n'exprime qu'une seul opinion sur une seule entité (par exemple: un seul produit).Ce niveau est la forme la plus simple de classification. [16]

1.4.2 Niveau de la phrase

À ce niveau, l'unité d'information traitée est une phrase , en considérant que Chacun d'entre eux porte une opinion unique. Ce niveau d'analyse est considéré comme une classification de la subjectivité.

1.4.3 Niveau des aspects

Ici , on effectue une analyse plus fine , plus poussée et de meilleure qualité que les autres niveaux.L'étape de base de ce niveau consiste à identifier et à extraire les entités cibles . Par exemple, la phrase «L'iPhone est très bon, mais il faut encore travailler sur la durée de vie de la batterie et les problèmes de sécurité» évalue trois aspects : iPhone (positif), la durée de vie de la batterie (négatif) et la sécurité (négative) - découvre ce que les gens aiment et n'aiment pas exactement - . [15]

1.5 Applications de l'analyse du sentiment

Comme le rappellent Pang et Lee dans « Opinion Mining and Sentiment Analysis »[17] , « ce que les autres pensent » est régulièrement convoqué dans tout processus décisionnel. Elle peut trouver de nombreuses applications dans le domaine de la prédiction et de la supervision.Nous mentionnons brièvement quelques applications ci-dessous:

1. **En tant que sous-partie des systèmes :** Les systèmes d'analyse du sentiment peuvent servir une amélioration pour d'autres systèmes comme les systèmes de recommandation. De même, en ce qui concerne les systèmes d'extraction de l'information, il a été démontré que nous pouvons les améliorer et les rendre plus performants en coupant l'information dans des phrases subjectives [18]. Les systèmes de réponse aux questions sont un autre domaine où l'analyse du sentiment peut être bénéfique [19].

2. **L'utilisation par les industries et les gouvernements et dans la politique:** Aujourd'hui, les acteurs politiques ont suivi la tendance de l'analyse des sentiments ,car il est hautement stratégique de connaître également l'opinion des internautes sur un politicien lors d'une élection présidentielle par exemple ou par les gouvernements aussi sur les décisions ou quelque chose de particulier comme les élections par exemple. En 2016, l'élection présidentielle américaine. Donald Trump a dit [20] : "Je doute que je serais ici sans les médias sociaux, pour être honnête avec vous.".une étude montre une forte corrélation entre les estimations basées sur des données provenant de **Google Trends** et le résultat de plusieurs élections [21]. De même, pour d'industries où l'analyse du sentiment fait partie de leurs stratégies de planification pour la prise de décision,les entreprises comme Google, Microsoft, Hewlett-Packard, Amazon, Oracle, et Adobe ont construit ou sont en train de construire leurs propres systèmes d'analyse du sentiment.

3. **Autres domaines d'application :** Les applications d'analyse du sentiment se sont étendues à presque tous les domaines possibles, comme l'économie dans une perspective d'amélioration des produits et d'augmentation de la ventes et revenus pour les entreprises. Aussi l'éducation

,les résultats de l'analyse des sentiments aident les enseignants et les établissements à prendre des mesures correctives en ce qui concerne la méthodologie d'enseignement et le programme du cours.

1.6 Approches de l'Analyse des Sentiments

Il faut tout d'abord expliquer la différence entre la méthode d'apprentissage supervisé et celle non supervisé.
La première implique la présence de deux ensembles de données, un ensemble d'entraînement et un ensemble de test, La méthode s'appelle supervisée puisque le système est entraîné sur un sous-ensemble d'apprentissage qui contient des modèles déjà traités La deuxième méthode (non supervisée) ne recommande qu'un seul ensemble de données, elle exige que le système de façon autonome restructure les informations d'une façon que les données les plus similaires soient dans le même groupe.
Il existe 3 approches d'analyse : approches basée sur lexique, approches basée sur corpus et Approche hybride (Approches mixte). [22]

1.6.1 Approche basée sur lexique

Elle permet d'identifier la polarité d'un texte à l'utilisation de deux ensembles de mots, ceux qui expriment un sentiment positif et ceux qui expriment un sentiment négatif. Le modèle compte dans le texte le nombre de mots positifs et le nombre de mots négatifs, la somme donne une évaluation globale du sentiment de texte, si le nombre de mots positifs l'emporte sur celle de mots négatifs, le texte considèré comme positif, inversement, le texte est considéré comme négatif, éventuellement neutre si les nombres sont égaux [22].Exemple d'algorithme : Support Victor Machine (SVM)...

1.6.2 Approche basée sur corpus

L'analyse automatique des sentiments basée sur des corpus requiert l'élaboration de deux corpus annotés manuellement. Le premier corpus constitue le corpus d'apprentissage qui s'utilise afin d'entraîner un système automatique. Il comporte des notes ajoutées par des annotateurs humains.A partir de ces notes, le système devrait être capable de procéder à une analyse de façon autonome. Le deuxième corpus sera utiliser le corpus de test.Ce dernier est formé afin de vérifier la performance du système automatique. Dans un scénario idéal, les résultats de l'analyse faite par le système automatique correspondraient cent pour cent avec ceux du corpus d'apprentissage. Afin que la performance du système automatique soit maximale, il importe que le corpus d'apprentissage soit représentatif pour le corpus de test.[22] Exemple d'algorithme : réseau

de neurone...

1.6.3 Approche hybride

Cette approche tire profit des deux méthodes précédentes il y a trois façon de faire. La première est d'exploiter les outils linguistiques pour élaborer le corpus puis classer les textes par un outil d'apprentissage supervisé. La deuxième façon est d'utiliser l'apprentissage automatique pour établir le corpus d'opinion nécessaire à l'approche basée sur lexique.
La troisième façon est la combinaison des deux approches précédentes et le conjointement de leurs résultats [23].
Exemple d'algorithme : SVM (Semi-Supervised Support Vector Machine).

1.7 Travaux connexes

Bien que le domaine de fouille d'opinion et d'analyse de sentiments soit un domaine émergent dans la communauté de traitement automatique des langages naturels, les travaux effectués pour la langue arabe sont encore très limités. En effet, la plupart des travaux se concentrent sur la classification de polarité des documents pour éviter le coût élevé de l'annotation des phrases, et adoptent aussi l'approche basée sur l'apprentissage automatique pour échapper du coût élevé de la création d'un lexique d'opinion ayant une bonne couverture. Ainsi, l'obstacle qui ralentit l'avancement du domaine de l'analyse des sentiments pour l'arabe est la disponibilité des ressources en termes de corpus annotés et lexiques d'opinion.
Dans cette section, nous passons en revue les travaux effectués dans le domaine de l'analyse de l'opinion en arabe.

- Elhawary et Elfeky[26] sont parmi les premiers chercheurs qui se sont intéressés à construire un lexique d'opinion pour la langue arabe. Leurs travaux ont été dans le cadre de développement d'un outil de classification des avis des intervenants dans le domaine des affaires. L'ensemble des mots de départ comptait plus que 600 mots positifs, 900 mots négatifs et 100 mots neutres. Les tests d'évaluation effectués ont montré une précision relativement bonne mais un faible rappel.

- El-Halees [27] a proposé une approche combinée basée sur trois étapes pour la classification de documents arabes selon leurs polarités.Il utilise un lexique d'opinion construit à partir de Le lexique anglais Sentistrength après avoir fait la traduction et des dictionnaires en ligne permettant d'enrichir le lexique par les synonymes des mots déjà existants. La taille du

lexique n'a pas été mentionnée, et l'exactitude de l'outil de classification pour l'étape basée sur le lexique est de 48.7%.

- Dans [28], Abdul-Mageed et Diabont ont présenté le lexique "Sifaat", un lexique construit manuellement et contenant 3325 adjectifs annotés selon les trois classes positive, négative et neutre. Les résultats de l'évaluation montrent une amélioration de 6% dans la classification de subjectivité et 40% dans la classification de polarité. Les auteurs ont proposé aussi d'étendre le lexique par la traduction de trois lexiques d'opinion anglais à savoir, SentiWordNet, Youtubelexicon et General Inquirer. Mais cette méthode, concentrée seulement sur les adjectifs, n'a pas été évaluée.

- Abdulla et al. [29] ont proposé de créer un lexique en traduisant 300 mots de Sentistrength, puis en l'enrichissant par les synonymes et les émoticons. La version finale du lexique comprenait 3479 entrées dont 1262 positives et 2217 négatives. Les résultats de l'expérimentation sur le corpus collecté ont atteint 59.6% en terme d'exactitude.

- Alhazmi et al. [30] ont proposé de construire une version arabe de SentiWordNet en passant par deux étapes : mettre à jour la base de WordNet arabe 2.0 en faisant le mappage vers WordNet Anglais 3.0, et faire le mappage de la base obtenu vers SentiWordNet Anglais 3.0. L'évaluation de la couverture du lexique reporte que 5% des mots du corpus annoté ne sont pas dans le lexique.

Chapter 2

Application des techniques d'apprentissage profond en NLP

2.1 introduction

En 2005, les choses ont commencé à basculer. Les perspectives dans le domaine de l'intelligence artificielle ont radicalement changé avec l'apprentissage automatique et l'émergence de l'« apprentissage profond »qui représente la plus grande part de la recherche effectuée par des spécialistes, en particulier dans la mesure où elle intervient dans plusieurs domaines tels que le traitement du langage naturel.

