REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : IA3 /M2/2018

Mémoire

présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Intelligence artificielle**

Deep learning-based sentiment analysis of Algerian-Arabic short texts

---

Par :

ABDELLI ADEL

Soutenu le 24/06/2018, devant le jury composé de :

Kalfali Toufik	MAA	Président
Guerrouf Fayçal	MAA	Rapporteur
Aloui Ahmed	MAA	Examinateur

Dédicace

Je dédie ce modeste travail comme un témoigne d'affectation, de respect et d'admiration à tous ceux qui me sont chers, à mon cher père et mère, à mes frères et soeurs, à mes meilleurs amis et à toutes personnes ayant contribué de près ou de loin à la réalisation de ce travail.

Remerciement

Je tiens premièrement à remercier Allah le tout puissant de m'avoir donné le courage et la patience pour terminer ce travail.

Je veux adresser tous mes remerciements aux personnes avec lesquelles j'ai pu échanger et qui m'ont aidé pour la rédaction de ce mémoire. En commençant par remercier tout d'abord mon encadreur "Guerrouf Fayçal", pour le suivi de mon travail, ses conseils, ses encouragements, sa patience, son aide précieuse et pour le temps qu'il m'a consacré.

Mes grands remerciements aussi aux enseignants de département d'informatique ainsi que les membres de jury qui ont pris la peine d'évaluer mon travail.

Enfin, j'adresse mes plus sincères remerciements à ma famille et mes amis, qui m'ont accompagné, aidé, soutenu et encouragé tout au long de la réalisation de ce mémoire.

Résumé

Après l'apparition du web 2.0, et prolifération rapide des services de réseaux sociaux tels que Facebook, Twitter et Youtube, l'analyse des sentiments a pris une place énorme dans le domaine du traitement du langage naturel (TLN), où de nos jours la recherche sur l'analyse des sentiments est appliquée dans diverses applications telles que le marketing et la politique. De nombreux chercheurs ont travaillés sur l'analyse des sentiments Arabes et les dialectes Arabes, mais peu d'entre eux ont concentrés sur l'Arabe Algérien.

Dans ce travail on a proposé une approche d'apprentissage profond pour l'avalyse des sentiments, précisément les réseaux de neurones récurrents, qui aborde à la fois l'Arabe standard moderne (MSA) et l'Arabe Algérien vernaculaire utilisé dans les réseaux sociaux, où on propose notre jeu de données (dataset) annoté manuellement, et aussi notre modèle de Word2Vec, et le corpus utilisé pour entrainer le Word2Vec.

Mots clés : Analyse des sentiments, Apprentissage profond, Word2vec, Apprentissage automatique, Traitement automatique de langage.

ملخص

مع ظهور الويب 2.0 ، والإنتشار الواسع لخدمات التواصل الإجتماعي مثل الفايسبوك، التويتر و اليوتوب، تحليل المشاعر أخذ مكان كبير في ميدان معالجة اللغات الطبيعية، ففي وقتنا الحالي تحليل المشاعر يستعمل في كثير من الميادين مثل التسويق والسياسة، هناك الكثير من الباحثين عملوا على تحليل المشاعر من النص المكتوب باللغة العربية الفصحى ومختلف لهجاتها، ولكن هناك القليل من الدراسات التي ركزت على اللهجة الجزائرية.

في هذا العمل، نقترح التعلم الذاتي العميق للآلة كمقاربة لتحليل المشاعر، ونقترح بالتحديد الشبكة العصبية المتكررة لتحليل كل من اللغة العربية الفصحى واللغة العامية الجزائرية المستعملة في مواقع التواصل الإجتماعي. أين نقترح مجموعة من البيانات المعلمة يدويا، ونقترح أيضا نموذجنا لتضمين الكلمات، والبيانات المستخدمة في إنشائه.

الكلمات المفتاحية: تحليل المشاعر، التعلم العميق للآلة ،تضمين الكلمات، تعلم الآلة، معالجة اللغات الطبيعية.

Abstract

After the appearance of web 2.0, and rapid proliferation of social networking services such as Facebook, Twitter and YouTube, sentiment analysis took a huge place in the natural language processing (NLP) field, where nowadays sentiment analysis research is applied in a variety of applications such as marketing and politics. There are many researchers worked on Arabic sentiment analysis and Arabic dialects, but few of these studies have focused on the Algerian Arabic.

In this work, we propose a deep learning approach for sentiment analysis, precisely the recurrent neural networks that addresses both modern standard Arabic (MSA) and the vernacular Algerian Arabic utilized in social networks. Where we propose our manually annotated dataset, and also our Word2Vec model, and the corpus used to train Word2Vec.

Keywords : Sentiment analysis, Deep learning, Word2vec, Machine Learning, Natural language processing.

Table des matières

Table des figures

Liste des tableaux

Introduction générale

Les dernières années sont principalement caractérisées par la prolifération rapide des services de réseaux sociaux tels que Facebook, Twitter et YouTube. Ces réseaux sociaux ont permis aux individus et aux groupes d'exprimer et de partager leurs opinions sur différents sujets (produits, événements politiques, économie, restaurants, livres, hôtels, clips vidéo, etc.).

Des milliards de commentaires et de critiques sont ajoutés chaque jour sur le web, ce qui a conduit à la nécessité d'exploiter les opinions des utilisateurs afin de découvrir des informations utiles. Exploitation de cet énorme volume de commentaires et de critiques est presque impossible manuellement.Par conséquent, une nouvelle thématique du traitement du langage naturel (TLN), connue sous le nom d'analyse du sentiment ou analyse d'opinions (Opinion Mining), a émergé. Le but principal de l'analyse des sentiments est d'extraire les sentiments/opinions des utilisateurs à partir des contenus créés en utilisant des techniques d'extraction automatique pour déterminer leurs attitudes par rapport à un sujet, souvent exprimé sous forme textuelle.

De nos jours, l'analyse des sentiments est principalement utilisée par les entreprises pour découvrir les opinions de différents clients dans le cadre d'un marketing. Il est également utilisé en politique pour prédire les résultats des élections ou pour connaître les opinions publiques sur les différentes politiques. Le champ de l'analyse de sentiment est considéré comme une tâche de classification pour décider si un avis est positif ou négatif.

La plupart des recherches existantes sur l'analyse des sentiments se concentrent sur le texte Anglais. En dépit de son importance en tant que l'une des langues les plus utilisées dans le monde, seules un nombre limité de recherches sur l'analyse du sentiment du texte Arabe ont été réalisées. Les approches de l'analyse du sentiment Arabe proposées se concentrent principalement sur l'Arabe moderne standard parmi lequel peu d'études ont étudié le cas des dialectes Arabes (Arabe familier), à savoir, Égyptien, Jordanien et Khaliji (dialecte utilisé dans les pays du Golfe). À notre connaissance, la recherche sur l'analyse des sentiments pour les dialectes Maghrébins ou l'Arabe Maghrébin (Algérien, Marocain et Tunisien)presque inexistant.

Le but de ce travail est de commencer une réflexion pour étudier l'analyse des sentiments pour le cas du dialecte Algérien, très différent par rapport aux autres dialectes arabes, non

seulement dans la prononciation, mais plutôt par ses différentes formes textuelles, très diverses et extrêmement riches, où on a utilisé l'une des méthodes de l'apprentissage profond.

Dans ce travail, nous avons commencé par la collection des données, où on a utilisé le Facebook et les journaux comme ressources d'extractions des données. Ensuite, on a prétraité les données ramassées dans le but d'éliminer les caractères non Arabe, les lettres répétées et la ponctuation (Tashkil). Et puis, on a entraîné un modèle Word2vec qui nous aide à vectoriser le dataset. Finalement, on a entraîné notre modèle de catégorisation des sentiments à l'aide des réseaux de neurones récurrents.

Ce mémoire comporte quatre chapitres et il est organisé de la manière suivante : le premier chapitre est consacré aux concepts de l'analyse des sentiments afin que les lecteurs connaissent bien le domaine de l'analyse des sentiments et leurs taches et les techniques utilisées. Le deuxième chapitre expose l'approche de l'apprentissage profond et ces architectures. Le troisième chapitre présente la conception du système à réaliser, son architecture globale et détaillée. Le quatrième chapitre explique l'implémentation du système, les expérimentations et les résultats obtenues. Le mémoire se termine par une conclusion générale contenant les perspectives envisagées.

Chapitre 1

Analyse des sentiments

1.1 Introduction

L'analyse des sentiments est l'un des domaines qui connaît un grand intérêt depuis une quinzaine d'années, et de nos jours il est très utilisé par les grandes firmes et les grands acteurs de l'informatique comme Google, Facebook, Microsoft, ...etc. Et il est utilisé même pour prédire le futur comme les pays développés font pour prédire Le candidat le plus populaire aux élections après l'analyse des tweets qui parle sur les candidats.

Dans ce chapitre, nous présentons les notions fondamentales d'analyse des sentiments. Après la définition générale de domaine de traitement automatique du langage naturel, d'un sentiment et d'une opinion, nous clarifions les types d'opinion, les niveaux d'analyse des sentiments, les taches de l'analyse des sentiments, les techniques utilisé pour réaliser cette analyse. Ensuite nous expliquons les difficultés d'application de l'analyse des sentiments sur la langue Arabe, et puis nous illustrons les défis de l'analyse des sentiments en général.

1.2 Définitions

Dans cette section, nous expliquons le domaine de traitement automatique du langage naturel où l'analyse des sentiments fait une partie de ce domaine, puis nous clarifions c'est quoi un sentiment et une opinion et la différence entre eux.

1.2.1 Traitement automatique du language naturel

(TAL ou TALN) est un domaine de l'informatique et de l'intelligence artificielle qui s'intéresse aux interactions entre les ordinateurs et les langages (naturels) humains, en particulier comment programmer des ordinateurs pour traiter de manière efficace de grandes quantités de données en langage naturel. Les défis dans le traitement du langage naturel impliquent fréquemment la reconnaissance de la parole, la compréhension du langage naturel et la génération du langage naturel.[48]

1.2.2 Opinion

Définition 1. Jugement, avis, sentiment qu'un individu ou un groupe émet sur un sujet, des faits, ce qu'il en pense.[1]

une opinion est un jugement, un point de vue ou une déclaration qui n'est pas concluante. Il peut traiter de questions subjectives dans lesquelles il n'y a pas de conclusion concluante, ou traiter de faits qui sont contestés par l'erreur logique que l'on a droit à leurs opinions. Ce qui distingue le fait de l'opinion, c'est que les faits sont plus susceptibles d'être vérifiables, c'est-à-dire qu'ils peuvent être acceptés par le consensus des experts. Un exemple est : "l'Algérie a été colonisé par la France" contre "la France a eu raison de coloniser l'Algérie". Une opinion peut être étayée par des faits et des principes, auquel cas elle devient un argument. Des personnes différentes peuvent tirer des conclusions opposées (opinions) même si elles sont d'accord sur le même ensemble de faits. Les opinions changent rarement sans que de nouveaux arguments soient présentés. On peut raisonner qu'une opinion est mieux soutenue par les faits qu'une autre en analysant les arguments à l'appui [13]. Dans un usage occasionnel, le terme d'opinion peut être le résultat de la perspective, de la compréhension, des sentiments particuliers, des croyances et des désirs d'une personne. Il peut se référer à des informations non corroborées, contrairement aux connaissances et aux faits.

1.2.3 Sentiment

Définion 2. État affectif complexe et durable lié à certaines émotions ou représentations.[2]

Le sentiment est la nominalisation du verbe sentir [4]. Le mot a d'abord été utilisé pour décrire la sensation physique du toucher à travers l'expérience ou la perception. Le mot est également utilisé pour décrire des expériences autres que la sensation physique du toucher, comme ≪un sentiment de chaleur≫ et de la sensibilité en général. En latin, sentire signifiait ressentir, entendre ou sentir. En psychologie, le mot est généralement réservé à l'expérience subjective consciente de l'émotion.[46] La phénoménologie et l'hétérophénoménologie sont des approches philosophiques qui fournissent une base pour la connaissance des sentiments. De nombreuses écoles de psychothérapie dépendent du fait que le thérapeute acquière une certaine compréhension des sentiments du client, pour lesquels il existe des méthodologies.

Les gens achètent des produits dans l'espoir que le produit leur donnera une certaine sensation : heureux, excité ou beau. Ou bien, ils trouvent le produit utile d'une manière ou d'une autre, même indirectement, par exemple pour soutenir une organisation caritative ou pour des raisons économiques altruistes. Certaines personnes achètent des produits de beauté dans l'espoir d'atteindre un état de bonheur ou un sens de la beauté de soi ou comme un acte ou une expression de la beauté. Les événements passés sont utilisés dans nos vies pour former des schémas dans nos esprits, et sur la base de ces expériences passées, nous attendons que

nos vies suivent un certain scénario. Cependant, la narration, la commémoration et la réserve de commémoration (la réticence à imposer des souvenirs), la recherche et l'investigation, et bien d'autres activités peuvent aider à régler des sentiments difficiles sans l'ambivalence de ce sentiment. ce qui n'est pas toujours vrai.

1.3 Types d'opinion

On peut distinguer deux types d'opinions la première s'appelle opinion régulière [28]. L'autre type est appelé opinion comparative [27]. En fait, nous pouvons également classer les opinions en fonction de la façon dont ils sont exprimés dans le texte, l'opinion explicite et l'opinion implicite.

1.3.1 Opinion régulière et opinion comparative

1.3.1.1 Opinion régulière

Une opinion régulière est souvent simplement considérée comme une opinion dans la littérature et il y a deux sous-types principaux [28]

Opinion directe : Une opinion directe fait référence à une opinion exprimée directement sur une entité ou un aspect de l'entité, par exemple, "La résolution de cet écran est excellente."

Opinion indirecte : Une opinion indirecte est une opinion exprimée indirectement sur une entité ou aspect d'une entité en fonction de ses effets sur d'autres entités. Ce sous-type se produit souvent dans le domaine médical. Par exemple, la phrase "Après l'injection du médicament, mes articulations senties pire" décrit un effet indésirable du médicament sur" mes articulations ", ce qui donne indirectement une opinion négative ou un sentiment au médicament. Dans le cas, l'entité est le médicament et l'aspect est l'effet sur les articulations.

Une grande partie de la recherche actuelle se concentre sur les opinions directes. Ils sont plus simples à manipuler. Les opinions indirects sont souvent plus difficiles à traiter. Par exemple, dans le domaine du médicament, il faut savoir si un état souhaitable et indésirable est avant ou après l'utilisation du médicament. Par exemple, la phrase "Puisque mes articulations étaient douloureuses, mon médecin m'a mis sur ce médicament" n'exprime pas un sentiment ou une opinion sur le médicament parce que "articulations douloureuses" (ce qui est négatif) est arrivé avant d'utiliser le médicament.

1.3.1.2 Opinion comparative

Un avis comparatif exprime une relation de similitudes ou de différences entre deux ou plusieurs entités et/ou une préférence du détenteur d'opinion sur la base de certains aspects partagés des entités [26]. Par exemple, les phrases ≪Dell est meilleur que HP≫ et ≪Dell est le

meilleur≫ expriment deux opinions comparatives. Une opinion comparative est habituellement exprimée en utilisant la forme comparative ou superlative d'un adjectif ou d'un adverbe, mais pas toujours (par exemple : l'utilisation de verbe préférer).

1.3.2 Opinion explicite et opinion implicite

1.3.2.1 Opinion explicite

Une opinion explicite est une déclaration subjective qui donne une opinion régulière ou comparative, par exemple : "Le couscous a bon goût" et "Facebook est mieux que twitter."