2.2 Apprentissage profond

2.2.1 Définition

L'apprentissage profond (DL : Deep Learning en anglais) est une branche de l'apprentissage automatique (ML : Machine Learning en anglais), ce dernier est une branche de l'intelligence artificielle[24],où les machines peuvent apprendre par l'expérience et acquérir des compétences sans implication humaine.En se basant sur les réseaux de neurones artificiels, des algorithmes inspirés du cerveau humain, apprennent à partir de grandes quantités de données. Nous illustrons cette imbrication entre les trois branches par la figure :

" Le Deep Learning permet à des modèles composés de plusieurs couches de traitement d'apprendre des représentations des données avec de multiples niveaux d'abstraction."[25]

2.2.2 L'apprentissage profond vs l'apprentissage automatique

2.2.2.1 Apprentissage automatique

L'apprentissage automatique (ou apprentissage machine, Machine Learning en anglais) est un sous-domaine de l'intelligence artificielle (IA) qui se concentre sur la conception de systèmes qui apprennent — ou améliorent le rendement — en fonction des données qu'ils consomment.Son but est d'entraîner un ensemble d'algorithmes sûr de grandes quantités de données afin de pouvoir classifier des données futures.
L'apprentissage automatique est généralement divisé en :

- L'apprentissage automatique supervisé :La forme la plus commune d'apprentissage automatique est l'apprentissage supervisé. Le but de cette méthode est que l'algorithme puisse «apprendre» en comparant sa sortie réelle avec les sorties «enseignées» pour trouver des erreurs et modifier le modèle en conséquence.

- L'apprentissage automatique non supervisé : L'apprentissage non supervisé, encore appelé apprentissage à partir d'observations.dans cette méthode l'algorithme d'apprentissage trouve tout seul des points communs parmi ses données d'entrée.

2.2.2.2 La différence entre l'apprentissage profond et l'apprentissage automatique

Dans un processus de l'apprentissage automatique , l'algorithme doit être informé de la façon d'effectuer une prédiction précise en utilisant plus d'informations (par exemple, en effectuant une extraction de caractéristiques pertinentes manuellement). Dans le Deep Learning, l'algorithme peut apprendre à effectuer une prédiction précise par le biais de son propre traitement de données, grâce à la structure du réseau neuronal artificiel(par exemple, il ignore les étapes manuelles et l'extraction de caractéristiques et le processus de modélisation se fait automatiques) comme montré la figure 2.1 :

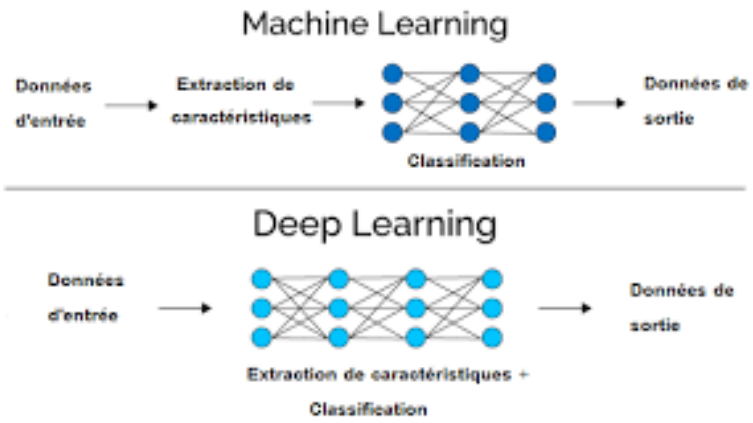

Figure 2.1: la différence entre ML et DL d'une vue architecture

Une autre différence majeure est le fait que les algorithmes de Deep Learning évoluent avec les données [31].Pour réussir une application de Deep Learning, on a besoin d'un volume de données très important -et de grande dimension- pour entraîner le modèle, en plus d'un ou de plusieurs GPU (processeur graphique) pour traiter les données rapidement. Si on n'a pas besoin de ces éléments, il est préférable d'utiliser l'apprentissage automatique plutôt que l'apprentissage profond.

Avec quatre paramètres, je vous ajuste un éléphant. Avec un cinquième, je peux même lui faire agiter la trompe.

Attribué à John von Neumann

La figure 2.2 représente la différence entre ML et DL d'une vue résultat :

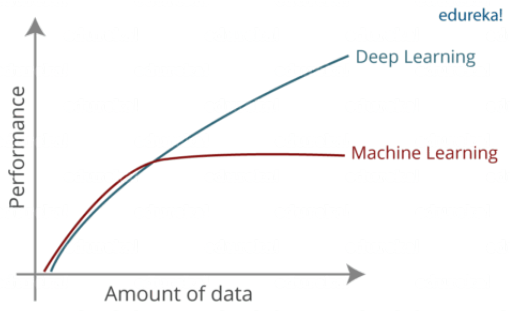

Figure 2.2: la différence entre ML et DL d'une vue résultat

2.2.3 L'importance de l'apprentissage profond

Extraction de caractéristiques est un des grands défis des modèles d'apprentissage machine traditionnels qui a été automatisée par les modèles de Deep Learning pour leur permettre d'atteindre

un taux de précision particulièrement élevé pour les tâches de vision par ordinateur [31]. La capacité de traiter un grand nombre de fonctionnalités rend l'apprentissage en profondeur très puissant lorsqu'il s'agit de données non structurées.

Cependant, les algorithmes d'apprentissage en profondeur peuvent être exagérés pour des problèmes moins complexes car ils nécessitent l'accès à une grande quantité de données pour être efficaces.

Le succès de DL appartient à la disponibilité de plus de données d'entrainement. Google, Facebook et Amazon a déjà commencé à l'utiliser pour faire l'analyse de leurs énormes quantités de données [42] [43].

La figure 3.2 représente un résumé des progrès de l'IA depuis ses débuts jusqu'à l'apparition de DL :

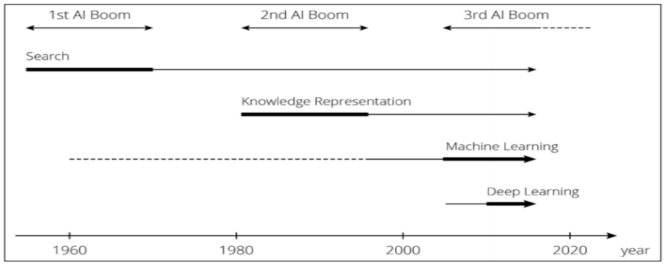

Figure 2.3: Les progrès de l'Intelligence Artificielle [44]

La relation entre les trois concepts IA, ML et DL est résumée par les auteurs dans la figure 2.4:

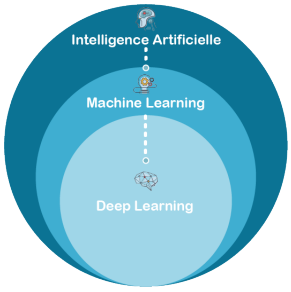

Figure 2.4: La relation entre les 3 concepts [45]

2.2.4 Réseau neuronal

La pratique, de tous les algorithmes de DL sont des réseaux neuronaux aussi appelés ANN [32]. Les ANN sont des modèles de traitement de l'information qui simulent le fonctionnement d'un système nerveux biologique. C'est similaire à la façon dont le cerveau manipule l'information au

niveau du fonctionnement. Tous les réseaux neuronaux sont constitués de neurones inter connectés qui sont organisés en couches [32].

Le détail du neurone et la description de son fonctionnement sera expliquer dans la partie prochaine.

2.2.4.1 Neurone

Ce qui forme les réseaux de neurones, ce sont les neurones artificiels inspirés du vrai neurone qui existe dans notre cerveau. La figure 2.5 montre une représentation d'un neurone réel et d'un neurone artificiel :

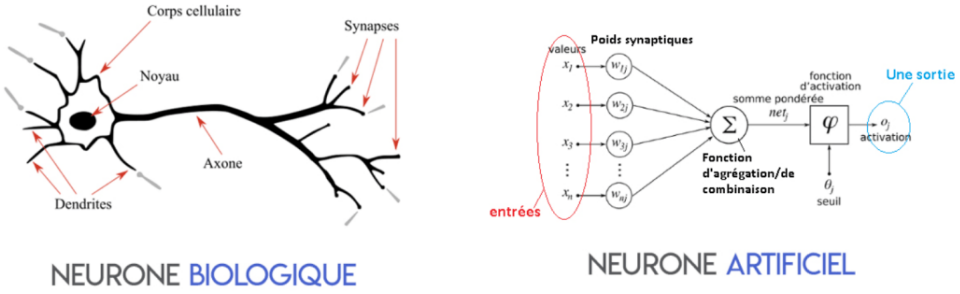

Figure 2.5: représentation d'un neurone réel et d'un neurone artificiel

2.2.4.2 Fonctions d'activation

La fonction d'activation est une composante essentielle du réseau neuronal. Ce que cette fonction a décidé est si le neurone est activé ou non. Il calcule la somme pondérée des entrées et ajoute le biais. C'est une transformation non linéaire de la valeur d'entrée.La non-linéarité est si importante dans les réseaux de neurones, sans la fonction d'activation, un réseau de neurones est devenu simplement un modèle linéaire.
Il existe de nombreux types de ces fonctions, parmi lesquelles nous trouvons [46] [47] : La fonction Sigmoïde , La fonction ReLu ,La fonction Softmax ...