1.3.2.2 Opinion implicite

Une opinion implicite est une déclaration objective qui implique une opinion régulière ou comparative. Une telle déclaration objective exprime habituellement un fait souhaitable ou indésirable, par exemple : "La durée de vie de la batterie de l'ordinateur portable Toshiba est plus longue que celle de l'ordinateur portable HP."

1.4 Analyse de sentiment

L'analyse des sentiments, aussi appelée minage d'opinions, est le domaine d'étude qui analyse les opinions, les sentiments, les évaluations, les attitudes et les émotions des individus envers des entités telles que les produits, services, organisations, individus, problèmes, événements et leurs attributs. Cela représente un grand espace de problème. Ex : analyse de sentiment, extraction d'opinion, exploration de sentiments, analyse de subjectivité, analyse d'émotions, analyse de contenu, etc. Cependant, ils sont tous sous l'égide de l'analyse des sentiments ou minage d'opinion. Tandis que dans l'industrie, le terme analyse de sentiment est plus communément utilisé, dans l'université l'analyse de sentiment et l'extraction d'opinion sont fréquemment employées.

Peu importe, ils représentent essentiellement le même domaine d'étude. Le terme d'analyse des sentiments est peut-être apparu pour la première fois dans Nasukawa et Yi (2003) [36], et le terme minage d'opinion est apparu pour la première fois dans Dave et al.(2003) [15]. Cependant, la recherche sur les sentiments et les opinions est apparue plus tôt (Das et Chen 2001 [14], Morinaga et al 2002 [34], Pang et al 2002 [38], Tong 2001 [44], Turney 2002 [45] et Wiebe 2000 [49]). Dans ce mémoire, nous utilisons les termes analyse de sentiment et opinion minière interchangeable. Pour simplifier la présentation, tout au long de ce mémoire, nous utiliserons le terme d'opinion pour désigner l'opinion, le sentiment, l'évaluation, l'appréciation, l'attitude et l'émotion. Cependant, ces concepts ne sont pas équivalents. Nous les distinguerons au besoin. La signification de l'opinion elle-même est encore très large. L'analyse des sentiments et l'analyse de l'opinion portent principalement sur les opinions exprimés ou im-

pliqués des sentiments positifs ou négatifs.

Bien que la linguistique et le traitement du langage naturel (TALN) aient une longue histoire, peu de recherches ont été faites sur les opinions et les sentiments des gens avant l'an 2000. Depuis lors, le domaine est devenu un domaine de recherche très actif. Il y a plusieurs raisons à cela. Tout d'abord, il a un large éventail d'applications, presque dans tous les domaines. L'industrie entourant l'analyse des sentiments a également prospéré en raison de la prolifération des applications commerciales. Cela fournit une forte motivation pour la recherche. Deuxièmement, il offre de nombreux problèmes de recherche difficiles, qui n'avaient jamais été étudiés auparavant. Ce mémoire définira et discutera systématiquement de ces problèmes, et décrira les techniques actuelles de pointe pour les résoudre.

Troisièmement, pour la première fois dans l'histoire humaine, nous avons maintenant un énorme volume de données sur les médias sociaux sur le Web. Sans ces données, beaucoup de recherches n'auraient pas été possibles. Sans surprise, la création et la croissance rapide de l'analyse du sentiment coïncide avec celles des médias sociaux. En fait, l'analyse des sentiments est maintenant au centre de la recherche sur les médias sociaux. Ainsi, la recherche sur l'analyse des sentiments a non seulement un impact important sur la TALN, mais peut également avoir un impact profond sur les sciences de gestion, les sciences politiques, l'économie et les sciences sociales car elles sont toutes influencées par les opinions. Bien que la recherche sur l'analyse des sentiments ait commencé au début des années 2000, certains travaux antérieurs sur l'interprétation des métaphores, les adjectifs sentimentaux, la subjectivité, les points de vue et les affects (Hatzivassiloglou et McKeown, 1997 [20] ; Hearst, 1992[22] ; Wiebe, 1990 [50], 1994 [51] ; Wiebe et al., 1999 [52]).

1.5 Niveaux d'analyse des sentiments

En général, l'analyse des sentiments a été étudiée principalement à trois niveaux.

1.5.1 Niveau document

La tâche à ce niveau est de classer si un document d'opinion entier exprime un sentiment positif ou négatif (Pang et al., 2002 [38], Turney, 2002 [45]). Par exemple, à la suite d'une revue de produit, le système détermine si l'avis exprime une opinion globale positive ou négative sur le produit. Cette tâche est communément appelée classification de sentiment au niveau du document. Ce niveau d'analyse suppose que chaque document exprime des opinions sur une seule entité (par exemple, un seul produit). Ainsi, il ne s'applique pas aux documents qui évaluent ou comparent plusieurs entités.

1.5.2 Niveau phrase

La tâche à ce niveau va aux phrases et détermine si chaque phrase exprime une opinion positive, négative ou neutre. Neutre signifie généralement aucune opinion. Ce niveau d'analyse est étroitement lié à la classification de la subjectivité (Wiebe et al., 1999 [52]), qui distingue les phrases (appelées phrases objectives) qui expriment des informations factuelles, de phrases (appelées phrases subjectives) exprimant des opinions et des opinions subjectives. Cependant, nous devons noter que la subjectivité n'est pas équivalente au sentiment car de nombreuses phrases objectives peuvent impliquer des opinions, par exemple : ≪Nous avons acheté un nouvel ordinateur portable le mois dernier et l'écran est tombé en panne." Les chercheurs ont également analysé les clauses (Wilson et al., 2004 [54]), mais le niveau de la clause n'est toujours pas suffisant, par exemple, "L'Algérie se porte très bien dans cette crise économique".

1.5.3 Niveau aspect

Les analyses au niveau du document et de la phrase ne permettent pas de découvrir exactement ce que les gens aimaient et n'aimaient pas. Le niveau d'aspect effectue une analyse plus fine. Le niveau d'aspect était auparavant appelé niveau caractéristique (extraction d'opinion basée sur les caractéristiques et résumé) (Hu et Liu, 2004 [25]). Au lieu de regarder des constructions de langage (documents, paragraphes, phrases, clauses ou expressions), le niveau d'aspect regarde directement l'opinion elle-même. Il est basé sur l'idée qu'une opinion consiste en un sentiment (positif ou négatif) et une cible (d'opinion).

Une opinion, sans que sa cible soit identifiée, est d'une utilité limitée. Réaliser l'importance des cibles d'opinion nous aide également à mieux comprendre le problème de l'analyse des sentiments. Par exemple, bien que la phrase "Bien que le service n'est pas génial, j'aime toujours Cet hôtel. "a clairement un ton positif, on ne peut pas dire que cette phrase soit entièrement positive. En fait, la phrase est positive sur l'hôtel (souligné), mais négative sur son service (non souligné). Dans de nombreuses applications, les cibles d'opinion sont décrites par les entités et / ou leurs différents aspects. Ainsi, le but de ce niveau d'analyse est de découvrir les sentiments sur les entités et / ou leurs aspects. Par exemple, la phrase "La performance de Dell est bonne, mais sa durée de vie de la batterie est courte." Évalue deux aspects : performance et autonomie de la batterie, de Dell (entité).

Le sentiment sur la performance de Dell est positif, mais le sentiment sur sa durée de vie est négatif. La performance et la durée de vie de la batterie de Dell sont les cibles d'opinion. Sur la base de ce niveau d'analyse, un résumé structuré des opinions sur les entités et leurs aspects peut être produit, ce qui transforme le texte non structuré en données structurées et peut être utilisé pour toutes sortes d'analyses qualitatives et quantitatives. Les classifications au niveau du document et de la phrase sont déjà très difficiles.

Pour rendre les choses encore plus intéressantes et stimulantes, il existe deux types d'opinions, à savoir des opinions régulières et des opinions comparatives (Jindal et Liu, 2006b [27]).

Une opinion régulière exprime un sentiment seulement sur une entité particulière ou sur un aspect de l'entité, par exemple, "Couscous a un goût très bon", ce qui exprime un sentiment positif sur le goût de Couscous. Un avis comparatif compare plusieurs entités en fonction de certains de leurs aspects communs, par exemple, ≪Couscous a meilleur goût que Chakhchoukha≫, ce qui compare Couscous et Chakhchoukha en fonction de leurs goûts (un aspect) et exprime une préférence pour Couscous.

1.6 Tâches de l'analyse des sentiments

Nous allons, dans cette section, aborder les diférentes tâches qui composent un système d'analyse de sentiments. Ce plan se réfère principalement au modèle de (Liu, 2012 [25]) et fournit une définition de chaque tâche. Nous nous focaliserons par la suite sur la tâche de catégorisation de sentiments.

1.6.1 Analyse de la subjectivité et détection de l'opinion

L'analyse de la subjectivité et détection de l'opinion consiste à déterminer si un texte donné contient une opinion ou non. Ce problème a été abordé dans un premier temps indépendamment de l'analyse de sentiments avant de devenir une tâche de base, mais elle n'en reste pas moins l'une des plus difficiles.

La recherche dans la détection automatique de l'opinion à partir du texte a été initié par (Wiebe et al., 1999 [52]) avec des travaux où ils proposent des méthodes discriminatives entre le texte objectif et le texte subjectif au niveau document, phrase et expression en utilisant un classifieur Naïve Bayes. Ce classifieur utilise un ensemble de caractéristiques à savoir la présence ou l'absence de classes syntaxiques particulières, la ponctuation et la position des phrases. Ces caractéristiques sont jugées indicatrices de subjectivité.

Par la suite, (Hatzivassiloglou and Wiebe, 2000 [21]) démontrent que les adjectifs gradables[1] automatiquement détectés sont une caractéristique utile pour la classification de la subjectivité. Plus récemment, (Wilson et al., 2005 [53]) ont efectué un travail pour la classification de la subjectivité au niveau document en utilisant l'algorithme des k plus proches voisins basé sur le nombre total de mots et expressions de subjectivité dans chaque document.

1.6.2 Catégorisation des sentiments

La tâche de catégorisation de sentiments consiste à déterminer si un texte exprime une opinion positive ou négative de son auteur vis-à-vis d'un sujet du texte. Cette tâche utilise les techniques de traitement du langage naturel et d'apprentissage automatique qui seront détaillées par la suite.

1. des adjectifs qui peuvent être employés avec des intensificateurs tels que très ou peu.

1.6.3 Identification du sujet et du porteur d'opinion

Une autre tâche de base de l'analyse de sentiments est la détection du porteur d'opinion et l'identification du sujet. L'avantage de cette tâche est de pouvoir filtrer les opinions selon un sujet particulier ou alors de regrouper les opinions d'une personne particulière pour des fins de personnalisation en sélectionnant les sujets que ce dernier préfère.

1.6.4 Résumé de l'opnion

Les applications de l'analyse de sentiments requièrent l'étude des opinions de beaucoup de personnes car un seul avis ne suffit pas, de ce fait, une certaine forme de résumé s'impose (Liu, 2012 [29]). La récapitulation d'opinions consiste finalement à générer un résumé concis et digeste d'un grand nombre d'opinions. (Hu and Liu, 2004 [25]) sont les premiers à proposer des résumés basés sur les aspects à partir de critiques de clients vis-à-vis des produits vendus en ligne. Ils résument leur travail en trois étapes qui sont :

— 1. identification des aspects du produit que les clients ont mentionnés dans leurs opinions.
— 2. identification des phrases qui contiennent une opinion positive ou négative pour chaque aspect.
— 3. production d'un résumé en utilisant les informations découvertes.

1.6.5 Détection de l'ironie et du sarcasme

Le sarcasme [2] et l'ironie [3] sont considérés dans l'analyse de sentiments comme des modificateurs de la polarité, de la même manière que la négation (Liu, 2012 [29]). La détection de ces derniers est importante pour identifier correctement les opinions présentes dans les textes. La compréhension des phrases sarcastiques n'est pas toujours facile, même pour les humains, ainsi une solution informatique est une tâche intéressante et difficile. L'approche générale pour la détection du sarcasme est basée sur l'apprentissage automatique en utilisant des traits lexicaux simples en complément de dictionnaires.

1.6.6 Détection des spams

Aujourd'hui, à travers les réseaux sociaux, les blogs et les micro-blogs, il est très facile pour les gens d'exprimer leurs opinions d'une façon anonyme. Malgré ses avantages, l'anonymat a produit de nouvelles difficultés pour l'analyse de l'opinion. Il permet aux gens avec des intentions malveillantes fausser les résultats des systèmes en postant de faux avis afin de promouvoir ou de discréditer des produits cibles, des services, des organisations ou des individus sans divulguer leurs véritables intentions. La tâche de la détection des spams vise

2. désigne le fait de dire le contraire de ce que l'on pense sans laisser de signes indicatifs.
3. consiste à dire le contraire de ce que l'on pense en le faisant comprendre par des signes.

essentiellement à repérer ces gens (les spammeurs d'opinion) afin d'assurer la fiabilité des sources. Contrairement à l'extraction d'opinions, la détection de spams n'est pas seulement un problème de traitement du langage naturel car elle est considérée aussi comme étant un problème d'extraction de données.

1.7 Techniques

Dans cette section on va présenter les techniques utilisées pour réaliser la tâche de catégorisation(classifi des sentiments. on peut illustrer les techniques dans la figure suivante :

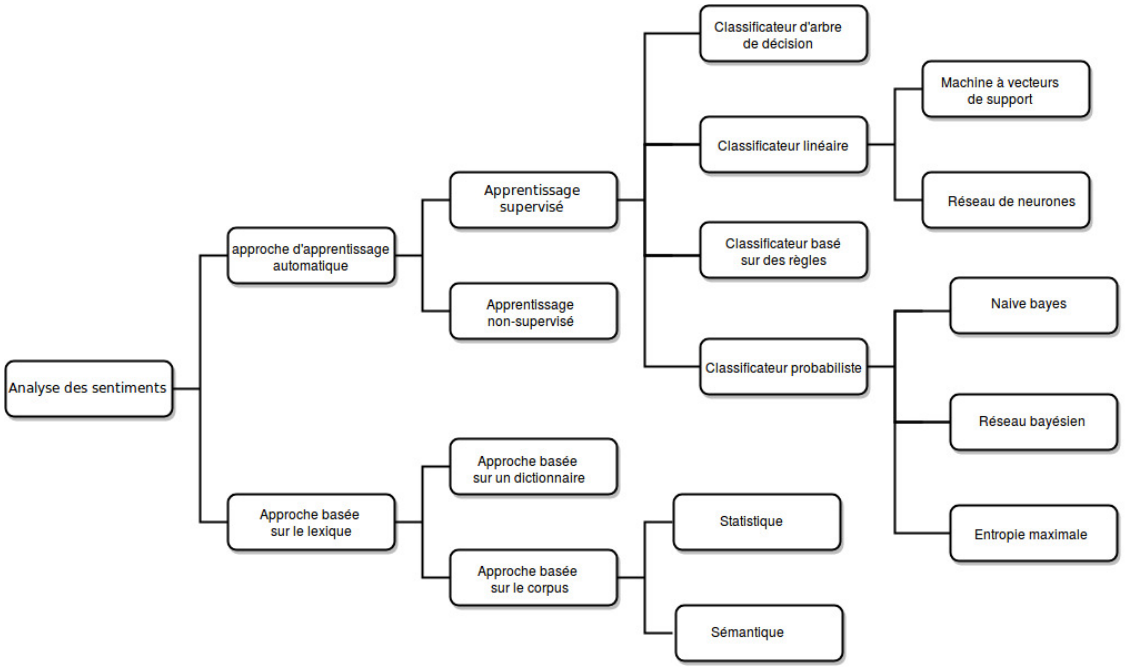


FIGURE 1.1: Les techniques utilisées dans l'analyse des sentiments.