- La fonction Sigmoïde : Cette fonction est l'une des plus couramment utilisées.Elle représente la fonction de répartition de la loi logistique. Elle est souvent utilisée dans les réseaux de neurones parce qu'elle est dérivable, ce qui est une contrainte pour l'algorithme de rétropropagation.La figure 2.6 représente graphiquement la fonction Sigmoïde :

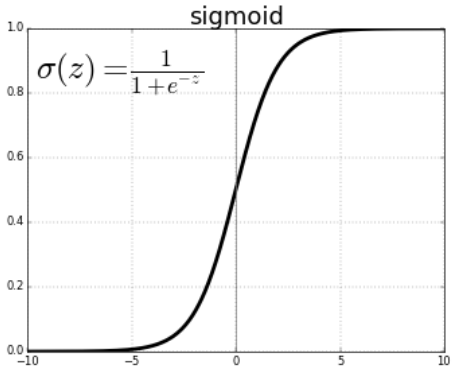

Figure 2.6: Représentation graphique de la fonction Sigmoïde [48]

2.2.5 Architectures des réseaux de neurones

Dans les années 1980, la plupart des réseaux de neurones ne formaient qu'une seule couche en raison du coût de calcul et de la disponibilité des données. De nos jours, nous pouvons nous permettre d'avoir plus de couches cachées dans nos réseaux de neurones,d'où le surnom d'apprentissage profond. Les différents types de réseaux de neurones disponibles à l'utilisation se sont également propagés,tels que les réseaux de neurones Convolutionnels (CNN), les réseaux de neurones récurrents(RNN) .. -qu'on va détailler ensuite-.

2.2.6 Recurrent Neural Networks

2.2.6.1 Introduction

Généralement , la réflexions humaines ne commence pas à partir du zéro mais en accumulant la compréhensions des choses précédentes. Les réseaux de neurones traditionnels ne peuvent pas faire cela, et cela semble être une lacune majeure.

2.2.6.2 Définition

Les réseaux de neurones récurrents sont une catégorie de réseaux de neurones classiques principalement utilisé dans la reconnaissance vocale et le traitement automatique du langage naturel.Ce sont des réseaux avec des boucles, permettant aux informations de persister.

Si la séquence que nous traitons est une phrase de 3 termes par exemple, le réseau va être déroulé en un réseau neuronal à 3 couches, une couche pour chaque mot, la figure 2.7 représente cette idée :

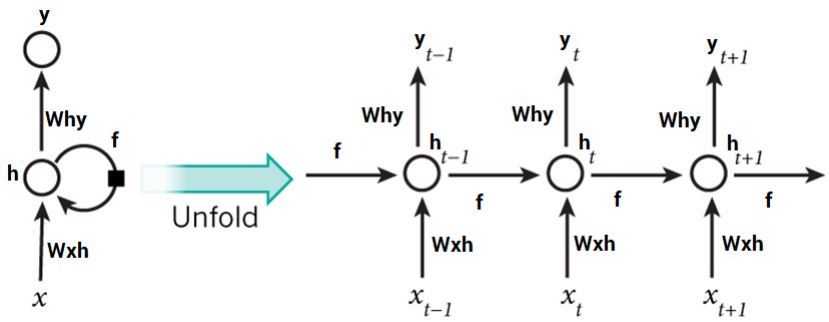

Figure 2.7: Un exemple d'un réseau récurrent qui se déroule[49]

2.2.6.3 Architecture du RNN simple

Les couches d'un RNN peuvent être divisées en couches d'entrée, couches cachées et couches de sortie. les couches d'entrée et de sortie sont caractérisées par des connexions à action directe, les couches cachées en contiennent des récurrentes. la figure 2.8 représente l'architecture du RNN simple.

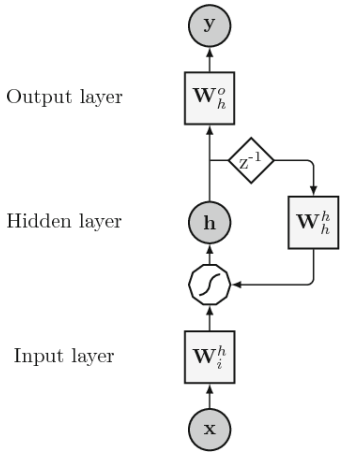

Figure 2.8: Architecture du RNN simple

- Les cercles représentent respectivement les nœuds d'entrée x, masqués h et de sortie y.
- Les carrés W_{ih} : la matrice qui représente le poids d'entrée, W_{hh} : la matrice qui représente le poids caché et W_{ho} : la matrice qui représente le poids de sortie. Leurs valeurs sont généralement ajustées dans la phase d'entraînement.
- Le polygone représente la transformation non linéaire effectuée par les neurones et z-1 est l'opérateur de retard unitaire.

2.2.6.4 Fonctionnement

Les réseaux de neurones récurrents produisent les résultats en suivant les étapes jointes :

- Fournir un seul pas de temps de l'entrée au réseau, c-à-d que x[t] est fourni au réseau (x[t] :l'entrée), elle peut contenir des valeurs réelles, des valeurs discrètes, des vecteurs one-hot .. etc.

- Calculer son état actuel en combinant l'entrée actuelle et l'état précédent, c'est-à-dire qu'on calcule h[t]. On applique cette formule :

$$h[t] = f(W_i(x[t] + b_i) + W_h(h[t1] + b_h))$$

  Où f (·) est la fonction d'activation des neurones, généralement implémentée par un sigmoïde ou par un hyperbolique tangente tanh

- L'état actuel h[t] devient l'état précédent h[t-1] pour le pas de temps suivant.

- Une fois toutes les étapes de temps terminées, l'état actuel final est utilisé pour calculer la sortie y[t] , ps :le nombre de fois de pas de temps dépend de l'exigence du problème ,il faut juste combiner les informations de tous les états précédents .
  On calcule la sortie via :

$$y[t] = g(W_{ho}(h[t] + b_o))$$

  g (·) une transformation, généralement linéaire, sur la matrice des poids de sortie W_{ho} appliquée à la somme de l'état actuel h[t] et du vecteur de biais b_o.

2.2.6.5 Limites des RNNs réguliers

Les RNNs réguliers peuvent avoir des difficultés à apprendre les dépendances à long terme parce que généralement les résultats dépendent de l'état précédent ou les n pas de temps précédents.
À la rétro-propagation de l'erreur et en appliquant la règle de chaîne :

$$\frac{\delta E}{\delta W} = \frac{\delta E}{\delta y_t} * \frac{\delta y_t}{\delta h_t} * \frac{\delta h_t}{\delta y_{t-1}} * \frac{\delta y_{t-1}}{\delta h_{t-2}} ....$$

Et si l'un des gradients s'approchait de 0, tous les gradients se précipiteraient à zéro en raison de la multiplication. Et c'est le problème du gradient qui disparaît ou qui explose [33].

2.2.6.6 Les Algorithmes d'optimisation de la descente de gradient

Il existe trois principaux types de variantes de l'algorithme de descente de gradient.La principale différence entre eux est la quantité de données que nous utilisons lorsque nous calculons le gradi-

ent pour chaque étape d'apprentissage [50][46].Parmi ces optimaseurs nous avons :

Adam : est un algorithme d'optimisation présenté en 2015 [51]. Le nom de cet algorithme est dérivé de Adaptive Moment Estimation. Lors de l'introduction de cet algorithme, les auteurs ont présenté les avantages de l'utilisation d'Adam sur des problèmes d'optimisation non convexes, comme suit :

- Simplicité de mise en œuvre.
- Efficacité du calcul.
- Peu de mémoire requise.
- Bien adapté aux problèmes volumineux en termes de données et/ou de paramètres.
- Les hyper-paramètres nécessitent généralement peu de réglages.

Il existe d'autres optimiseurs avec différents mécanismes de fonctionnement,comme:

- Adagrad
- RMSProp
- Adadelta

2.2.6.7 Architectures des RNNs

Pour surmonter le problème du gradient du fuite , d'autres architectures évoluées des RNNs ont apparait. Il existe des architectures comme le LSTM (mémoire à court terme long) et le GRU (Unités récurrentes fermées) qui peuvent être utilisés.

1. **Unités récurrentes fermées - GRU :** Un réseau réseau récurrent fermé - ou à portes- ,en anglais Gated Recurrent Unit (GRU) est une variante des LSTM introduite en 2014 par Kyunghyun Cho et al.
2. **Architecture du GRU :** Les GRU ont la forme d'une chaîne de modules répétitifs de réseau de neurones, où chaque module a quatre couches de réseau neuronal interagissant. Et c'est la différence pour les RNN classiques qui ont une seule couche.
   La figure 2.9 représente l'architecture d'une cellule GRU :

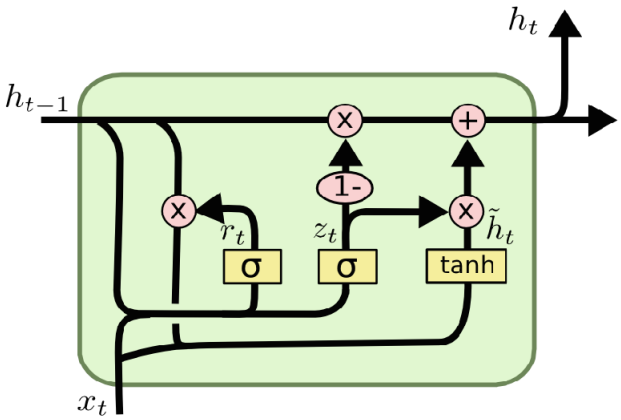

Figure 2.9: une cellule GRU

On a aussi l'état de cellule, le nouveau mécanisme adopté par les LSTM et les GRU par lequel les informations peuvent se transiter. De cette façon, ils peuvent se souvenir ou oublier des choses de façon sélective.

Ces états sont gérés par des mécanismes appelés les portes qui prennent la responsabilité de laisser éventuellement passer les informations.

Ils ont moins de paramètres que LSTM, car ils ne disposent pas d'une porte de sortie.

3. **Architecture détaillée du GRU :** GRU, (Gated Recurrent Unit), pour sa part dispose de deux portes et un état en sortie . L'état ou la mémoire de la cellule est combiné avec de nouvelles informations provenant de l'entrée et la cellule décide de ce qu'il est pertinent de conserver dans l'état de la cellule et quelles informations sont inutiles et les oublie.

    - Dans la première étape, nous avons l'opération sigmoïde appliquée à la combinaison de l'entrée actuelle et de la sortie de la cellule précédente ,comme montré dans la figure 2.10 :

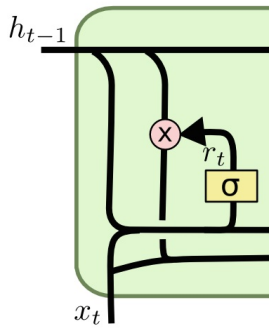

Figure 2.10: l'opération segmoïde dans un GRU

Dans l'opération sigmoïde, le réseau décide quelles informations sont pertinentes pour continuer entre la sortie précédente et la nouvelle entrée en affectant des valeurs de 0 à 1 où 0 correspond aux informations que nous voulons oublier et 1 aux informations que nous voulons conserver. Ce résultat est ensuite multiplié par la sortie précédente pour continuer pour l'étape suivante.

- Dans la deuxième étape, nous allons avoir 2 portes montrées dans la figure 2.11 :

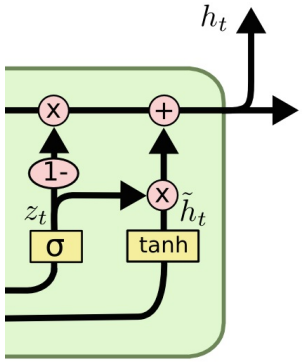

Figure 2.11: Porte de reset de mise à jour dans un GRU

Porte de reset (reset gate) : Cette porte sert à contrôler combien d'information passée le réseau doit oublier. L'état caché précédent, concaténé avec les données d'entrée, passe par une sigmoïde (pour ne conserver que les coordonnées pertinentes).
Puis est multiplié par l'ancien état caché : on n'en conserve donc que les coordonnées importantes (telles qu'elles) de l'état précédent (on a donc perdu une partie de l'état précédent dans cette porte).
La porte de reset (réinitialisation) R_t est calculée comme suit:

$$R_t = \sigma(x_t W_{xr} + h_{t-1} W_{hr} + b_r)$$

Porte de mise à jour (update gate) : Cette porte agit exactement de la même manière que les portes oubli et d'entrée du LSTM .

Elle décide des informations à conserver et de celles à oublier. Les données d'entrées et l'ancien état caché sont concaténés et passent par une fonction sigmoïde dont le rôle est de déterminer quelles sont les composantes importantes.

La porte de mise à jour Z_t est calculée comme suit:

$$Z_t = \sigma(x_t W_{xz} + h_{t-1} W_{hz} + b_z)$$

- Ensuite pour calculer la sortie du réseau GRU on commençon par intégrer la porte de réinitialisation avec un mécanisme régulier de mise à jour de l'état latent.

$$h_t = tanh(x_t W_{xh} + h_{t-1} W_{hh} + b_h)$$

- Si on veut pouvoir réduire l'influence des états précédents, nous pouvons multiplier H_{t1} avec R_t élément par élément. Chaque fois que les entrées dans R_t sont proches de 1 on récupère un RNN. Tout état caché préexistant est donc «réinitialisé» aux valeurs par défaut. Cela conduit au candidat suivant pour un nouvel état caché (c'est un candidat puisque nous devons encore incorporer l'action de la porte de mise à jour).

$$\tilde{h} = tanh(x_t W_{xh} + R_t * H_{t-1}) W_{hh} + b_h)$$

- Ensuite, nous devons incorporer l'effet de la porte de mise à jour. Cela détermine dans quelle mesure le nouvel état h_t est juste l'ancien état h_{t1} et de combien le nouvel État candidat $\tilde{H}$ est utilisé. La variable de gating Z_t peut être utilisé à cette fin, simplement en prenant des combinaisons convexes élément par élément entre les deux candidats. Cela conduit à l'équation de mise à jour finale pour le GRU.

$$h_t = Z_t * h_{t-1} + (1 - Z_t) * \tilde{h_t}$$

- Chaque fois que la porte de mise à jour est proche de 1 nous conservons simplement l'ancien état. Dans ce cas, les informations de x_t est essentiellement ignoré, sautant effectivement le pas de temps t dans la chaîne de dépendance. Chaque fois qu'il est proche de 0 le nouvel état latent h_t s'approche de l'état latent candidat $\tilde{h_t}$.

2.3 Apprentissage profond et analyse des sentiments

Le Deep Learning a prouvé son efficacité dans de nombreux problèmes complexes avec l'utilisation de réseaux de neurones artificiels pour apprendre et extraire des modèles et des informations significatives depuis les données. Par conséquent, nous trouvons de nombreuses contributions qui

tentent d'adapter cette approche comme une solution au problème de l'analyse du sentiment.

Maintenant nous allons présenter les taches nécessaires pour effectuer ce travail.

2.3.1 préparation des données

La première étape est l'étape de préparation de l'ensemble de données qui représente le processus de chargement d'un ensemble de données ,
Sachant qu'il n'est pas suffisant d'avoir beaucoup de données. Il faut aussi qu'elles soient de bonne qualité.Les problèmes viennent presque toujours de cette qualité.
Pour que les modèles soient correctement entrainés et qu'ils fournissent les résultats attendus, les données utilisées doivent être représentatives , propres, précises, complètes et bien labélisées.A titre d'exemple, si nous souhaitons prédire les sentiments des commentaires issus de réseaux sociaux, le corpus doit contenir le même type de documents. La préparation de ces données est donc une étape cruciale.

2.3.2 Pré-traitement du corpus

Le pré-traitement et le nettoyage des données sont des tâches importantes qui doivent intervenir avant d'utiliser un jeu de données pour la formation de modèles.Dans cette phase on fait appel à des techniques qui peuvent modifier la forme d'un mot ou l'éliminer complètement. à titre d'exemple la suppression des caractères non-arabes .

Ensuite tous les mots du corpus préparé doivent maintenant être convertis en quelque chose qui donne des informations sur les caractéristiques (c'est-à-dire les mots) d'une manière qui peut être utilisée pour l'apprentissage.
Actuellement, les modèles de texte de l'état de l'art sont les incorporations de mots (en anglais : word embeddings) ou les vecteurs de mots, qui sont appris à partir des données de texte.

2.3.3 Word embedding

Word embeddings ou plongement des mots sont le mappage de mots à des vecteurs de nombres réels dans un espace dimensionnel réduit [34] 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 .
Les vecteurs d'incorporation de mots représentent les mots et leurs contextes; ainsi, les mots avec des significations similaires (synonymes) ou avec des relations sémantiques étroites auront des plongements plus similaires. De plus, les incorporations de mots doivent refléter la manière dont

les mots sont liés les uns aux autres. Par exemple, les inscriptions pour «homme» devraient être «roi» comme «femme» est «reine».

Étant donné que l'apprentissage des word embeddings demande du temps et de la puissance de calcul, nous pourrions également commencer par des incorporations pré-entraînées entre autre : Bert ,Flair, ELmo .., en particulier si nous n'avons pas beaucoup de données d'entraînement.

Dans notre modèle, nous voulons apprendre les embeddings de mots en utilisant un modèle de langage Flair pré-entraîné et utiliser directement ces embeddings appris pour la classification.

2.4 Flair

2.4.1 Introduction

La compréhension du contexte a fait tomber les barrières qui avaient empêché les techniques de NLP de progresser auparavant. Et ici ,on va parler d'une de ces bibliothèques - Flair.

2.4.2 Définition

Flair est une simple bibliothèque open-source de traitement du langage naturel (NLP) développée par Zalando-Research.

En utilisant cette bibliothèque on aura Flair embeddings qui sont des incorporations de mots contextuelles qui capturent la signification des mots dans leur contexte et donc produire des plongements différents pour les mots polysémiques en fonction de leur utilisation, et modéliser les mots et le contexte fondamentalement comme des séquences de caractères.

Le framework Flair s'appuie directement sur PyTorch, l'un des meilleurs frameworks d'apprentissage en profondeur.Il est conçu pour faciliter la formation et la distribution des modéles de l'etat de l'art. L'équipe de recherche de Zalando a également publié plusieurs modèles pré-formés pour les tâches NLP suivantes:

- Reconnaissance de nom-entité (Name-Entity Recognition -NER): Il peut reconnaître si un mot représente une personne, un lieu ou des noms dans le texte.
- Balisage des parties du discours (Parts-of-Speech Tagging -PoS): balise tous les mots du texte donné en fonction de la «partie du discours» à laquelle ils appartiennent.
- Classification du texte: classification du texte en fonction des critères (étiquettes)
- Formation de modèles personnalisés: fabrication de nos propres modèles personnalisés.