1.7.1 Apprentissage automatique

L'apprentissage automatique est un domaine de l'informatique qui donne aux systèmes informatiques la capacité d' "apprendre" (c'est-à-dire d'améliorer progressivement les performances sur une tâche spécifique) avec des données, sans être explicitement programmé. Il existe beaucoup de techniques d'apprentissage automatiques comme : machine à vecteurs de support, les réseaux de neurones, classification naïve bayésienne, réseau bayésien, principe d'entropie maximale.

Dans cette section Nous nous focaliserons sur les réseaux de neurones(la technique utilisée dans ce mémoire) et la machine à vecteurs de support.

1.7.1.1 machine à vecteurs de support

Machine à vecteurs de support sont des modèles d'apprentissage supervisé avec des algorithmes d'apprentissage associés qui analysent les données utilisées pour la classification et

l'analyse de régression. Étant donné un ensemble d'exemples d'apprentissage, chacun marqué comme appartenant à l'une ou l'autre des deux catégories, un algorithme d'apprentissage SVM construit un modèle qui attribue de nouveaux exemples à une catégorie ou à une autre, ce qui en fait un classificateur binaire linéaire non probabiliste. telles que la mise à l'échelle Platt existe pour utiliser SVM dans un cadre de classification probabiliste). Un modèle SVM est une représentation des exemples sous la forme de points dans l'espace, mappés afin que les exemples des catégories séparées soient divisés par un espace libre aussi large que possible. De nouveaux exemples sont ensuite cartographiés dans ce même espace et prédits pour appartenir à une catégorie en fonction de quel côté de l'écart ils tombent.

En plus d'effectuer une classification linéaire, les SVM peuvent effectuer efficacement une classification non-linéaire en utilisant ce que l'on appelle l'astuce du noyau, en mettant implicitement en correspondance leurs entrées dans des espaces de caractéristiques de grande dimension.

Lorsque les données ne sont pas étiquetées, l'apprentissage supervisé n'est pas possible, et une approche d'apprentissage non supervisée est nécessaire, qui tente de trouver un regroupement naturel des données à des groupes, puis mapper de nouvelles données à ces groupes formés. L'algorithme de clustering de vecteurs de support [11] créé par Hava Siegelmann et Vladimir Vapnik applique les statistiques des vecteurs de support développés dans l'algorithme des machines vectorielles de support pour classer les données non étiquetées et est l'un des algorithmes de clustering les plus utilisés dans les applications industrielles.

1.7.1.2 Réseaux des neurones

Un réseau des neurones artificiel (RNA) est basé sur une collection d'unités ou de nœuds connectés appelés neurones artificiels (une version simplifiée des neurones biologiques dans un cerveau animal). Chaque connexion (une version simplifiée d'une synapse) entre neurones artificiels peut transmettre un signal de l'un à l'autre. Le neurone artificiel qui reçoit le signal peut le traiter puis signaler les neurones artificiels qui lui sont connectés.

Dans les implémentations RNA courantes, le signal d'une connexion entre neurones artificiels est un nombre réel, et la sortie de chaque neurone artificiel est calculée par une fonction non linéaire de la somme de ses entrées. Les neurones artificiels et les connexions ont généralement un poids qui s'ajuste à mesure que l'apprentissage progresse. Le poids augmente ou diminue la force du signal lors d'une connexion. Les neurones artificiels peuvent avoir un seuil tel que seulement si le signal agrégé franchit ce seuil, c'est le signal envoyé.

Typiquement, les neurones artificiels sont organisés en couches. Différentes couches peuvent effectuer différents types de transformations sur leurs entrées. Les signaux voyagent de la première (entrée) à la dernière (sortie) couche, éventuellement après avoir traversé les couches plusieurs fois.

Dans le deuxième chapitre on va bien détaillé le fonctionnement des RNA et spécialement le réseau des neurones récurrents dans l'analyse des sentiments.

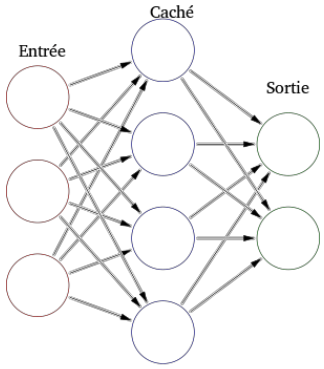


FIGURE 1.2: Un réseau des neurones artificiel simple.

1.7.2 Approche basé sur le lexique

Les approches basées sur le lexique reposent principalement sur un lexique de sentiment, c'est-à-dire une collection de termes, phrases et même idiomes de sentiment connus et précompilés, développés pour des genres de communication traditionnels, tels que le lexique Opinion Finder ; mais, même des structures plus complexes comme des ontologies ou des dictionnaires mesurant l'orientation sémantique des mots ou des phrases peuvent être utilisés à cette fin. Deux sous-classifications peuvent être trouvées ici : Approches basées sur un dictionnaire et sur un corpus.[29]

1.7.2.1 approche basé sur le dictionnaire

Le premier est généralement basé sur l'utilisation d'un ensemble initial de termes (graines) qui sont habituellement collectés et annotés de manière manuelle. Cet ensemble se développe en recherchant les synonymes et les antonymes d'un dictionnaire. Un exemple de ce dictionnaire pourrait être WordNet, qui a été utilisé pour développer un thésaurus appelé SentiWordNet. Le principal inconvénient de ce type d'approche est l'incapacité de traiter les orientations spécifiques au domaine et au contexte, même ainsi, cela pourrait être une solution intéressante selon le problème.[29]

1.7.2.2 approche basé sur le corpus

Les techniques basées sur le corpus ont pour objectif de fournir des dictionnaires liés à un domaine spécifique. Ces dictionnaires sont générés à partir d'un ensemble de termes d'opinion de la graine qui se développe à travers la recherche de mots apparentés au moyen de l'utilisation de techniques statistiques ou sémantiques. Des méthodes basées sur des statistiques telles que l'analyse sémantique latente (LSA), ou simplement la fréquence d'occurrence des mots dans une collection de documents peuvent être utilisées. Et d'autre part, les méthodes sémantiques telles que l'utilisation de synonymes et d'antonymes ou de relations à partir de thésaurus comme WordNet peuvent également représenter une solution intéressante.[29]

1.8 L'Arabe et l'analyse des sentiments

Le dialecte Algérien ou l'Arabe algérien(DALG) est considéré comme l'un des dialectes Arabes les plus ≪difficiles à comprendre≫. Il est beaucoup moins normalisé et standardisé par rapport à l'Arabe stadard moderne. Il a un vocabulaire inspiré de l'Arabe mais les mots originaux ont été modifiés phonologiquement [31]. DALG appartient à l'Arabe maghrébin (groupe occidental) et est principalement utilisé dans la vie quotidienne. Il est caractérisé par l'absence de ressources d'écriture, donc il est considéré comme un langage sous-financé [41].

DALG diffère de ASM et d'autres dialectes Arabes en ayant de nombreuses caractéristiques spécifiques. Outre le ASM et l'Arabe dialectal, un vocabulaire riche constitué de mots étrangers d'origine française constitue une partie essentielle de la langue parlée des Algériens.

La phonologie, la morphologie, le lexique et la syntaxe de DALG sont très difficiles à comprendre pour les citoyens des autres pays Arabes. Pour des raisons historiques, DALG s'est enrichie de nombreuses langues (Tamazight,Turques, Italiennes, Espagnoles et surtout Françaises) ce qui a engendré une situation linguistique complexe.

Avec l'arrivée des réseaux sociaux, DALG est de plus en plus utilisé par les utilisateurs web Algériens selon l'UIT [4], 28% des Algériens utilisent activement l'internet. La plupart de cette activité est dominée par l'utilisation des réseaux sociaux. Des millions de commentaires et des vues sont ajoutées tous les jours. Extraire ce volume énorme de commentaires et de critiques nécessite de prendre en compte des aspects particuliers de DALG.

1.9 Conclusion

L'analyse des sentiments est un domaine intéressant, où ce domaine est largement utilisé par les grandes entreprises et les grandes firmes pour avoir une idée de la façon dont les clients sont heureux avec les produits à partir du rapport entre les tweet positifs et négatifs à leur sujet. Il peut également être utilisé pour trouver des personnes qui sont satisfaites des produits ou services et leurs expériences peuvent être utilisées pour promouvoir ces produits.

L'analyse des sentiments est une tâche un peu complexe et a besoin de grands efforts surtout avec les langages vernaculaires écrits dans les réseaux sociaux, comme on a vu dans ce chapitre que l'Arabe Algérien écrits dans le web est très complexe, et nécessite une concentration sur l'orthographe et le vocabulaire du dialecte.

4. International Telecommunication Union

Chapitre 2

Apprentissage profond

2.1 Introduction

L'apprentissage profond est un sous-domaine de l'apprentissage automatique, qui connaît un grand intérêt depuis l'année 2012, grace à ces résultats et à sa précision où des fois il touche plus de 95%, cette méthode est actuellement utilisée dans plusieurs applications comme : la reconnaissance automatique de la parole, reconnaissance de l'image, traitement de language naturel, découverte de médicament et toxicologie, bioinformatique ...etc.

Dans ce chapitre, nous présentons les notions fondamentales de l'apprentissage profond. Après la définition générale des réseaux de neurones et de l'apprentissage profond, nous clarifions les paradigmes d'apprentissage, les architectures d'apprentissage profond, la classification, et la motivation derrière l'utilisation de l'apprentissage profond. Ensuite nous expliquons les applications de l'apprentissage profond, et puis nous illustrons la relation entre l'apprentissage profond et l'analyse des sentiments.

2.2 Définitions

2.2.1 Réseaux de neurones

Le réseau de neurones est basé sur une collection d'unités ou de nœuds connectés appelés neurones artificiels (une version simplifiée des neurones biologiques dans un cerveau animal). Chaque connexion (une version simplifiée d'une synapse) entre neurones artificiels peut transmettre un signal de l'un à l'autre. Le neurone artificiel qui reçoit le signal peut le traiter puis signaler les neurones artificiels qui lui sont connectés.[47]

2.2.2 Apprentissage profond

L'apprentissage profond est un type particulier d'apprentissage automatique qui atteint une grande puissance et flexibilité en apprenant à représenter le monde comme une hiérarchie imbriquée de concepts, chaque concept étant défini par rapport à des concepts plus simples et des représentations plus abstraites calculées en termes moins abstraits.[18]

L'apprentissage profond permet à l'ordinateur de construire des concepts complexes à partir de concepts plus simples. La figure 2.1 montre comment un système d'apprentissage profond peut représenter le concept d'image d'une personne en combinant des concepts plus simples, tels que des coins et des contours, qui sont à leur tour définis en termes d'arêtes.

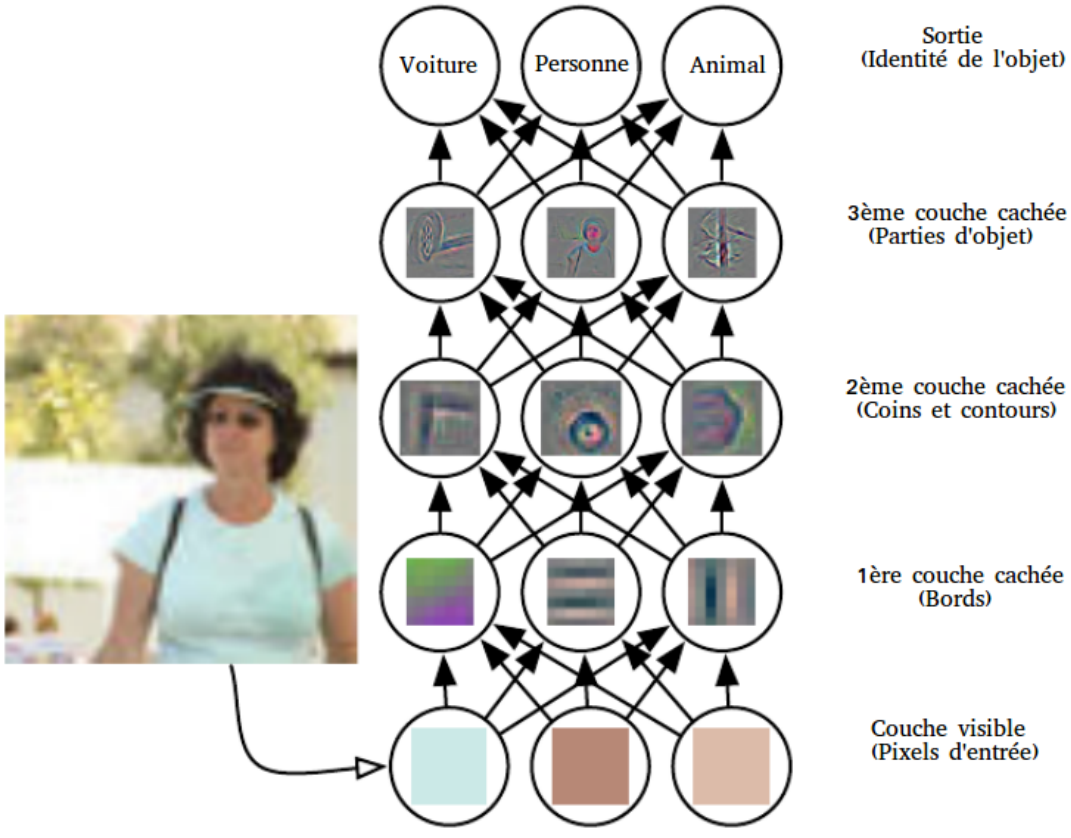


FIGURE 2.1: Illustration d'un model de l'apprentissage profond.[18]

2.3 Les types d'apprentissage automatique

La plupart des algorithmes d'apprentissage automatique peuvent être divisés en catégories d'apprentissage supervisé et apprentissage non supervisé, mais il y a aussi d'autres types comme l'apprentissage par renforcement et l'apprentissage semi-supervisé.

2.3.1 Apprenstissage supervisé

Les algorithmes d'apprentissage supervisé subissent des données contenant des caractéristiques, mais chaque exemple est également associé à une étiquette ou une cible. Par exemple, notre ensemble de données est annoté avec positive ou négative. Un algorithme d'apprentissage supervisé peut étudier l'ensemble de ces données et apprendre à classer les commentaires en classes différentes en fonction de leurs sentiments soit négative soit positive.

2.3.2 Apprentissage non-supervisé

Une tâche d'apprentissage non-supervisée classique consiste à trouver la ≪meilleure≫ représentation des données. Par ≪meilleur≫, nous pouvons dire différentes choses, mais en général, nous cherchons pour une représentation qui conserve autant d'informations sur x que possible tout en obéir à une pénalité ou à une contrainte visant à garder la représentation plus simple ou plus accessible que x lui-même.[18]

2.3.3 Apprentissage par renforcement

L'apprentissage par renforcement est une approche de l'apprentissage automatique qui s'inspire de la psychologie behavioriste. Il est similaire à la façon dont un enfant apprend à effectuer une nouvelle tâche. L'apprentissage par renforcement contraste avec d'autres approches d'apprentissage automatique en ce sens que l'algorithme n'est pas explicitement dit comment exécuter une tâche, mais travaille seul sur le problème.