2.4.3 Avantages de Flair

Beaucoup de fonctionnalités impressionnantes intégrées dans la bibliothèque Flair ont ajouté des améliorations dans l'état de l'art du NLP. Voici la sélection des plus importantes:

- Il comprend des embeddings de mots populaires et à la pointe de la technologie , tels que GloVe, BERT, ELMo, les incorporations de caractères, .. etc.
- Il est très facile à utiliser grâce à l'API Flair
- L'interface de Flair nous permet de combiner différents embeddings de mots et de les utiliser pour intégrer des documents. Cela conduit à son tour à une augmentation significative des résultats
- «Flair Embedding» sont les incorporations de mots fournies dans la bibliothèque Flair. Il est alimenté par des incorporations de chaînes contextuelles .
- Flair prend en charge un certain nombre de langues - et cherche toujours à en ajouter de nouvelles

2.4.3.1 Empilement des incorporations

Les modèles actuels combinent souvent différents types d'incorporations en concaténant chaque vecteur d'incorporation pour former les vecteurs de mots finaux. Nous expérimentons de la même manière avec différents empilements de plongements comme dite en anglais "stacking embeddings" ; par exemple, dans de nombreuses configurations, il peut être avantageux d'ajouter des incorporations de mots classiques pour ajouter une sémantique latente au niveau des mots potentiellement plus grande à nos incorporations proposées.

Dans ce cas, la représentation des derniers mots est donnée par :

$$w_i = \begin{bmatrix} w_i^{CharLM} \\ w_i^{Glove} \end{bmatrix}$$

Ici , w_i^{Glove} sont des incorporations de Glove précalculées.

2.4.4 Classification de texte à l'aide de Flair Embeddings

La classification du texte - analyse des sentiments - en utilisant Flair embeddings se fait en suivant ces étapes :

1. Importer les données dans l'environnement local de Colab
2. Installation de Flair

3. Préparation du texte pour travailler avec Flair
4. Intégration de mots avec flair
5. Vectoriser le texte
6. Partitionnement des données en données d'entraînement et données de test
7. Faire des prédictions !

2.5 Conclusion

Le DL a prouvé son efficacité dans de nombreux problèmes complexes avec l'utilisation de réseaux de neurones artificiels pour apprendre et extraire des modèles et des informations significatives depuis les données. Par conséquent, nous trouvons de nombreuses contributions qui tentent d'adapter cette approche comme une solution au problème de l'analyse du Deep Learning sentiment.

Chapter 3

Méthodologie et Conception

3.1 Introduction

Ce travail porte sur la tâche d'analyse du sentiment des commentaire ecrites dans la langue Arabe. Pour atteindre cette objectif et pour obtenir la meilleure performance possible, nous avons suivi la méthodologie présentée au dessous ,présentant la conception de notre système en commençant par sa conception générale puis sa conception détaillée.

3.2 Conception globale du système

L'architecture de notre système de classification des sentiments peut être présenter par la figure 3.1 :

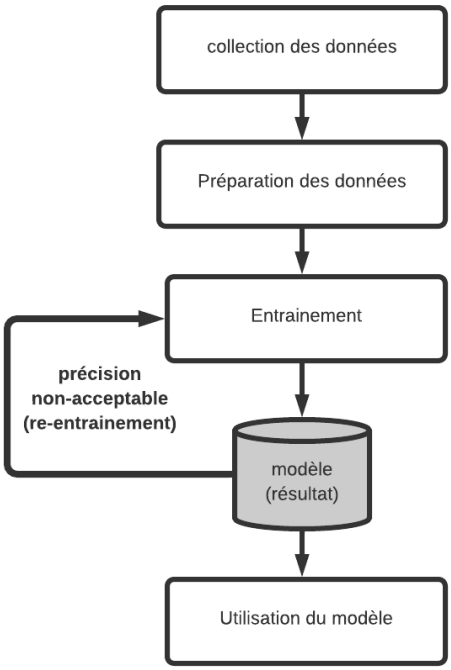

Figure 3.1: Conception globale du système

3.3 Conception détaillée

3.3.1 Collection des données

Pour profiter de l'existant et réutiliser ce que à été rassembler, nous avons exploité le Dataset utilisé dans le travail de ABDELI adel(2018) réalisé au niveau de l'université de Biskra. Le dataset est une collection de commentaires exprimés en dialecte algérien et en Arabe Standard. La collection était réalisé par un Crawler. Les données sont sauvegardées dans un fichier CSV pour le donner comme entrée à l'algorithme d'entraînement.

Cet ensemble de données compris 49864 commentaires.

3.3.2 Préparation de données

Dans ce module , on prétraite notre dataset par la suppression de tous les caractères non Arabe. Également, on marque les données avec un label positif ou négatif ensuite on les sauvegarde dans un fichier Excel en format CSV. Et finalement on divise le corpus en 2 datasets, la première pour l 'entraînement et la deuxième pour le test (validation).

3.3.3 Entraînement

3.3.3.1 Utilisation de modèle de langage Flair

pour entraîner notre modèle de classification on a besoin de transformer les données collectées en vecteurs de nombres réels . Donc après les deux précédentes phases (collection et pré-traitement des données) on passe vers l'entraînement des incorporations des mots. Nous avons utilisé le modèle de langage pré-entraîné Flair ar-forward (basé sur le contexte qui suivent le mot) et ar-backward (basé sur le contexte qui précéde le mot). En le combinant avec des incorporations de mots classiques de Glove précalculées pour ajouter une sémantique latente au niveau des mots. A titre d'exemple dans notre système, nous utilisons 3 incorporations empilées comme montre la capture d'écran dans la figure 3.2 :

Figure 3.2: Capture d'écran des modèles de langage utilisés

Les sorties d'empilement des embeddings (modèle de langage Flair et Glove) seront sous forme de vecteur-pytorch (pytorch-tensors) qui vont par la suite transformer en numpy-array pour qu'elles soient l'entrée de réseau reccurent GRU .
Nous avons choisi GRU comme architecture du réseau parce que ils s'entraînent plus rapidement et fonctionnent mieux que les LSTM avec moins de données d'entraînement si nous faisons de la modélisation de langage ,ils sont plus simples et donc plus faciles à modifier, par exemple en ajoutant de nouvelles portes en cas d'entrée supplémentaire sur le réseau.les GRU n'a pas besoin d'unités de mémoire, donc plus rapide à entraîner que LSTM et à donner selon les performances.

3.3.3.2 Architecture de modèle de classification

Pour entraîner le modèle on est besoin d'abord de transformer notre données à l'aide des fichiers Numpy en matrice des index avec une dimension de [taille des données * longueur maximale de la séquence], ou chaque ligne de matrice contient des index de chaque mot dans les vecteurs des mots, car chaque mot a un vecteur de 4196 dimensions, et chaque mot sera traité dans une cellule GRU, où le nombre des mots est limité par la variable de longueur maximale. Et après la fin d'entraînement on va sauvegarder le modèle si sa précision est acceptable, pour l'utiliser apr‘es, sinon on re-faire l'entraînement avec d'autres hyperparamètres. Pour entraîner le modèle on est besoin aussi d'une fonction qui nous retourne un lot d'échantillons (de commentaires) - en anglais:sample- avec un nombre d'échantillons, car on ne peut pas transmettre tout le dataset dans

un réseau de neurones en m^eme temps, et ce lot sera transmis avec ces étiquettes de chaque item : positif ou négatif .

la figue 3.3 récapitule l'architecture de notre système:

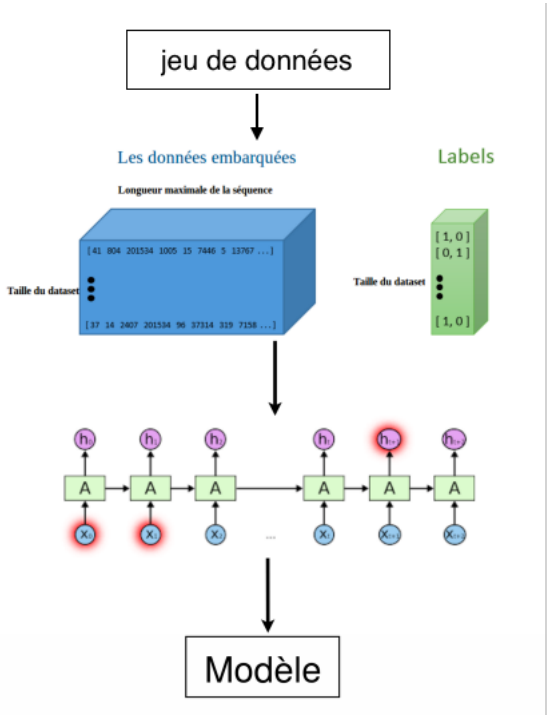

Figure 3.3: Architecture de modèle de classification

3.3.3.3 Architecture du réseau

Pour réaliser notre système, nous avons appliqué une méthode supervisée de l'apprentissage profond, qui est le réseau de neurones récurrent, (en anglais, Recurrent Neural Network -RNN), et nous avons choisi exactement la mthétode de réseau récurrent à porte, en anglais, Gated recurrent unit(GRU). La figure 3.4 représente les différentes couches que contient notre modèle :