En tant qu'agent, qui peut être une voiture autonome ou un programme jouant aux échecs, interagit avec son environnement, reçoit un état de récompense en fonction de sa performance, comme conduire à destination en toute sécurité ou gagner un match. Réciproquement, l'agent reçoit une pénalité pour avoir mal exécuté, comme sortir de la route ou être maté.[43]

2.3.4 Apprentissage semi-supervisé

L'apprentissage semi-supervisé utilisent également des données non étiquetées pour l'apprentissage, généralement une petite quantité de données étiquetées avec une grande quantité de données non étiquetées. L'apprentissage semi-supervisé se situe entre un apprentissage non supervisé (sans données d'entraînement étiquetées) et un apprentissage supervisé (avec des données d'entraînement complètement étiquetées). De nombreux chercheurs en apprentissage

automatique ont découvert que les données non étiquetées, lorsqu'elles sont utilisées conjointement avec une petite quantité de données étiquetées, peuvent entraîner une amélioration considérable de la précision de l'apprentissage.[12]

2.4 Architectures d'apprentissage profond

L'apprentissage profond n'est pas une approche unique, mais plutôt une classe d'algorithmes et de topologies que vous pouvez appliquer à un large éventail de problèmes. Bien que l'apprentissage profond ne soit certainement pas nouveau, il connaît une croissance explosive en raison de l'intersection de réseaux neuronaux à couches profondes et de l'utilisation de GPU pour accélérer leur exécution. Le big data a également alimenté cette croissance. Parce que l'apprentissage en profondeur repose sur des algorithmes d'apprentissage supervisé (ceux qui entraînent les réseaux de neurones avec des exemples de données et les récompensent en fonction de leur succès), plus il y a de données, mieux c'est.

Dans cette section nous présentons les architectures de l'apprentissage profond les plus utilisé comme : réseau de neurones récurrents, réseau de neurones convolutif, réseaux adversatifs génératifs.

2.4.1 Réseau de neurones récurrents

Le réseau de neurone récurrent(recursive neural networks RNN en anglais) est l'une des architectures de réseau fondamentales à partir desquelles d'autres architectures d'apprentissage profond sont construites. La principale différence entre un réseau multicouche typique et un réseau récurrent est que les connexions sont complètement anticipées, un réseau récurrent peut avoir des connexions qui sont réinjectées dans des couches précédentes (ou dans la même couche). Cette rétroaction permet aux réseau de neurone récurrent de conserver la mémoire des entrées passées et des problèmes de modèle dans le temps.

Les réseaux de neurone récurrent sont constitués d'un ensemble riche d'architectures (nous allons voir une topologie populaire appelée LSTM qu'on va ensuite l'utiliser dans notre projet). Le différentiateur clé est la rétroaction dans le réseau, qui pourrait se manifester à partir d'une couche cachée, la couche de sortie, ou une combinaison de ceux-ci.

Ils existent plusieurs type de réseau de neurone récurent comme : fully recurrent networks, recursive neural networks, neural history compressor, gated recurrent unit neural networks et long short-term memory networks (LSTM) qui est l'architecture qu'on va l'utiliser.

Long short-term memory networks (LSTM) :

en français réseau récurrent à mémoire court et long terme ou plus explicitement réseau de neurones récurrents à mémoire court-terme et long terme, est l'architecture de réseau de neurones récurrents la plus utilisée en pratique qui permet de répondre au problème de disparition de gradient. Le réseau LSTM a été proposé par Sepp Hochreiter et Jürgen Schmidhuber en 1997 [24]. L'idée associée au LSTM est que chaque unité computationnelle est liée non seulement à un état caché mais également à un état de la cellule qui joue le rôle de mémoire. Le passage à se fait par transfert à gain constant et égal à 1.

De cette façon les erreurs se propagent aux pas antérieurs (jusqu'à 1 000 étapes dans le passé) sans phénomène de disparition de gradient. L'état de la cellule peut être modifié à travers une porte qui autorise ou bloque la mise à jour (input gate). De même une porte contrôle si l'état de cellule est communiqué en sortie de l'unité LSTM (output gate). La version la plus répandue des LSTM utilise aussi une porte permettant la remise à zéro de l'état de la cellule (forget gate) [17] .

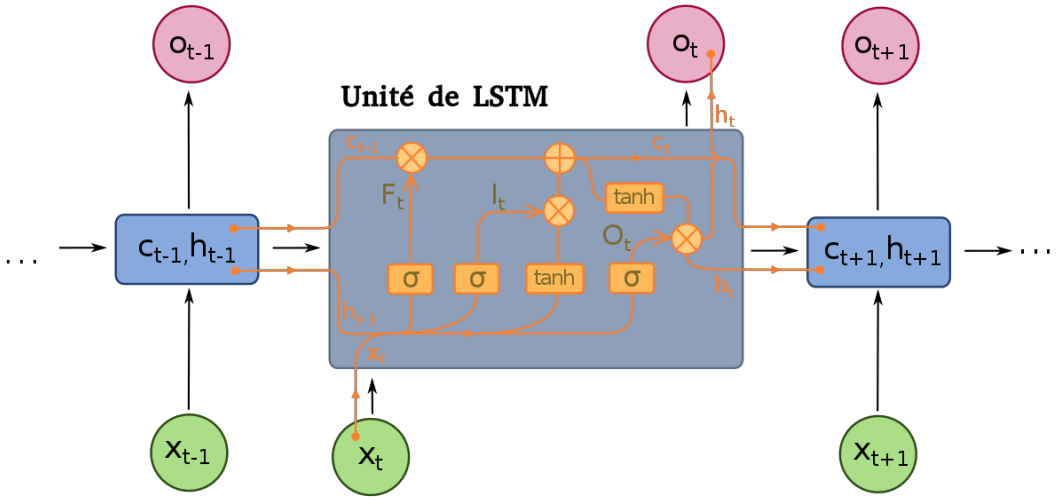


FIGURE 2.2: Schéma d'une unité de réseau LSTM. [18]

Les mathématiques derrière réseau LSTM :

Valeurs initiales : $c_0 = 0$ et $h_0 = 0$. L'opérateur . (point) symbolise le produit matriciel de Hadamard (produit terme à terme). Les symboles σ et tanh représentent respectivement la fonction sigmoïde et la fonction tangente hyperbolique.

$$F_t = \sigma(W_F x_t + U_F h_{t-1} + b_F) \quad (forget\,gate)$$

$$I_t = \sigma(W_I x_t + U_I h_{t-1} + b_I) \quad (input\,gate)$$

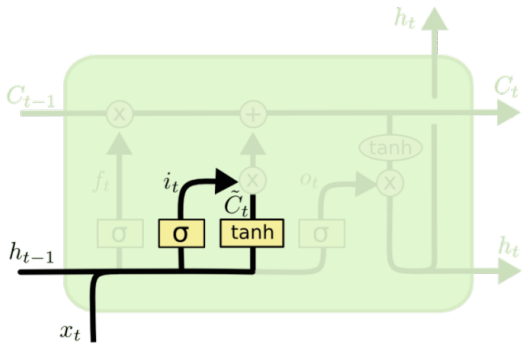
$$O_t = \sigma(W_O x_t + U_O h_{t-1} + b_O) \quad (output\,gate)$$

$$c_t = F_t \,.\, c_{t-1} + I_t \,.\, tanh(W_c x_t + U_c h_{t-1} + b_c)$$

$$h_t = O_t \,.\, tanh(c_t)$$

On peut résumer les LSTMs étape par étape comme suit :

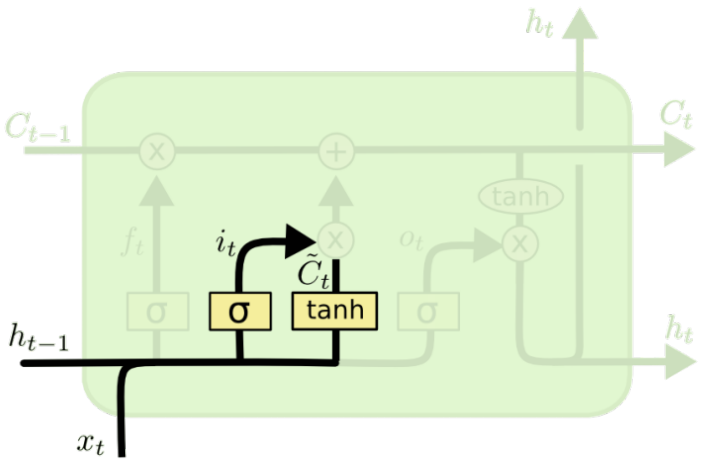
La première étape dans notre LSTM est de décider quelles informations nous allons jeter de l'état de la cellule. Cette décision est prise par une couche sigmoïde appelée «couche de porte d'oubli». Elle regarde h_{t-1} et x_t, et génère un nombre entre 0 et 1 pour chaque nombre dans l'état de cellule C_{t-1}. Un 1 représente "complètement garder ceci" tandis qu'un 0 représente "complètement se débarrasser de ceci".



$$F_t = \sigma(W_F x_t + U_F h_{t-1} + b_F) \quad (forget\,gate)$$

FIGURE 2.3: La première étape dans LSTM

L'étape suivante consiste à décider quelles nouvelles informations nous allons stocker dans l'état de la cellule. Cela a deux parties. Tout d'abord, une couche sigmoïde appelée "couche de la porte d'entrée" décide quelles valeurs nous allons mettre à jour. Ensuite, une couche tanh crée un vecteur de nouvelles valeurs candidates, $\tilde{C}_t$, qui pourraient être ajoutées à l'état. Dans l'étape suivante, nous allons combiner ces deux pour créer une mise à jour de l'état.

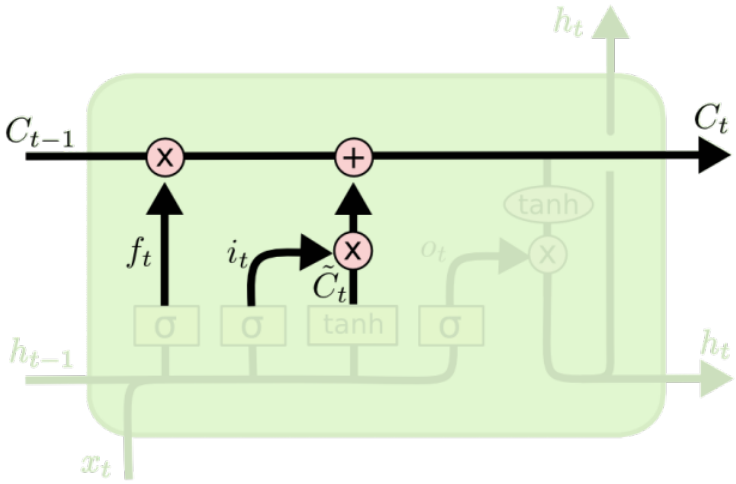


$$I_t = \sigma(W_I x_t + U_I h_{t-1} + b_I) \quad (input\ gate)$$
$$\tilde{C}_t = tanh(W_c x_t + U_c h_{t-1} + b_c)$$

FIGURE 2.4: La deuxième étape dans LSTM

Il est maintenant temps de mettre à jour l'ancien état de cellule, C_{t-1}, dans le nouvel état de cellule C_t. Les étapes précédentes ont déjà décidé quoi faire, nous devons juste le faire réellement.Nous multiplions l'ancien état par F_t, oubliant les choses que nous avons décidé d'oublier plus tôt.

Ensuite, nous l'ajoutons I_t*$\tilde{C}_t$. Ce sont les nouvelles valeurs candidates, mises à l'échelle de combien nous avons décidé de mettre à jour chaque valeur d'état.

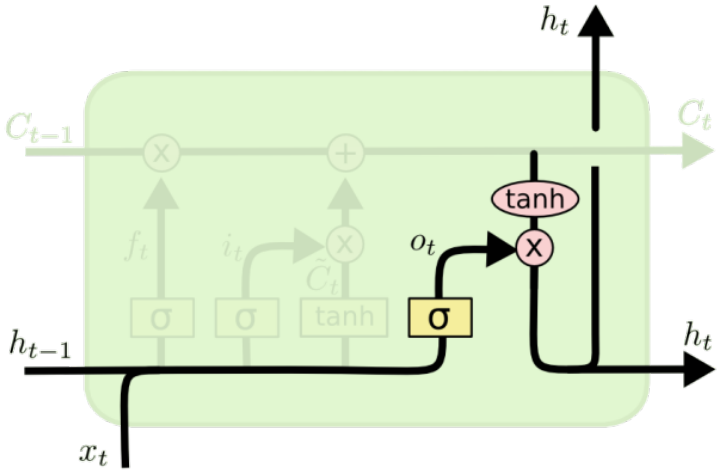


$$C_t = F_t \,.\, c_{t-1} + I_t \,.\, \tilde{C}_t$$

FIGURE 2.5: La troisième étape dans LSTM

Enfin, nous devons décider de ce que nous allons produire. Cette sortie sera basée sur notre état de cellule, mais sera une version filtrée. Tout d'abord, nous exécutons une couche sigmoïde qui détermine les parties de l'état de la cellule que nous allons produire. Ensuite, nous

mettons l'état de la cellule à travers tanh (pour pousser les valeurs entre -1 et 1) et nous le multiplions par la sortie de la porte sigmoïde, de sorte que nous ne produisons que les parties que nous avons décidées.



$$o_t = f(W_o h_t + b_o)$$
$$h_t = O_t \,.\, tanh(c_t)$$

FIGURE 2.6: La dérnière étape dans LSTM

2.4.2 Réseau de neurones convolutif

Un réseau de neurones convolutifs ou réseau de neurones à convolution (en anglais CNN ou ConvNet pour Convolutional Neural Networks) est un réseau neuronal multicouche inspiré biologiquement du cortex visuel animal. L'architecture est particulièrement utile dans les applications de traitement d'image. Le premier CNN a été créé par Yann LeCun[18] ; À l'époque, l'architecture était axée sur la reconnaissance de caractères manuscrits, comme l'interprétation de codes postaux. En tant que réseau profond, les premières couches reconnaissent les entités (telles que les arêtes) et les couches ultérieures recombinent ces entités en attributs de niveau supérieur de l'entrée.

L'architecture CNN est composée de plusieurs couches qui implémentent l'extraction de caractéristiques, puis la classification (voir l'image suivante). L'image est divisée en champs réceptifs qui alimentent une couche convolutionnelle, qui extrait ensuite des entités de l'image d'entrée. L'étape suivante est la mise en commun, qui réduit la dimensionnalité des entités extraites (par le biais d'un échantillonnage vers le bas) tout en conservant les informations les plus importantes (généralement via la mise en pool maximale). Une autre étape de convolution et de mise en commun est ensuite effectuée qui alimente un perceptron multicouche entièrement connecté. La couche de sortie finale de ce réseau est un ensemble de nœuds qui identifient les caractéristiques de l'image (dans ce cas, un nœud par numéro identifié). Vous formez le réseau en utilisant la rétropropagation.[18]

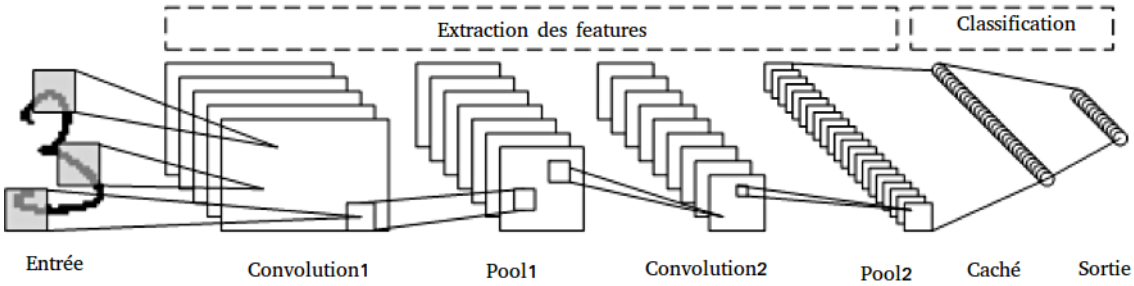


FIGURE 2.7: Architecture standard d'un réseau à convolutions.

L'utilisation de couches profondes de traitement, de convolutions, de mise en commun et d'une couche de classification entièrement connectée a ouvert la porte à diverses nouvelles applications des réseaux neuronaux d'apprentissage profond. En plus du traitement de l'image, le CNN a été appliqué avec succès à la reconnaissance vidéo et à diverses tâches dans le traitement du langage naturel.

Les applications récentes des CNN et des LSTM ont produit des systèmes de sous-titrage d'image et de vidéo dans lesquels une image ou une vidéo est résumée dans un langage naturel. Le CNN implémente le traitement de l'image ou de la vidéo, et le LSTM est formé pour convertir la sortie CNN en langage naturel.

2.4.3 Réseaux antagonistes génératifs

Réseaux antagonistes génératifs (en anglais GAN pour Generative Adversarial Networks), sont des modèles génératifs conçus par Goodfellow et al. en 2014 [19]. Dans une configuration GAN, deux fonctions différentiables, représentées par des réseaux de neurones, sont enfermées dans un jeu. Les deux acteurs (le générateur et le discriminateur) ont des rôles différents dans ce cadre.

Le générateur tente de produire des données provenant d'une distribution de probabilité. Ce serait d'essayer de reproduire les billets du parti.

Le discriminateur agit comme un juge. Il arrive à décider si l'entrée provient du générateur ou du vrai jeu d'entraînement. Ce serait la sécurité de la partie en comparant votre faux billet avec le vrai billet pour trouver des défauts dans votre conception.

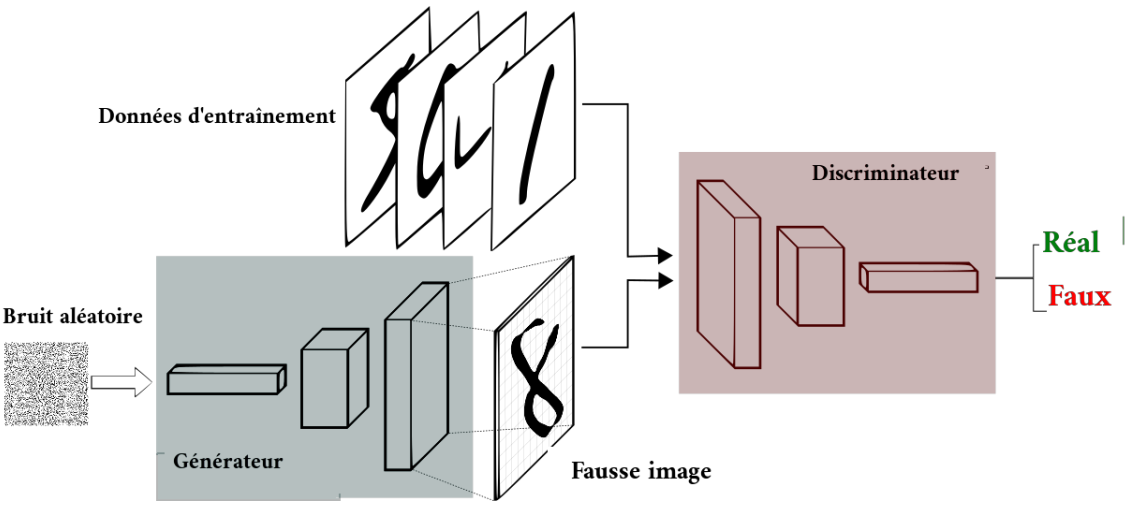


FIGURE 2.8: Architecture standard d'un réseaux antagonistes génératifs.

2.5 Classification

Dans l'apprentissage automatique et les statistiques, la classification est une approche d'apprentissage supervisé dans laquelle le programme informatique apprend à partir des données qui lui sont données en entrée et utilise ensuite cet apprentissage pour classer une nouvelle observation. Cet ensemble de données peut simplement être bi-classe (comme identifier si la personne est un homme ou une femme ou que le courrier est un spam ou un non-spam) ou il peut aussi être multi-classe.

Un algorithme qui implémente la classification, en particulier dans une implémentation concrète, est appelé classificateur. Le terme "classificateur" fait parfois référence à la fonction mathématique, implémentée par un algorithme de classification, qui mappe les données d'entrée à une catégorie.

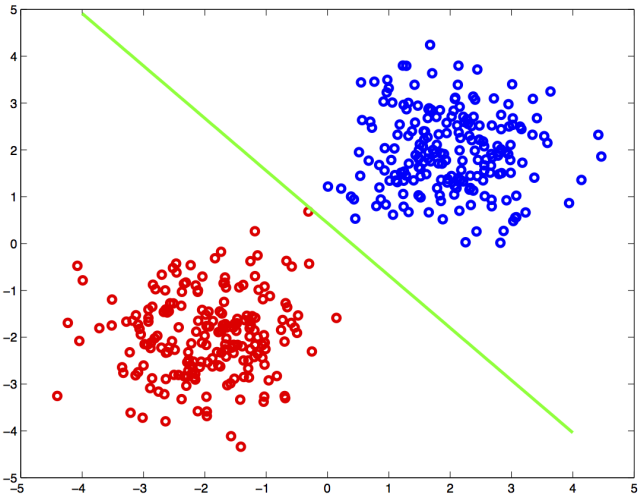


FIGURE 2.9: Un exemple d'un classificateur entre deux classe.

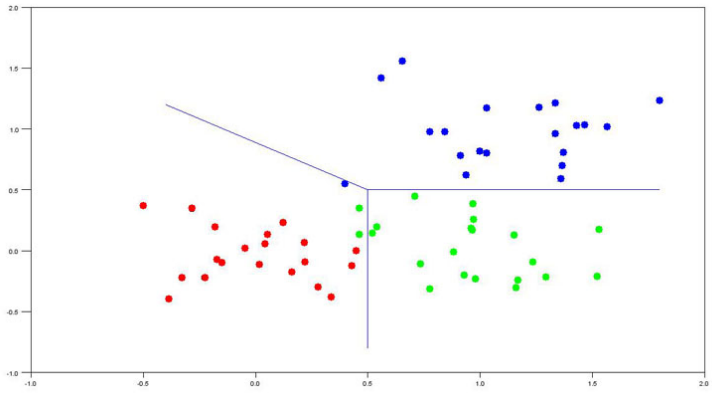


FIGURE 2.10: Un exemple d'un classificateur multi-classe.

Dans notre projet on essaye de classifier les commentaires ou les tweets(un texte en général), selon les sentiments existent dans le texte, si le texte contient des sentiments positifs alors le classificateur doit dire que ce texte est positif(1), sinon il doit dire que ce texte est négatif(0).

2.6 Les applications de l'apprentissage profond

2.6.1 Reconnaissance automatique de la parole

C'est le sous-domaine interdisciplinaire de la linguistique computationnelle qui développe des méthodologies et des technologies qui permettent la reconnaissance et la traduction de la langue parlée en texte par les ordinateurs. Il est également connu sous le nom de «reconnaissance automatique de la parole», «reconnaissance de la parole par ordinateur» ou simplement «speech to text». Il intègre des connaissances et des recherches dans les domaines de la linguistique, de l'informatique et de l'ingénierie électrique.[23]

2.6.2 Reconnaissance d'image

Est un domaine interdisciplinaire qui traite de la façon dont les ordinateurs peuvent acquérir une compréhension de haut niveau à partir d'images ou des vidéos numériques. Du point de vue de l'ingénierie, il cherche à automatiser les tâches que le système visuel humain peut faire.

Reconnaissance d'image comprend des méthodes pour acquérir, traiter, analyser et comprendre des images numériques, et extraire des données de grande dimension du monde réel afin de produire des informations numériques ou symboliques.[35]

2.6.3 Traitement du langage naturel

Est un domaine de l'informatique et de l'intelligence artificielle qui s'intéresse aux interactions entre les ordinateurs et les langages (naturels) humains, en particulier comment

programmer des ordinateurs pour traiter de manière efficace de grandes quantités de données en langage naturel.

Les défis dans le traitement du langage naturel impliquent fréquemment la reconnaissance de la parole, la compréhension du langage naturel et la génération du langage naturel.[18]

2.6.4 Les systèmes de recommandation

Un système de recommandation est un sous-classe du système de filtrage d'information qui cherche à prédire ≪l'évaluation≫ ou la ≪préférence≫ qu'un utilisateur donnerait à un article.

Les systèmes de recommandation sont devenus de plus en plus populaires ces dernières années et sont utilisés dans divers domaines, notamment les films, la musique, les actualités, les livres, les articles de recherche, les requêtes de recherche, les tags sociaux et les produits en général. Il existe également des systèmes de recommandation pour les experts, les collaborateurs, les blagues, les restaurants, les vêtements, les services financiers, l'assurance-vie.[40]

2.6.5 Bioinformatique

La bioinformatique est un domaine interdisciplinaire qui développe des méthodes et des outils logiciels pour la compréhension des données biologiques. En tant que domaine interdisciplinaire de la science, la bioinformatique combine l'informatique, la biologie, les mathématiques et l'ingénierie pour analyser et interpréter les données biologiques.

La bioinformatique a été utilisée pour des analyses de requêtes biologiques à l'aide de techniques mathématiques et statistiques. Plus largement, la bioinformatique est appliquée aux statistiques et à l'informatique en sciences biologiques.[33]

2.7 Les défis de l'apprentissage profond

L'apprentissage profond est devenu l'un des principaux domaines de recherche dans le développement de machines intelligentes. La plupart des applications bien connues (telles que la reconnaissance de la parole, le traitement d'image et le traitement automatique du langage) de l'intelligence artificielle sont pilotées par l'apprentissage profond. Les algorithmes d'apprentissage profond imitent les cerveaux humains en utilisant des réseaux neuronaux artificiels et apprennent progressivement à résoudre avec précision un problème donné. Mais il y a des défis importants dans les systèmes d'apprentissage profond que nous devons surveiller.

2.7.1 La masse des données

Les algorithmes d'apprentissage profond sont formés pour apprendre progressivement en utilisant des données. De grands ensembles de données sont nécessaires pour s'assurer que la machine fournit les résultats souhaités. Comme le cerveau humain a besoin de beaucoup d'expériences pour apprendre et déduire des informations, le réseau de neurones artificiels

nécessite une quantité importante de données. Plus l'abstraction est puissante, plus les paramètres doivent être réglés et plus de paramètres nécessitent plus de données.

Par exemple, dans l'analyse des sentiments les entrées sont des textes, et chaque texte contient des dizaines mots, et chaque mot est représenté par un vecteur de 200 nombres réels et peut-être plus, donc si on a un texte de 1000 mots on aura une matrice de (1000,200) de dimensions, et pour faire l'apprentissage on est besoin d'un corpus de 25000 textes au moins, donc on aura une matrice de 3 dimensions avec les dimensions M[25000][1000][200].

2.7.2 La nécessité d'un matériel de haute performance

L'entrainement d'un ensemble de données pour une solution d'apprentissage profond nécessite beaucoup de données. Pour effectuer une tâche afin de résoudre des problèmes du monde réel, la machine doit être équipée d'une puissance de traitement adéquat. Pour assurer une meilleure efficacité et une consommation de temps moindre, les scientifiques de données optent pour des GPU multi-cœurs très performants et des unités de traitement similaires. Ces unités de traitement sont coûteuses et consomment beaucoup de puissance.

2.7.3 Optimisation de l'hyperparamètre

Les hyperparamètres sont les paramètres dont la valeur est définie avant le début du processus d'apprentissage. La modification de la valeur de ces paramètres peut entraîner une modification importante des performances de votre modèle.

S'appuyer sur les paramètres par défaut et ne pas effectuer l'optimisation hyperparamétrique peut avoir un impact significatif sur les performances du modèle. En outre, avoir trop peu d'hyperparamètres et les ajuster manuellement plutôt que d'optimiser par des méthodes éprouvées est également un aspect de la performance.[18]

2.7.4 Surapprentissage dans les réseaux de neurones

Parfois, il y a une différence d'erreur nette dans l'ensemble de données d'apprentissage et l'erreur rencontrée dans un nouvel ensemble de données non vu. Cela se produit dans des modèles complexes, comme avoir trop de paramètres relatifs au nombre d'observations. L'efficacité d'un modèle est jugée par sa capacité à bien fonctionner sur un ensemble de données non visible et non par sa performance sur les données d'entraînement qui lui sont fournies.[18]

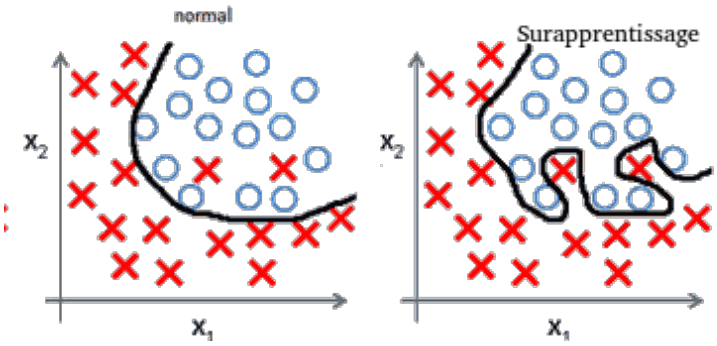


FIGURE 2.11: Un exemple d'un modèle avec un suraprentissage et un autre ordinaire.

En général, un modèle est généralement formé en maximisant ses performances sur un ensemble de données d'apprentissage particulier. Le modèle mémorise donc les exemples d'apprentissage mais n'apprend pas à généraliser à de nouvelles situations et ensembles de données.

2.7.5 Manque de flexibilité et de multitâche

Les modèles d'apprentissage profond, une fois formés, peuvent fournir une solution extrêmement efficace et précise à un problème spécifique. Cependant, dans l'état actuel, les architectures de réseau neuronal sont hautement spécialisées dans des domaines d'application spécifiques.

2.8 L'apprentissage profond et l'analyse des sentiments

Dans cette section nous présentons la relation entre l'analyse des sentiments et l'apprentissage profond, et les tâches nécessaires pour effectuer ce travail.

2.8.1 Préparation des données

L'un des points les plus importants dans l'apprentissage profond supervisé est, sans aucun doute ; celui de la disponibilité des données labellisées au préalable. En effet, la phase d'apprentissage requière une grande quantité d'exemples dont la classe est déjà connue. Ceci permettra à la fonction de prédire la classe de nouveaux documents en fonction des exemples déjà rencontrés lors de la phase d'apprentissage.

De plus, les données labellisées doivent être représentatives des données qui devront être catégorisées par la suite grâce au modèle construit. À titre d'exemple, si nous souhaitons prédire les sentiments des commentaires issus de réseaux sociaux, le corpus doit contenir le même type de documents.

2.8.2 Prétraitement

La phase de prétraitement fait appel à des techniques qui peuvent modifier la forme d'un mot ou l'éliminer complètement. Les techniques en question sont, à titre d'exemple la suppression des caractères spéciaux,les marques de ponctuation ou encore la suppression des mots

vides. On peut également appliquer des fonctions plus élaborées comme la lemmatisation ou la racinisation.

2.8.3 Word embedding

Word embedding ≪plongement de mots≫, ou ≪plongement lexical≫ en français, est un ensemble des techniques de modélisation et d'apprentissage dans le domaine du traitement automatique du langage naturel (TALN) où des mots ou des phrases du vocabulaire sont associés à des vecteurs de nombres réels. Conceptuellement, il implique une intégration mathématique d'un espace avec une dimension par mot à un espace vectoriel. Parmi ces techniques il y le Word2Vec qui est très utilisé dans le domaine du traitement automatique de langage en général et dans l'analyse des sentiments en particulier.