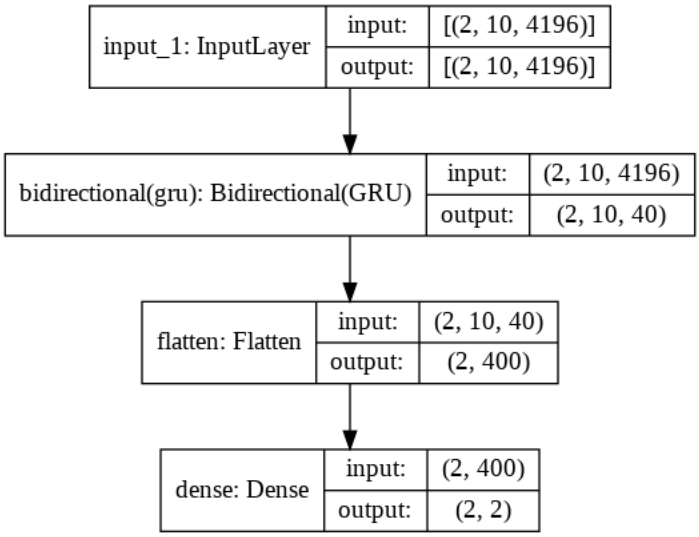

Figure 3.4: L'architecture du modèle GRU

- La couche des entrées (InputLayer) comprend la taille du lot (batchsize) , longueur max de la séquence (maxlengh) , la taille des embeddings (embeddingsize).
- La taille du lot qui est égal à 2 ,est un hyperparamètre qui définit le nombre d'échantillons à traiter avant de mettre à jour les paramètres du modèle interne. longueur maximum de la séquence qui est 10 . la taille des des embeddings qui est égal à 4196 .
- Considérant un lot (batch) comme une boucle for itérant sur un ou plusieurs échantillons et faisant des prédictions. A la fin du lot, les prédictions sont comparées aux variables de sortie attendues et une erreur est calculée. A partir de cette erreur, l'algorithme de mise à jour est utilisé pour améliorer le modèle, par exemple descendre le long du gradient d'erreur.
- La couche bidirectionnelle -la couche GRU- a pour entrées les sorties de la couche précédentes et comme sortie la taille du lot ,la longueur maximale ,et la taille de chaque couche GRU qui est égal à 20 .
- La couche aplatir (flatten layer) Une opération d'aplatissement sur un tenseur redéfinit la forme du tenseur pour qu'elle soit égale au nombre d'éléments contenus dans le tenseur sans la taille du lot .
- Une couche dense qui est une couche de neurones classique, complètement connectée avec la couche précédente et la couche suivante. Une couche dense en sortie avec un neurone par classe, et une fonction d'activation "sigmoïqui donde"ne une probabilité (dont la somme vaut 1) en sortie de chaque neurone, le neurone de sortie avec la probabilité la plus grande permettant alors de décider que sa classe associée est la classe prédite.

3.3.3.4 Test du modèle

Pour le tester, on calcule la précision sur un corpus de test jamais vue par le modèle, et si la précision est élevée (plus de 70%) on occupe le modèle sinon, on refait le traitement avec d'autres hyperparamètres.

3.3.3.5 Utilisation du modèle

On prend un texte(commentaire, tweet, critique...etc), on le prétraite et le donne à notre modèle de catégorisation des sentiment sauvegardé comme une entrée, et notre modèle va catégoriser notre texte soit en sentiment négatif ou positif.

3.4 Conclusion

Dans ce chapitre ,nous avons présenté notre méthode proposée pour arriver à notre modèle . on a bien détaillé les modules utilisés et les trois phases essentielles(Collection des données, Préparation des données et Entraînement). Et dans le chapitre suivant, nous allons décriver l'implémentation de notre système.

Chapter 4

Evaluation

4.1 Inroduction

Dans ce chapitre, nous allons présenter l'environnement de travail, le langage de programmation, et les outils que nous avons utilisés pour construire le système. Par la suite nous allons expliquer toutes les expérimentations que nous avons appliquées sur la méthode proposée et les résultats obtenus.

4.2 Environnement et outils de développement

Dans ce partie, nous allons présenter l'environnement de travail, le langage de programmation, et les outils que nous avons utilisés pour construire le système. Par la suite nous allons expliquer toutes les expérimentations que nous avons appliquées sur la méthode proposée et les résultats obtenus.

Pour développer notre système et valider notre proposition, nous avons utilisé le langage de programmation Python et l'environnement Google Colab pour écrire les programmes.Pour l'apprentissage profond nous avons utilisé la bibliothèque de Google qui s'appelle TensorFlow. Et aussi la bibliothèque Numpy pour manipuler les matrices.

4.2.1 Environnement de développement

4.2.1.1 Python

Figure 4.1: Python logo

Python est un langage de programmation de haut niveau Crée par Guido van Rossum et sorti en 1991.Python dispose d'un système de type portable, dynamique, extensible, gratuit, syntaxe très simple, code plus court que C ou Java, multi thread, orienté objet, évolutif ... [35].

Également, nous avons utilisé le package CSV : Une bibliothèque grâce à lui, nous pouvons manipuler les fichiers de format csv.

4.2.1.2 Google Colab

Figure 4.2: Google colab logo

Google Colaboratory[40], souvent raccourci en "Colab" est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur.cet environnement nous permet d'écrire et d'exécuter du code Python dans votre navigateur, avec aucune configuration requise ,accès gratuit aux GPU ,partage facile.

4.2.1.3 Outils de développement

4.2.1.4 Tensorflow

Figure 4.3: Tensorflow logo

TensorFlow est une bibliothèque logicielle open source pour le calcul numérique haute performance. Son architecture flexible permet un déploiement facile du calcul sur une variété de plates-formes (CPU, GPU, TPU), et des ordinateurs de bureau aux clusters de serveurs aux périphériques mobiles et périphériques. Développé à l'origine par des chercheurs et des ingénieurs de l'équipe Google Brain au sein de l'organisation AI de Google, il bénéficie d'un fort soutien pour l'apprentissage automatique et l'apprentissage en profondeur et le calcul numérique flexible est utilisé dans de nombreux autres domaines scientifiques.
TensorFlow a été développé pour une utilisation interne de Google. Et après il a été publié sous licence open source Apache 2.0 le 9 novembre 2015 [36].

4.2.1.5 Numpy

Figure 4.4: Numpy logo

NumPy est une bibliothèque pour le langage de programmation Python, ajoutant un support pour les matrices et matrices multidimensionnelles de grande taille, ainsi qu'une grande collection de fonctions mathématiques de haut niveau pour fonctionner sur ces matrices.L'ancêtre de NumPy, Numeric, a été créé à l'origine par Jim Hugunin avec des contributions de plusieurs autres développeurs.
En 2005, Travis Oliphant a créé NumPy en incorporant les fonctionnalités de Numarray en Numeric, avec de nombreuses modifications. NumPy est un logiciel open-source et compte de nombreux contributeurs [37].

4.2.1.6 Keras

Figure 4.5: Keras logo

Keras [38] est un framework open source d'apprentissage profond pour le Python, capable de s'exécuter sur TensorFlow. Il est écrit par Francis Chollet, membre de l'équipe Google. Keras est utilisé dans un grand nombre de startups, de laboratoires de recherche [39] (dont le CERN, Microsoft Research et la NASA) et de grandes entreprises telles que Netflix, Yelp, Square, Uber, Google, etc. Il a été développé dans le cadre de l'effort de recherche du projet ONEIROS (Open-ended Neuro Electronic Intelligent Robot Operating System). En 2017, l'équipe TensorFlow de Google a pris la décision de fournir un support pour Keras et de l'intégrer dans la bibliothèque principale de TensorFlow. Il présente un ensemble d'abstractions de plus haut niveau et plus intuitives qui facilitent la configuration des réseaux neuronaux [38].

4.3 Système de catégorisation des sentiments

4.4 Expérimentation

Pour arriver à notre système, nous avons passer par les expérimentations qui suivent :

4.4.1 Première expérimentaion

Dans la deuxième expérimentaion nous avons entraîné notre modèle de classification binaire des sentiments sur une dataset de 20742 items, 10371 positifs et 10371 négatifs,comme montré sur la figure 4.6:

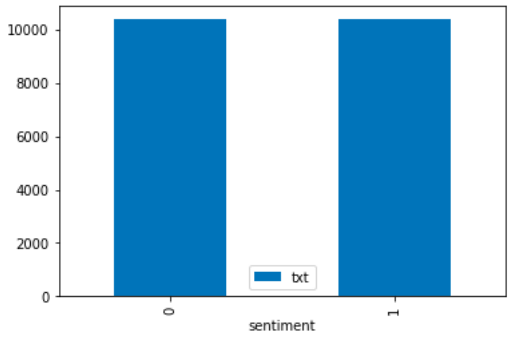

Figure 4.6: Le nombre de données dans chacune des classes positive / négative dans le 1 èr dataset

Nous avons divisé le corpus en 4149 commentaires de test(qui représente 20% de dataset) pour tester notre modèle et avoir sa précision, et 16593 commentaires d'entraînement, et nous avons obtenue comme précision 75% pour l'entraînement et 70% pour le test . la figure 4.7 montre les résultats plottés dans des graphes :