Word2vec :

Word2vec a été créé par une équipe de chercheurs dirigée par Tomas Mikolov chez Google. Et Word2Vec est un groupe de modèles associés utilisés pour produire des plongements de mots. Ces modèles sont des réseaux neuronaux à deux couches, peu profonds, formés pour reconstruire les contextes linguistiques des mots. Word2vec prend en entrée un grand corpus de texte et produit un espace vectoriel, typiquement de plusieurs centaines de dimensions, avec pour chaque mot unique du corpus un vecteur correspondant dans l'espace. Les vecteurs de mots sont positionnés dans l'espace vectoriel de sorte que les mots qui partagent des contextes communs dans le corpus sont situés à proximité les uns des autres dans l'espace [32] (voir la figure suivante).

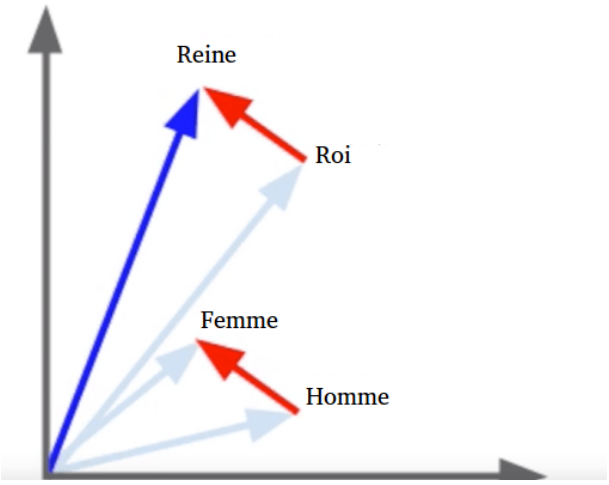


FIGURE 2.12: Un exemple simplifié d'un Word2Vec.

2.9 Conclusion

L'apprentissage profond est un domaine intéressant, où il est largement utilisé par les grandes entreprises et les grandes firmes pour avoir des bonnes résultats, et pour avoir des solutions aux problèmes complèxes comme la création des véhicules autonomes, reconnaissance faciale...etc.

L'apprentissage profond est un peu complèxe et il a besoin d'une grande masse de donnée, et des machines d'haute performance pour faire les calcules dans les meilleurs délais, comme les clusters ou l'utilisation de Cloud qui un peu cher.

Chapitre 3

Conception de système

3.1 Introduction

Dans ce chapitre on va expliqué les étapes et les modules composant notre système, où nous présentons la conception de notre système en commençant par sa conception générale puis sa conception détaillée en expliquant les différents éléments du système et précisant leur fonctionnement.

3.2 Méthodologie suivie

Pour réaliser à 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 méthode de réseau récurrent à mémoire court et long terme, en anglais, Long Short-Term Memory Networks(LSTM).

Cette technique, à besoin d'un grand corpus marqué(chaque donnée est attribuée à sa classe), et besoin d'une technique pour rendre ce corpus compréhensible pour la machine. Et pour cela, nous avons ramassé des commentaires à partir des status des pages Facebook Algériennes populaires, ensuite nous avons marqué ces commentaires en commentaires de sentiments positifs ou négatifs. Et nous avons aussi ramassé un grand corpus avec une taille qui dépasse un gigaoctet de texte qui est écrit soit en Arabe Algérien ou en Arabe standard pour entraîner notre Word2vec.

3.3 Conception globale du système

Globalement, on peut représente l'architecture de notre système de catégorisation des sentiments d'un texte comme suit :

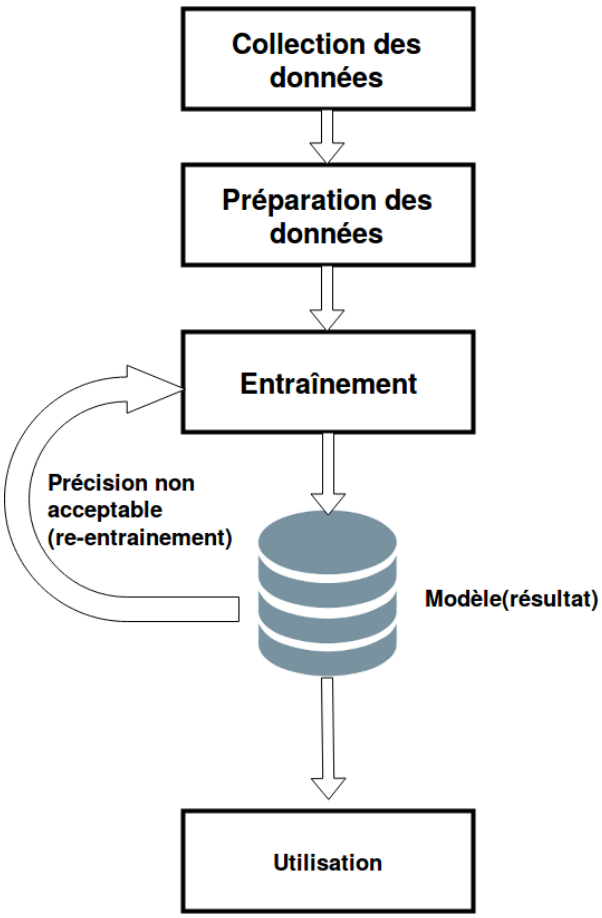


FIGURE 3.1: L'architecture générale du système

Comme il est montré dans l'architecture précédente, on peut deviser notre système en trois grandes modules.

3.3.1 Collection des données

Dans ce module on ramasse deux corpus de données, le premier qui est ramassé à partir des contenues des pages web écrites en Arabe pour entraîner le Word2vec qui va nous aider à plonger les mots de corpus d'entraînement. Le deuxième corpus de données pour entraîner notre modèle de catégorisation des sentiments, qui est un corpus ramassé à partir des pages Facebook Algériennes populaires.

La relation entre le modèle de Word2vec et le modèle de catégorisation des sentiments, est qu'à l'aide de modèle de word2vec on peut rendre dataset (l'entrée d'entrainement de modèle de catégorisation des sentiments) en vecteur des réelles.

3.3.2 Préparation des données

Dans ce module , on prétraite les deux corpus par la suppression de tous les caractères non Arabe, et aussi par la suppression de "Tashkil" (la ponctuation), ensuite on supprime le vide généré après la suppression de certains caractères non Arabe, et après on supprime les

caractères dupliqués.

Et aussi, on marque les données avec un label positif ou négatif ensuite on les sauvegarde dans un fichier Excel en format CSV.

3.3.3 Entraînement

Dans ce module, on va entraîner notre système qui va essayer d'apprendre et de créer un modèle de catégorisation des sentiments à partir les données marquées, et ensuite il va sauvegarder le modèle comme montré dans la figure 3.8, et si le modèle a une bonne précision on l'occupe sinon on refait l'entraînement avec d'autres paramètres.

3.4 Conception détaillé du système

Dans cette partie, on va présenter séparément chaque partie du système proposé en détaillant le principe du travail de chaque partie.

3.4.1 Collection des données

3.4.1.1 Collection de corpus de Word2vec

Word2vec est une méthode pour représenter un mot en vecteur, donc on a ramassé un grand corpus pour atteindre le maximum des mots existent dans la langue Arabe standard ou vernaculaire, et pour cela on a développé un web crawler, ce dernier est un robot qui aide à explorer les sites web, et nous permet de collecter le contenue de ces pages comme montré dans la figure 3.2, on donne au robot un domaine spécifié comme : "www.echoroukonline.com/" dans le but de ne pas visiter un lien hors ce domaine comme : "www.google.com/", et aussi on le donne des liens pour le démarrage comme : "www.echoroukonline.com/sport/", "www.echoroukonline.com/world/", ... etc. Et ces liens s'appelent les **graines**, et le crawler va visiter les graines et liens existent dans ces graines, en visitant chaque lien uns seules fois.

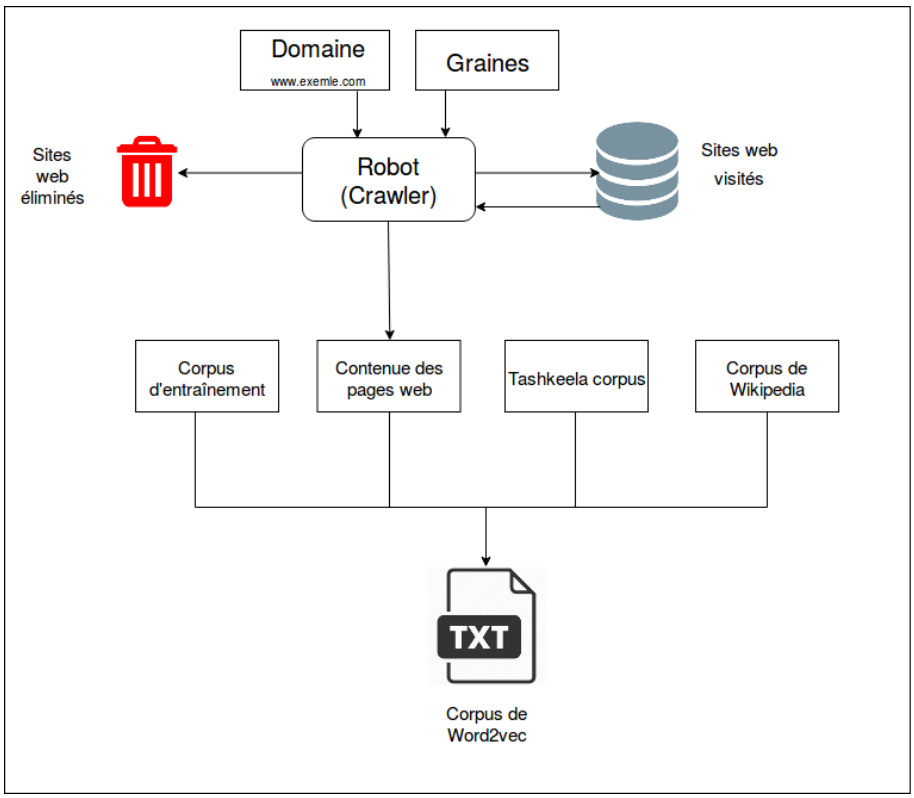


FIGURE 3.2: Système de collection du corpus Wor2vec

Et aussi on a utilisé le corpus d'entraînment qu'on a ramassé à l'aide de l'API de Facebook après le combiner dans seule fichier en format Txt pour assurer que le modèle de Word2vec va s'entraîner aussi sur les mots existe dans le corpus d'entraînment, en plus en a utilisé le contenue de site web de Wikipédia qu'on a trouvé sur le site de "archive.org", et aussi on a utilisé le corpus de certains chercheurs dans le domaine comme : le corpus "tashkeela" [1] [55]. Et finalement, on les combine dans un seul fichier en format Txt.

3.4.1.2 Collection de corpus d'entraînement

Pour le corpus d'entraînement on va donner à notre collecteur des données l'Access Token [2], l'identificateur de la page qui on peut le trouver à l'aide d'un site [3] qui on le donne le lien de la page et puis il nous donne l'identificateur de cette page comme montré dans la figure 3.4 , en plus on donne au code l'identificateur de statut qui on le trouve dans le lien de ce statut et comme montré dans la figure 3.5, et le code va amener tous les commentaires de ce statut, et va les sauvegarder initialement dans des fichiers textes à raison de faciliter la manipulation mais aprés on va les sauvegarder dans un seule fichier Csv pour le donner comme entrée à l'algorithme d'entraînement.

On peut conclure ces étapes dans la figure suivante :

1. Un corpus des livres en Arabe, et contenue des pages web, disponible en : https ://source-forge.net/projects/tashkeela/
2. Access Token contient les informations d'identification de sécurité pour une session de connexion et identifie l'utilisateur
3. https ://findmyfbid.com

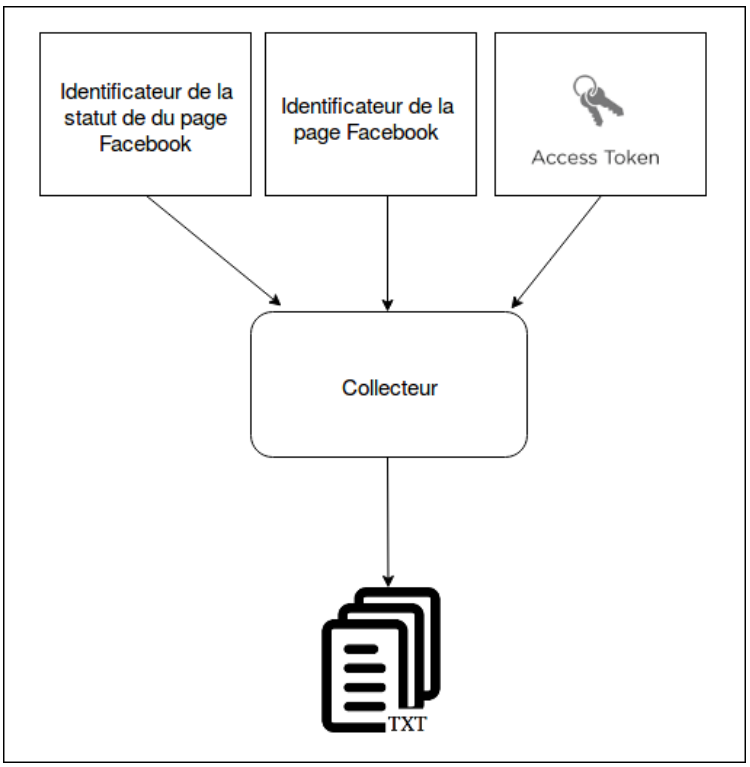


FIGURE 3.3: Système du collection de corpus d'entraînement

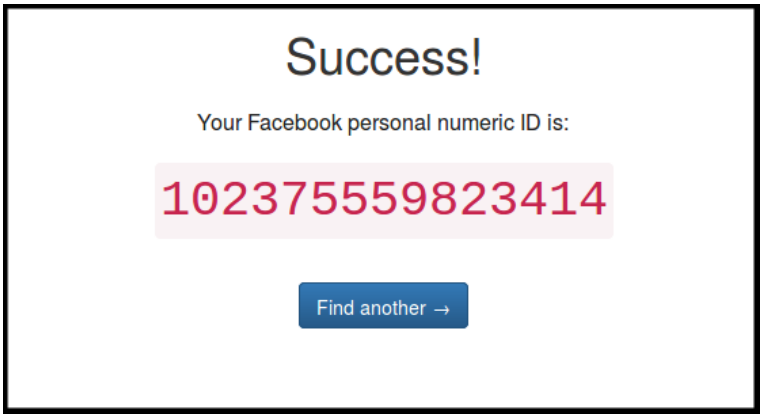


FIGURE 3.4: Résultat de site après lui donner le lien de la page Facebook Elbilad

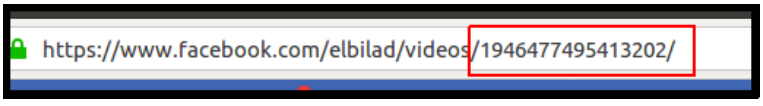


FIGURE 3.5: Exemple d'un identificateur d'un statut Facebook

3.4.2 Préparation des données

3.4.2.1 Prétraitement des données

Pour le prétraitement des données, on supprime tous les caractères non Arabe et les caractères spéciaux, ensuite, on supprime les caractères dupliqués comme la duplication de caractère ا dans le mot شكرااااااا et les ponctuations "Tashkeel" comme la phrase اَلسَّلَامُ عَلَيْكُمْ sera السلام عليكم et on supprime les vides générés après la suppression des caracteres non Arabes, en plus, on replace أ, آ, إ par ا et ة par ه et ى par ي, car la majorité des internautes mélange entre ces caractères, entre أ, آ, إ et ا, et entre ة et ه, et entre ى et ي .

Ce prétraitement est dans le but de normaliser les mots car la machine ne comprend pas que le mot مَرْحَبًا et مرحبا sont les mêmes mots, ou أهلا et اهلا sont les mêmes.

3.4.2.2 Marquage des données

Pour le marquage des données on va déplacer chaque commentaire vers un dossier, où chaque dossier représente un label, et nous on a trois dossier :(négative, positive, ignorer) si le texte contient des sentiments négatifs on le déplace au dossier "négatifs" et s'il contient des sentiments positifs on le déplace au dossier "positifs", et si le texte ne contient ni sentiment positive ni sentiment négative ou si le texte est incompréhensible on le déplace vers le dossier "ignorés". Et tous les textes dans le dossier "ignorés" on va les éliminer après.

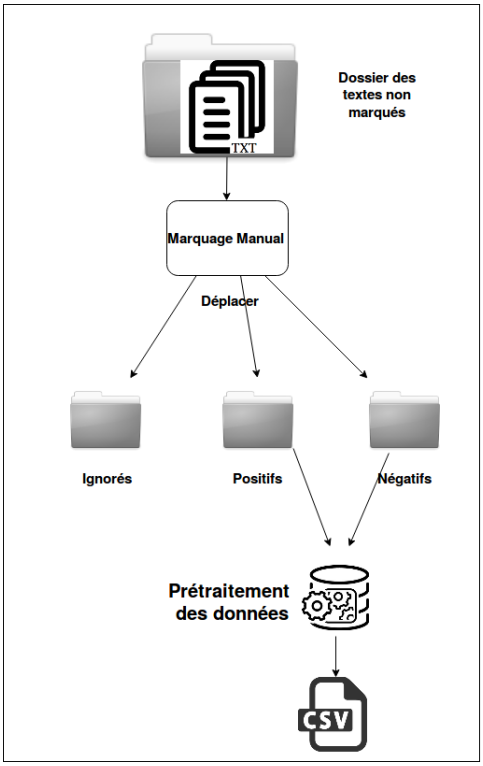


FIGURE 3.6: Système de marquage des données

Par exemple, le commentaire زرت بزاف بلدان لي يعجبوني مبصح بلادنا خيرة البلدانْ on va le classer manuellement en texte de sentiment positif, et ماتحشم ما تخاف ولات ماكاينش الناسْ

مَن ربيْ on va le classer manuellement en texte de sentiment négatif, et تُقع الجزائر في شمال إفريقيا on va le déplacer manuellement dans le dossier des textes ignorés.

3.4.3 Entraînement

3.4.3.1 Entraînement de Word2vec

Pour entraîner un Word2vec on est besoin d'un grand corpus de texte en format Txt et prétraité, et après l'entraînement de Word2vec on obtient trois fichiers, le premier c'est le modèle Word2vec en format 'bin', qu'on va utiliser pour évaluer notre Word2Vec. Le deuxième fichier, c'est des vecteurs des mots, où chaque mot est un vecteur des réelles. Le troisième fichier est une liste des mots qui ont une représentation vectorielle, car ce n'est pas tous les mots existent dans le fichier texte seront être vectoriser, juste les mots qui sont répétés dans le fichier texte avec certains nombres de répétitions qu'on le choisit.

Le deuxième et le troisième fichier sont sauvegardés en format de vecteur Numpy(bibliothèque de Python destinée à manipuler des matrices ou tableaux multidimensionnels) comme montrés dans la figure 3.7. Et ces deux fichiers vont nous aider à représenter notre corpus d'entraînement en matrices des réelles.

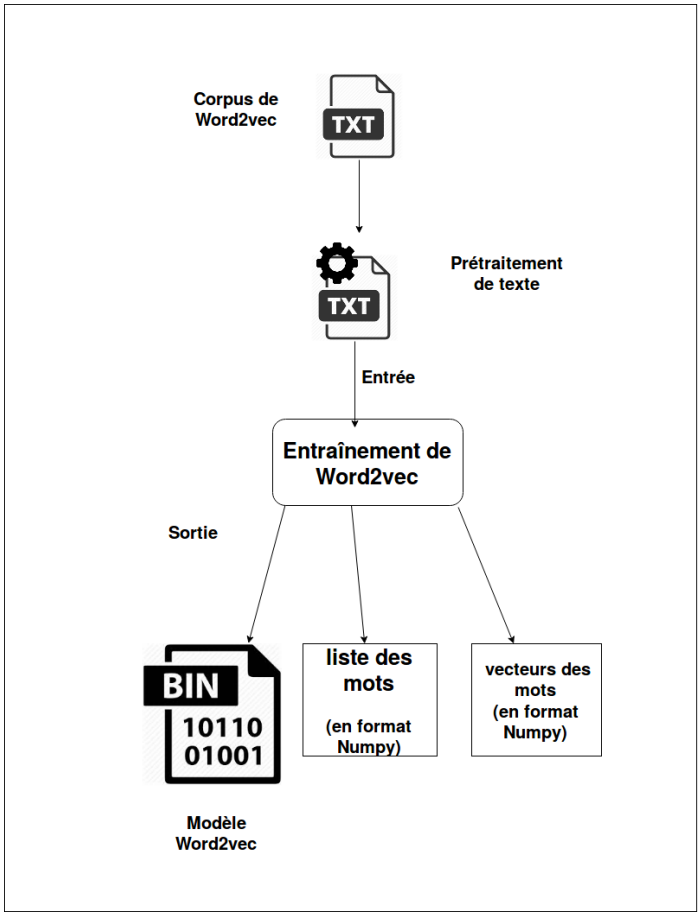


FIGURE 3.7: Entraînement de Word2vec

Le vecteur Numpy de liste des mots contient la liste de tous les mots qui a un vecteur des réelles calculés à l'aide de Word2vec, car ce n'est pas tous les mots existent dans le corpus auront un vecteur de réelle car on a fixé une variable s'appelle "min count" (voir 4.3.1.1), où chaque mot est répété dans le corpus "min count" fois aura un vecteur de réelle. Par exemple : [السلام, مرحبا, ..., عجائ]

Le vecteur des réelles qui est en format Numpy aussi, sa dimension est : [nombre totale des mots * 300], où 300 c'est la taille de vecteur de chaque mot. par exemple : [[0.5, 0.2, ..., 0.25, 0.8], ...,[1.2, 1.6, ..., 1.9]]

3.4.3.2 Entraînement du modèle de catégorisation des sentiments

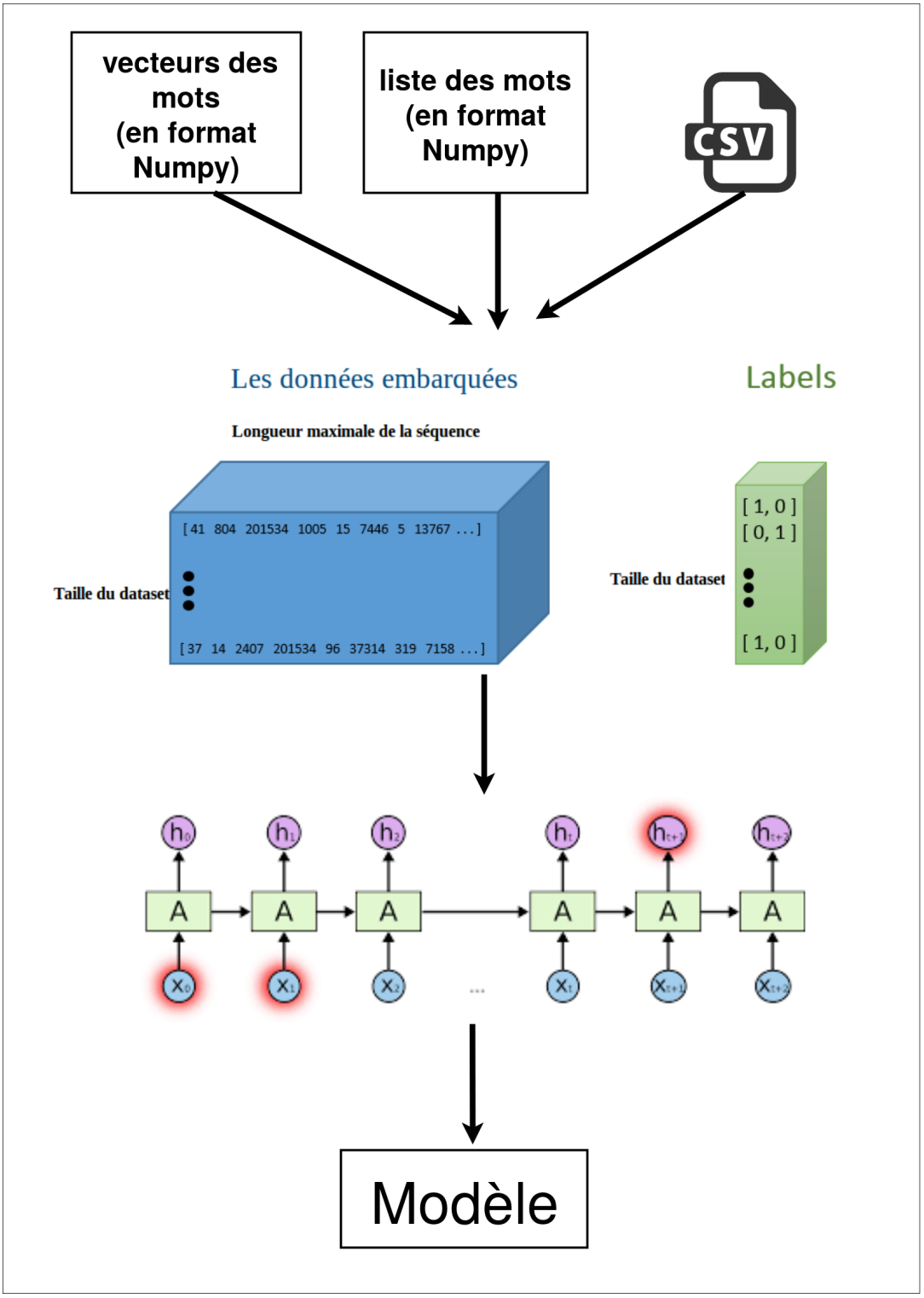


FIGURE 3.8: Entraînement du Modèle

Pour entraîner le modèle on est besoin tout d'abbord de transformer notre données à l'aide des fichiers Numpy générés dans la phase d'entraînement de Word2vec, 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 300 dimensions, et chaque mot sera traité dans une cellule LSTM(Long short-term memory, en Français : réseau récurrent à mémoire court et long terme) [4] ou le nombre des mots est limité par variable de longueur maximale. Et après la fin d'entraînement on va sauvegarder le modèle si sa précision est acceptable, pour l'utiliser après, 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) avec un nombre d'échantillons, car on ne peut pas transmettre tout le dataset dans un réseau de neurones en même temps, et ce lot sera transmis avec ces lables(l'étiquette de chaque item : positif ou négatif) comme montré dans l'algorithme au-dessus.

Algorithm 1 Algorithme d'entraînement du modèle de catégorisation des sentiments

```
matriceIds ← EmbarqueDonnees(dataset, vecteurMots,vecteurRéelles)
for i <iterations do
    lot, labels ← avoirLot(matriceIds)
    entrainerModele(lot,labels, nombreLSTM,longeurMax)
sauvegarderModele()
```

FIGURE 3.9: Algorithme d'entraînement du modèle de catégorisation des sentiments

3.4.3.3 Teste 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 75 %) on occupe le modèle sinon, on refait le traitement avec d'autres hyperparamètres, où on essaie de perfectionner notre Word2vec par le re-entraîner, et aussi de modifier les hyperparamètres.

3.4.3.4 Utilisation du modèle

On peut utiliser notre modèle sauvegardé, avec la manière d'utilisation est comme suit :

4. des cellules de notre réseau de neurones(voir le 2ème chapitre)

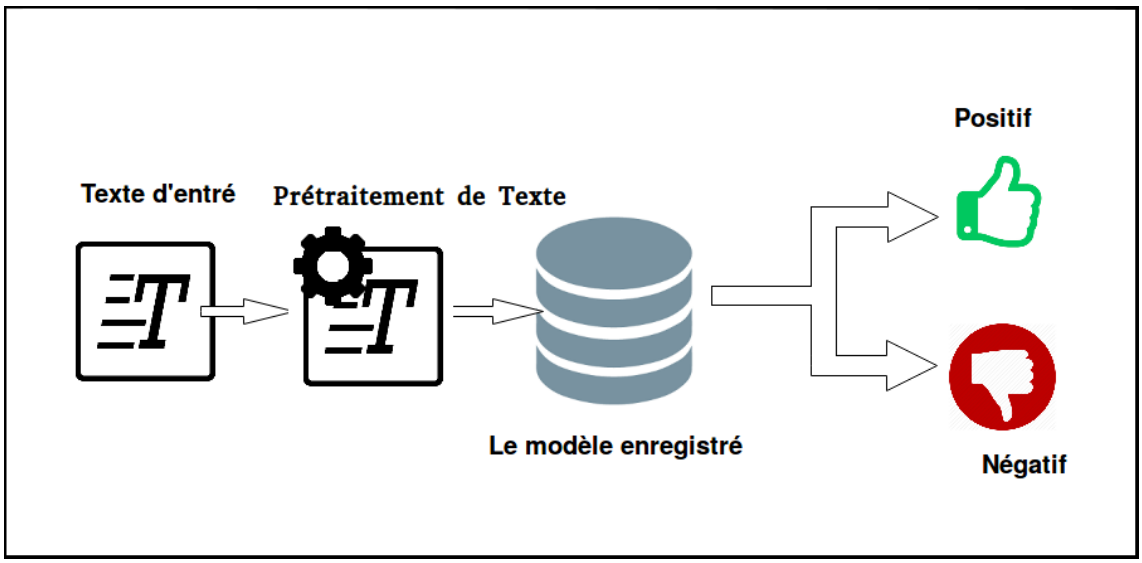


FIGURE 3.10: Utilisation du modèle

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

3.4.4 Conclusion

Dans ce chapitre, on a présenté notre méthode proposée, où on a présenté la conception de notre système, et on a détaillé les étapes qu'on a passé pour arriver à notre système, et 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.

Chapitre 4

Implémentation

4.1 Introduction

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

Pour développer notre système et valider notre proposition, nous avons utilisé le langage de programmation Python et l'environnement Pycharm pour écrire les programmes. Pour la collection des données nous avons utilisé l'API de Facebook pour ramasser les commentaires, et la bibliothèque Beautiful Soup pour collecter des pages web Arabes et parser le langage HTML et d'extraire le contenu de ces pages.

Pour l'apprentissage profond nous avons utilisé la bibliothèque de Google qui s'appelle TensorFlow. Et pour l'entraînement de Word2vec nous avons utilisé la bibliothèque de Gensim de RaRe Technologies qui est basée sur la bibliothèque word2vec de Google, 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 utiliser pour la programmation générale. Créé par Guido van Rossum et sorti en 1991, Python a une philosophie de conception qui met l'accent sur la lisibilité du code, notamment en utilisant des espaces importants. Il fournit des constructions qui permettent une programmation claire à petite et à grande échelle. Python dispose d'un système de type dynamique et d'une gestion automatique de la mémoire. Il prend en charge de multiples paradigmes de programmation, y compris orientée objet, impératif, fonctionnel et procédural, et dispose d'une bibliothèque standard vaste et complète.