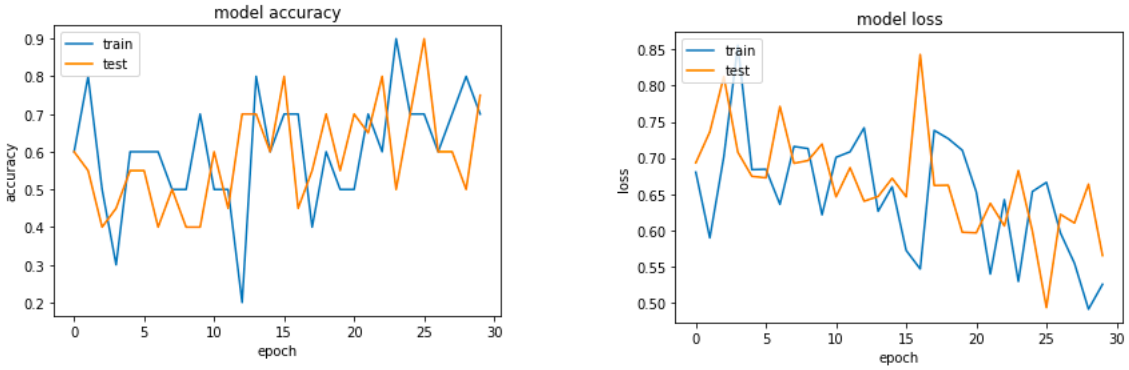

Figure 4.7: la précision et l'erreur du modèle sur le 1 èr jeu de données

4.4.2 Deuxième expérimentaion

Dans la deuxième expérimentaion nous avons entraîné notre modèle de classification binaire des sentiments sur une dataset de 66666 items, 33333 positifs et 33333 négatifs,comme montré sur la figure 4.8:

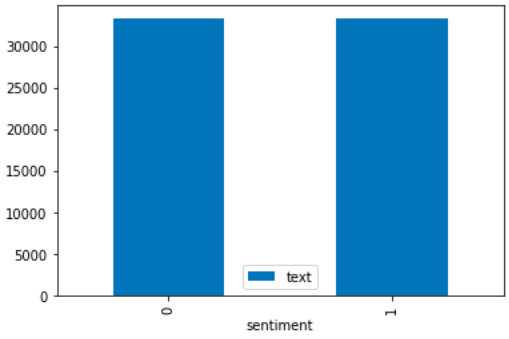

Figure 4.8: Le nombre de données dans chacune des classes positive / négative dans le 2 ème dataset

Nous avons divisé le corpus en 6667 commentaires de test(qui représente 20% de dataset) pour tester notre modèle et avoir sa précision, et 59999 commentaires d'entraînement, et nous avons obtenue comme précision 62% pour l'entraînement et 85% pour le test . la figure 4.9 montre les résultats plottés dans des graphes :

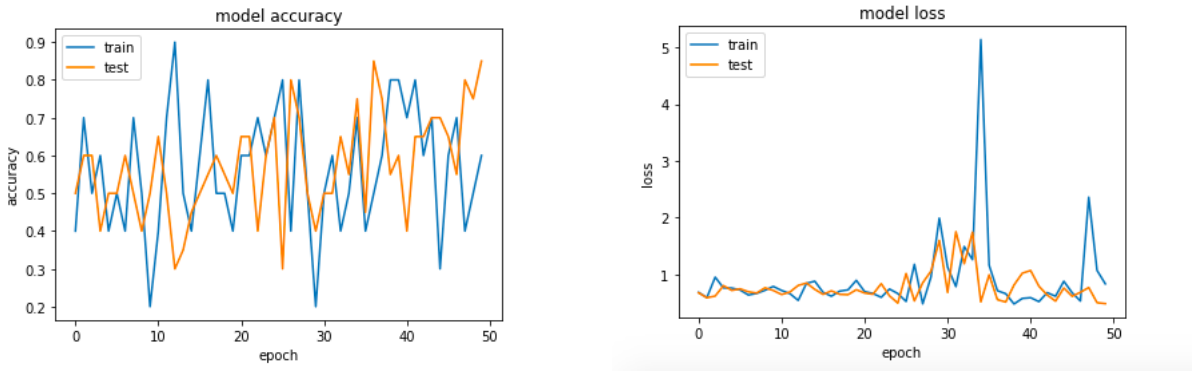

Figure 4.9: la précision et l'erreur du modèle sur le 2 ème jeu de données

4.4.3 Troisième expérimentaion

Dans la troisième expérimentaion nous avons entraîné notre modèle de classification binaire des sentiments sur un dataset de 49870 items, plus de 24900 positifs et 24900 négatifs, comme la figure 4.10 montre :

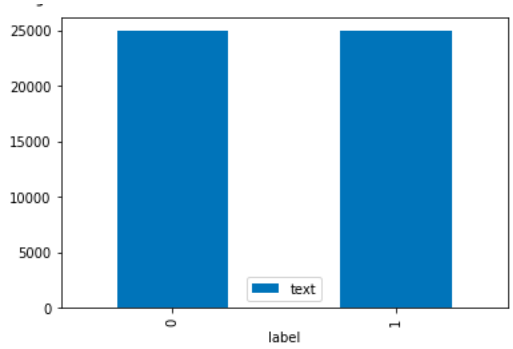

Figure 4.10: Le nombre de données dans chacune des classes positive / négative du 3 ème dataset

Et nous avons divisé le corpus en 9973 commentaires de test(qui représente 20% de dataset) pour tester notre modèle et avoir sa précision, et 39890 commentaires d'entraînement, et nous avons obtenue comme précision 90% pour l'entraînement et 70% pour le test . la figure 4.11 montre les résultats plottés dans des graphes

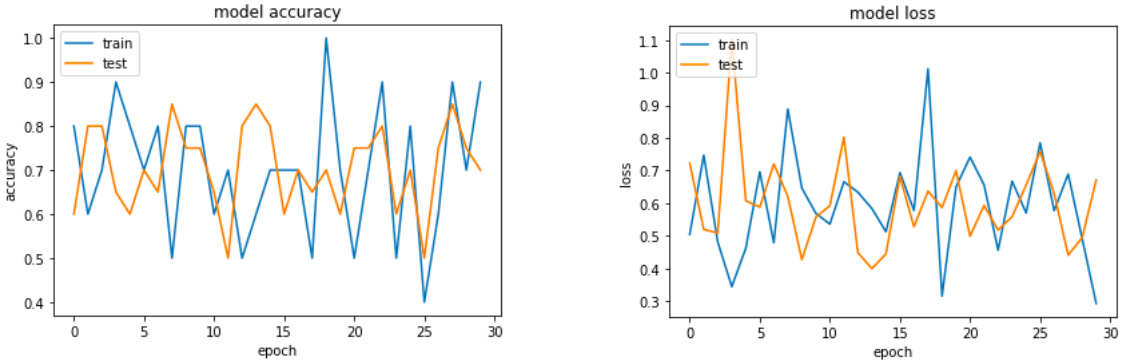

Figure 4.11: La précision et l'erreur du modèle sur le 3ème jeu de données

4.5 Résultas

Notre modèle a été appliqué sur les trois ensembles de données mentionnés précédemment et l'évaluation s'est faite en terme de la métriques précision "accuracy", comme résumé dans le tableau suivant :

dataset	données d'entrainement	données de test	précision
20742	16593	4149	75%
66666	59999	6667	62%
49870	39890	9973	90%

4.5.1 Les résultats obtenus pour le modèle

Notre décision de choisir différents ensembles de données en fonction de leur taille a été de voir comment un modèle de DL fonctionne dans deux situations où nous avons un grand et un petit volume de données. Pour les tests finaux sur les ensembles de données, nous avons entrainé le modèle à 30 époques pour le .A partir de ce que nous avons comme résultats, l'efficacité de notre proposition et la possibilité de son utilisation dans ce domaine. En effet, notre modèle de classification des sentiments, a permet de catégoriser correctement plus de 81de test de totalité de 7480 items, ¸ca veut dire que notre syst'eme a bien catégoris´e plus de 5984 items. Ainsi que, les expérimentations prouvent qu'avec l'augmentation de la taille des donn´ees d'entra^ınement le taux de pr´ecision augmente, aussi le pourcentage de dataset de test est important, comme dans les premi'eres exp´erimentations nous avons utilis´e un pourcentage faible 10% la précision aussi été faible, modèles donnent de bonnes performances si nous les utilisons pour le problème de l'analyse du sentiment où la quantité des données de l'apprentissage joue un rôle important. notre système a réussi à accrocher une précision qui est de 90%.

4.6 Conclusion

Dans ce chapitre, nous avons vu notre part de contribution au problème de l'analyse du sentiment, représentant les outils et les jeux de données utilise, ainsi que les étapes que nous avons suivies pour obtenir les résultats que nous montrons également pour notre modèle.

Conclusion

Dans ce travail, nous avons exploré le domaine de l'analyse des sentiments qui, comme tous les autres domaines du traitement du langage naturel, a connu une évolution majeure depuis les années 2000 et a réalisé une évolution majeure et un grand intérêt depuis la naissance de l'apprentissage profond. Afin d'atteindre ces résultats, nous avons passé beaucoup de temps à lire et réviser des publications et des articles pour voir et comprendre les concepts et comment appliquer un modèle de deep learning à notre problème.

Pour effectuer l'analyse des sentiments sur un texte Arabe, nous avons proposé une méthode basée sur l'apprentissage profond et plus particulièrement sur les réseaux de neurones récurrents, où on a utilisé l'architecture de réseau récurrent à port (en anglais : Gated Reccurent Unit - GRU) qui est l'une des architectures de réseaux de neurones récurrents. Dans cette méthode, on a utilisé un dataset de presque 50000 éléments est générée, où nous avons entrainé notre modèle de classification des sentiments sur cette dataset et on a atteint une précision de 90%, qui est un résultat excellent dans ce domaine, et qui nous encourage à améiorer nos recherches de plus en plus.

Enfin, avant de passer aux perspectives, ce travail nous a permis de mettre en pratique notre connaissances des réseaux de neurones et de les enrichir, et le plus important est que nous fassions le premier pas vers l'apprentissage profond, un des champs les plus importants de l'intelligence artificielle.

Pour les perspectives et les travaux de futur qui peuvent améliorer notre système de classification des sentiments , nous avons des idéés telles que :

- Tester notre modèle sur d'autres ensembles de données
- Détection et élimination des commentaires de spam car ils contiennent des textes objectifs avec aucune opinion.
- Développer le modèle pour être plus précis dans la détection de la négation.
- Ajouter une autre étiquette "neutre" pour les commentaires neutre.

- Marquer les commentaires avec certaine dégrée de polarité comme(+1, +2, ..., +5 / -1, -2, ..., -5). essayant aussi de proposer notre propre modèle de langage pour entraîner des Flair embeddings.

Référence

1. Waters, John K. The Everything Guide to Social Media: All you need to know about participating in today's most popular online communities. Adams Media, 2010.
2. "Sentiment analysis", Wikipedia, the free encyclopedia. Last Accessed 5 May. 2013 <http://en.wikipedia.org/wiki/Ser
3. Chiang, David et al. "Parsing Arabic dialects." Proceedings of the European Chapter of ACL (EACL) 2006: 112. Versteegh, K., Versteegh, C.: The Arabic Language. Columbia University Press (1997)
4. Internet world stats, http://www.internetworldstats.com/stats7.html , accédé : 26/04/2020 .
5. Versteegh, K., Versteegh, C.: The Arabic Language. Columbia University Press (1997)
6. Habash, N.: Introduction to Arabic natural language processing. Synthesis Lectures on Human Language Technologies 3(1), 1–187 (2010)
7. Habash, N., Rambow, O., Roth, R.: Mada+ tokan: A toolkit for arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In: Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR), pp. 242–245 (2009)
8. Farra, N., Challita, E., Assi, R., Hajj, H.: Sentence-Level and Document-Level Sentiment Mining for Arabic Texts. In: Proceedings of International Conference on Data Mining Workshops (ICDMW), pp. 1114–1119. IEEE (2010)
9. Brooke, J., Tofiloski, M., Taboada, M.: Cross-linguistic sentiment analysis: From English to Spanish. In: Proceedings of the 7th International Conference on Recent Advances in Natural Language Processing, Borovets, Bulgaria, pp. 50–54 (2009)
10. Denecke, K.: Using SentiWordNet for multilingual sentiment analysis. In: IEEE 24th International Conference on Data Engineering Workshop, ICDEW, pp. 507–512. IEEE (2008).
11. Elarnaoty, M., AbdelRahman, S., Fahmy, A.: A Machine Learning Approach For Opinion Holder Extraction Arabic Language. CoRR, abs/1206.1011 (2012).

12. Abbasi, A., Chen, H., Salem, A.: Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. ACM Transactions on Information Systems.

13. Liu, B.: Sentiment analysis and subjectivity. In: Handbook of Natural Language Processing, pp. 627–666 (2010) (TOIS) 26(3), 12 (2008)

14. Definition of Sentiment Analysis by Oxford Dictionary, https ://en.oxforddictionaries.com/definition/sentiment, accédé: 25/04/2020 .

15. Sentiment analysis : mining opinions sentiments, and emotions. Bing Liu. First edition USA 2015.

16. Study of Different Levels for Sentiment Analysis. Seema Kolkur, Gayatri Dantal and Reena Mahe. International Journal of Current Engineering and Technology. 2015.

17. B. Pang, L. Lee, 2008, « Opinion Mining and Sentiment Analysis », Foundations and Trends in Information Retrieval, vol. 2, no 1-2, p. 1-135, [en ligne] [URL : http://www.cs.cornell.edu/home/llee/omsa/omsa-published.pdf]. DOI : 10.1561/1500000001.

18. Exploiting subjectivity classification to improve information extraction. E. Riloff, J. Wiebe, and W. Phillips. in Proceedings of AAAI. 2005.

19. Qualitative dimensions in question answering : Extending the definitional QA task. L.V. Lita, A.H. Schlaikjer, W. Hong, and E. Nyberg. 2005.

20. Baynes, C. Trump says he would not be President without Twitter. Retrieved from https ://www.independent.co.uk/news/world/americas/us-politics/donald-trump-tweetstwitter-social-mediafacebook-instagram-fox-business-network-would-not-bea8013491.html . 2019, Mars 27.

21. Spyros E. Polykalas, George N. Prezerakos et Agisilaos Konidaris, « A General Purpose Model for Future Prediction Based on Web Search Data: Predicting Greek and Spanish Election », 27th International Conference on Advanced Information Networking and Applications Workshops, 2013

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

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

24. Josh Patterson Adam Gibson ,2017. Deep Learning A Practitioner's Approach,1ère (Ed), O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472 ,Mike Loukides Tim McGovern, 532 p, (pp. 28)

25. Yann LeCun, Yoshua Bengio Geoffrey Hinton Nature 521, 436–444 (28 May 2015)

26. M. Elhawary and M. Elfeky. Mining Arabic Business Reviews. In Proceedings of International Conference on Data Mining Workshops (ICDMW), pages 1108–1113. IEEE, 2010.

27. A. El-Halees. Arabic Opinion Mining Using Combined Classification Approach. In Proceedings of the International Arab Conference on Information Technology (ACIT), 2011.

28. M. Abdul-Mageed and M. Diab. Toward building a large-scale Arabic sentiment lexicon. In Proceedings of the 6th International Global WordNet Conference, Matsue, Japan, 2012.

29. Nawaf A. Abdulla, Nizar A. Ahmed, Mohammed A. Shehab and Mahmoud Al-Ayyoub: "Arabic Sentiment Analysis: Lexicon-based and Corpus-based", 2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), 2013.

30. Alhazmi, S., Black, W. and McNaught, J, Arabic SentiWordNet in Relation to SentiWordNet 3.0, International Journal of Computational Linguistics. 2013 et 4(1):1-11.

31. Saggie , https://www.saagie.com/fr/blog/qu-est-ce-que-le-deep-learnig/, accédé : 10/05/2020 .

32. Python Deep Learning. Ivan Vasilev, Daniel Slater, Gianmario Spacagna, Peter Roelants, Valentino Zocca. 2019

33. S. El Hihi, Y. Bengio, Hierarchical Recurrent Neural Networks for Long-term Dependencies, in: Proceedings of the 8th International Conference on Neural Information Processing Systems, NIPS'95, MIT Press, Cambridge, MA, USA, 493–499, DOI: http://dl.acm.org/citation.cfm?id=2998828.2998898, 1995.

34. Learning Semantic Relations from Text, Preslav Nakov,Diarmuid Ó Séaghdha, Vivi Nastase Stan Szpakowicz, Empirical Methods in Natural Language Processing (EMNLP), 2015.

35. Presentation du language Python, http://www.linux-center.org/articles/ 9812/python.html, accédé : le 15/06/2020 .

36. Site web de tensorflow. www.tensorflow.org , accédé : 23/06/2020 .

37. Travis E Oliphant. Python for scientific computing. Computing in Science Engineering, 9(3), 2007.

38. Keras: the Python deep learning API , http ://KERAS.io , accédé : 03/07/2020 .

39. LinkedIn , https ://www.linkedin.com/in/fchollet , accédé : 10/07/2020 .

40. Google Colab https ://colab.research.google.com/ ,accédé : 06/09/2020 .

41. Find my Facebook ID , https ://findmyfbid.com , accédé : 20/07/2020 .

42. Google AI , https ://ai.google, accédé : 20/07/2020.

43. Tools for Advancing the World's AI , https ://ai.facebook.com/tools, accédé : 22/07/2020 .

44. Java Deep Learning Essentials. Yusuke Sugomori. 2016

45. Deep Learning with Keras. Antonio Gulli, Sujit Pal. 2017

46. Deep Learning. Ian Goodfellow, Yoshua Bengio, Aaron Courville. MIT Press. 2016

47. Fundamentals of Deep Learning Designing Next-Generation Machine Intelligence Algorithms. Nikhil Buduma. 2017

48. Fundamentals of Deep Learning Designing Next-Generation Machine Intelligence Algorithms. Nikhil Buduma. 2017

49. Twitter US Airline Sentiment , https ://www.kaggle.com/crowdflower/twitter-airline-sentiment ; accédé : 01/08/2020

50. Python Deep Learning. Ivan Vasilev, Daniel Slater, Gianmario Spacagna, Peter Roelants, Valentino Zocca. 2019

51. ADAM : A METHOD FOR STOCHASTIC OPTIMIZATION. Diederik P. Kingma, Jimmy Lei Ba. 2017