Les interpréteurs Python sont disponibles pour de nombreux systèmes d'exploitation. CPython, l'implémentation de référence de Python, est un logiciel open source et possède un modèle de développement basé sur la communauté, comme presque toutes ses implémentations de variantes. CPython est géré par la fondation Python Software à but non lucratif [7].

4.2.1.2 PyQt

FIGURE 4.2: PyQt logo

PyQt est une extension de la boîte à outils graphique **Qt** qui est multiplateforme. PyQt est un logiciel libre développé par la firme britannique Riverbank Computing. Il est disponible sous des termes similaires aux versions de Qt antérieures à 4.5, cela signifie une variété de licences, y compris la licence publique générale GNU (GPL) et la licence commerciale. PyQt prend en charge Microsoft Windows ainsi que diverses versions d'UNIX, y compris Linux et MacOS.[9]

4.2.1.3 PyCharm

FIGURE 4.3: PyCharm logo

PyCharm est un environnement de développement intégré (IDE) utilisé dans la programmation informatique, spécifiquement pour le langage Python. Il est développé par la société tchèque JetBrains.Il fournit une analyse de code, un débogueur graphique, un testeur d'unité intégrée, l'intégration avec des systèmes de contrôle de version (VCS), et prend en charge le développement web avec Django.PyCharm est multi-plateforme, avec les versions Windows, macOS et Linux. L'édition de communauté est libérée sous la licence d'Apache, et il y a également l'édition professionnelle libérée sous une licence de propriétaire [6].

4.2.2 Jupyter Notebook

FIGURE 4.4: Jupyter Notebook logo

Jupyter Notebook est une application Web open-source qui vous permet de créer et de partager des documents contenant du code en direct, des équations, des visualisations et du texte narratif. Les utilisations incluent : le nettoyage et la transformation des données, la simulation numérique, la modélisation statistique, la visualisation des données, l'apprentissage automatique et bien plus encore [3].

4.2.3 Les outils utilisés

4.2.3.1 Facebook API

L'API Facebook est une plate-forme pour créer des applications qui sont disponibles pour les membres du réseau social de Facebook. L'API permet aux applications d'utiliser les connexions sociales et les informations de profil pour rendre les applications plus impliquées et pour publier des activités sur le fil d'actualités et les pages de profil de Facebook, sous réserve des paramètres de confidentialité des utilisateurs individuels. Grâce à l'API, les utilisateurs peuvent ajouter un

contexte social à leurs applications en utilisant des données de profil, d'ami, de page, de groupe, de photo et d'événement. L'API utilise le protocole RESTful et les réponses sont au format JSON [5].

4.2.3.2 Beautiful Soup

Beautiful Soup est une bibliothèque de parsage pour le langage HTML écrite en Python par Leonard Richardson.Cette bibliothèque logicielle peut aussi être utilisée pour traiter du XML. La bibliothèque Beautiful Soup permet de naviguer au sein de l'arbre créé par le parser, de chercher des éléments dans cet arbre ou les modifier [10].

4.2.3.3 TensorFlow

FIGURE 4.5: 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 [8].

4.2.3.4 Gensim

Gensim est un outil robuste de modélisation de l'espace vectoriel open-source et de modélisation de sujet implémenté en Python. Il utilise NumPy, SciPy et éventuellement Cython pour les performances. Gensim est spécialement conçu pour gérer de grandes collections de textes, en utilisant le streaming de données et des algorithmes incrémentaux efficaces, ce qui le différencie de la plupart des autres logiciels scientifiques qui ne ciblent que le traitement par lot et en mémoire [39].

4.2.3.5 Numpy

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].

4.3 Système de catégorisation des sentiments

4.3.1 Ensemble des données utilisés

4.3.1.1 Pour l'entraînement de Word2vec

Pour l'entraînement de Word2vec nous avons utilisé les pages Wikipédia en Arabe comme corpus, qui sont téléchargé du site web archive.org de taille 500 mégaoctets, et nous avons utilisé aussi notre corpus ramassé de Facebook (presque 100,000 commentaires) à l'aide de l'API de Facebook, et nous avons utilisé aussi notre corpus ramassé depuis les journaux d'actualité Algérienne (Elchorouk, Elkhaber,...etc.) à l'aide de la la bibliothèque "Beatiful Soup"(bs4), qui est une bibliothèque écrite en Python, et nous avons aussi utilisé un corpus trouvé sur internet qui s'appelle Tashkeela (réalisé par Taha Zerrouki et Amar Balla) [55] de taille de plus d'un gigabyte. Donc nous avons un corpus de 1.5 gigaoctets, et après le prétraitement (qu'on a bien détaillé dans le chapitre 3) de corpus on obtient un corpus de taille de 1.4 gigaoctets en totale.

4.3.1.2 Pour l'entraînement de modèle de catégorisation des sentiments

Pour l'entraînement du modèle, nous avons utilisé une dataset de 49864 items (24932 positifs et 24932 négatifs). Où nous avons marqué plus de 15,000 de notre corpus ramassé, et nous avons aussi utilisé deux autres datasets trouvés sur l'internet la première est celle d'Omar Zelamti et al [30], et l'autre c'est d'Hady ElSahar et al [16].

4.3.2 Entraînement et test

4.3.2.1 Word2vec

Pour l'entraînement de Word2vec nous avons utilisé la bibliothèque de Gensim, qui est une bibliothèque écrite en Python, et nous avons choisi certaines hypers paramètres qui sont montrés dans la figure 4.6, comme la dimension de vecteur des mots qu'on a choisi la dimension 300, et nous avons choisi aussi 10 mots comme paramètre des mots qu'on va les prendre en compte, donc prend juste les mots qui ont une occurrence de 10 mots ou plus. Nous avons choisi aussi 9 comme paramètre de "Window", ce paramètre c'est le nombre des mots proches qu'a une relation avec le mot actuel qu'on va calculer son vecteur.

```
import gensim
from gensim.models import KeyedVectors
import numpy as np
from gensim.models import word2vec

sentences = gensim.models.word2vec.LineSentence('Word2Vec_corpus.txt')
model = word2vec.Word2Vec(sentences, size=300,window=9,min_count=10)
```

FIGURE 4.6: Une partie de code du word2vec

Et nous avons testé notre Word2vec sur quelques mots, par la vérification de similarité des mots, comme montré dans les exemples suivants à l'aide de Jupyter Notebook.

```
In [4]: model.most_similar(positive='ممتاز', topn=10)
Out[4]: [('0.6881198883056641 ,'رائع'),
         ('0.6755329370498657 ,'جيد'),
         ('0.6232588887214661 ,'راقي'),
         ('0.578782320022583 ,'لذيذ'),
         ('0.5537036061286926 ,'متميز'),
         ('0.5481072664260864 ,'ونظيف'),
         ('0.5469245910644531 ,'سئ'),
         ('0.5399249792098999 ,'ممتازه'),
         ('0.5343469977378845 ,'جميل'),
         ('0.5266199707984924 ,'مذهل')]
```

FIGURE 4.7: Teste du word2vec exemple 1

Dans le premier exemple, nous avons testé notre Word2vec sur le mot ممتاز (excellent en Français), et le modèle de Word2vec nous a montré ces similaires lesquelles : رائع (super en Français) avec une probabilité du similarité de 0.69 qui est la plus élevé, qui signfie que le mot رائع et la plus proche au mot ممتاز dans notre modèle de Word2vec, راقي (sophistiqué en Français) avec une probabilité du similarité de 0.62 ...etc, qui sont synonymes et proches en terme du sens.

```
In [7]: model.most_similar(positive='باريس', topn=10)
Out[7]: [('0.6657314896583557 ,'فيينا'),
         ('0.6540641784667969 ,'لندن'),
         ('0.6491469144821167 ,'برلين'),
         ('0.6453872323036194 ,'بروكسل'),
         ('0.6443612575531006 ,'ستراسبورغ'),
         ('0.6425604224205017 ,'زيورخ'),
         ('0.6334376335144043 ,'فرانكفورت'),
         ('0.6319502592086792 ,'تولوز'),
         ('0.6260506510734558 ,'فرنسا'),
         ('0.6212066411972046 ,'بباريس')]
```

FIGURE 4.8: Teste du word2vec exemple 2

Dans le deuxième exemple, nous avons testé notre Word2vec sur le mot باريس (Paris en Français), et le modèle de Word2vec nous a montré ces similaires lesquelles : فيينا (Vienne en Français), لندن (Londres en Français)...etc, qui sont proches en terme du sens, car ces similarités sont des villes et des capiteaux dans l'europe comme Paris .

Et comme on voit dans ces exemples, la précision de notre Word2vec est parfaite, où notre modèle de Word2vec a bien éstimé les mots similaires de mot ممتاز et le mot باريس.

4.3.2.2 Modèle de catégorisation des sentiments

Pour l'entraînement du modèle nous avons utilisé la bibliothèque de TensorFlow et de Numpy qui sont écrits en Python, et nous avons choisi les hypers paramètres comme suit, nombre d'iterations sont égales à 100.000, nombre des cellules LSTM(Long short-term memory, en Français : réseau récurrent à mémoire court et long terme) [1] est 64 celles, la taille de lot [2] (batch en Anglais) est 20, le nombre maximum des mots est 250 mots dans chaque échantillon (commentaire), et nous avons chosisi ces choix car ils sont utilisé dans plusieurs recherches et donnent des bons résultats. Et après nous avons lancé l'exécution de code, après la fin d'exécution, on teste la précision du modèle à l'aide de la bibliothèque si elle est bonne on occupe le modèle, sinon on fait des modifications sur les hypers paramètres ou nous ajoutons plus de données au dataset.

Et nous avons testé notre modèle sur quelques commentaires, comme montré dans les exemples suivants à l'aide de Jupyter Notebook.

```
In [9]: s = "الكتاب هذا عجبني بزاف، حاجة مليحة"
        predict(s)

['الكتاب', 'هذا', 'عجبني', 'بزاف', 'حاجه', 'مليحه']
['مليحه', 'حاجه', 'بزاف', 'عجبنى', 'هذا', 'الكتاب']


Le texte contient des sentiments positifs
```

FIGURE 4.9: Teste du modèle de catégorisation des sentiments, exemple 1

Dans le premier exemple, nous avons testé notre modèle sur le texte الكتاب هذا عجبني بزاف، حاجة مليحة (en Français : ce livre me plaît beaucoup, quelque chose de bien) et le modèle nous a dit que ce texte est positif, et c'est vrai, car le texte contient des sentiments positifs, où le rédacteur de ce commentaire a aimé le livre.

```
In [10]: s = "الفيلم لي قتلي عليه ماعجبنيش تعبني في باطل، ندمت علاه شفتو"
         predict(s)

['الفيلم', 'لي', 'قتلي', 'عليه', 'ماعجبنيش', 'تعبني', 'في', 'باطل', 'ندمت', 'علاه', 'شفتو']
['شفتو', 'علاه', 'ندمت', 'باطل', 'في', 'تعبني', 'ماعجبنيش', 'عليه', 'قتلي', 'لي', 'الفيلم']


Le texte contient des sentiments négatifs
```

FIGURE 4.10: Teste du modèle de catégorisation des sentiments, exemple 2

Dans le deuxième exemple, nous avons testé notre modèle sur le texte الفيلم لي قتلي عليه ماعجبنيش تعبني في باطل، ندمت علاه شفتو (en Français : le film dont tu m'as

1. des cellules de notre réseau de neurones(voir le 2ème chapitre)
2. un groupe d'échantillons d'entraînement, car on ne peut pas transmettre tout le dataset dans un réseau de neurones en même temps. Donc, on divise l'ensemble de données en lots ou ensembles ou parties.

parlé ne m'a pas plu mais m'a fatigué, j'ai regretté pourquoi je l'ai vu.) et le modèle nous a dit que ce texte est négatif, et c'est vrai, car le texte contient des sentiments négatifs, où le rédacteur de ce commentaire n'a pas aimé le film.

4.3.3 Présentaion des interfaces

Dans cette section, nous avons présenté les interfaces graphiques de notre système, où ces interfaces graphiques sont créées à l'aide de la bibliothèque PyQt qui est écrite en Python.

4.3.3.1 Interface de collection des données

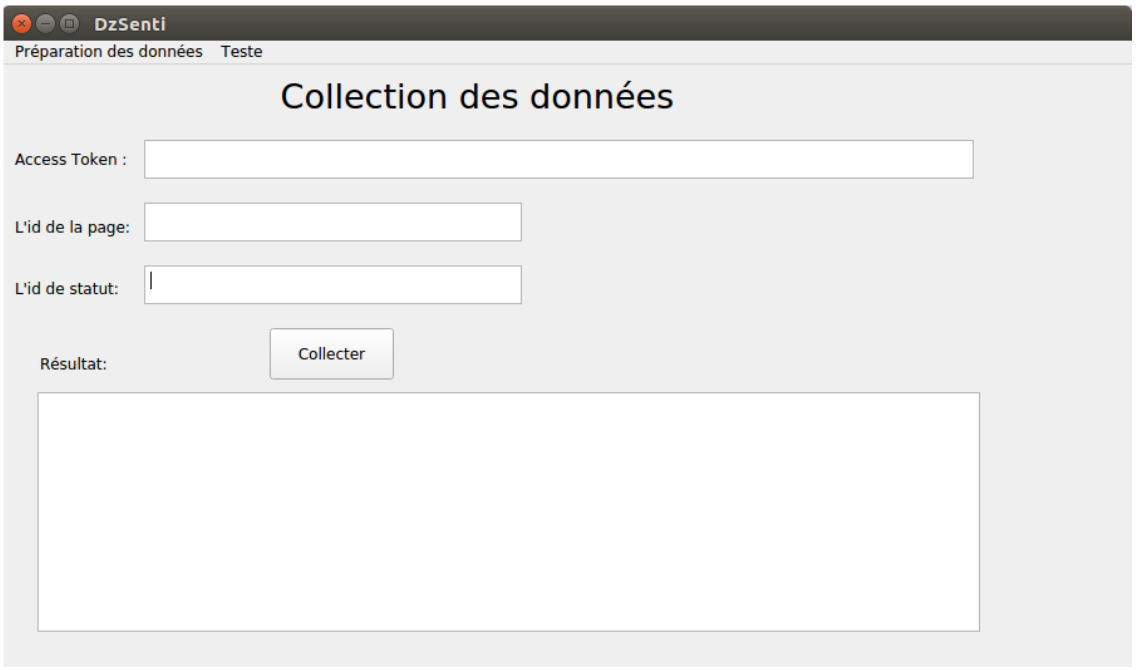


FIGURE 4.11: Interface de collection des données

Cette interface permet de ramasser des commentaires à partir d'une statut du page Facebook, à l'aide de :

- L'access token de compte Facebook.
- Identificateur de la page .
- Identificateur du statut.

Où chaque commentaire va être stocké dans un fichier unique pour faciliter la manipulation et la gestions des commentaires.

4.3.3.2 Interface de marquage des données

FIGURE 4.12: Interface de marquage des données

Cette interface permet de marquer les commentaires manuellement soit : positifs, négatifs ou l'ignorer. Chaque commentaire marqué on va changer son répertoire selon le button clicker, si le button clické est positif le commentaire (fichier) va être déplacé vers le dossier positif, et le même pour les autres buttons.

4.3.3.3 Interface de teste du modèle de catégorisation des sentiments

FIGURE 4.13: Interface de teste du modèle de catégorisation des sentiments

Cette interface permet de tester le modèle de catégorisation des sentiments sur un texte, et on voit si le texte est de sentiment positif ou sentiment négatif.

4.3.3.4 Interface de teste du Word2vec

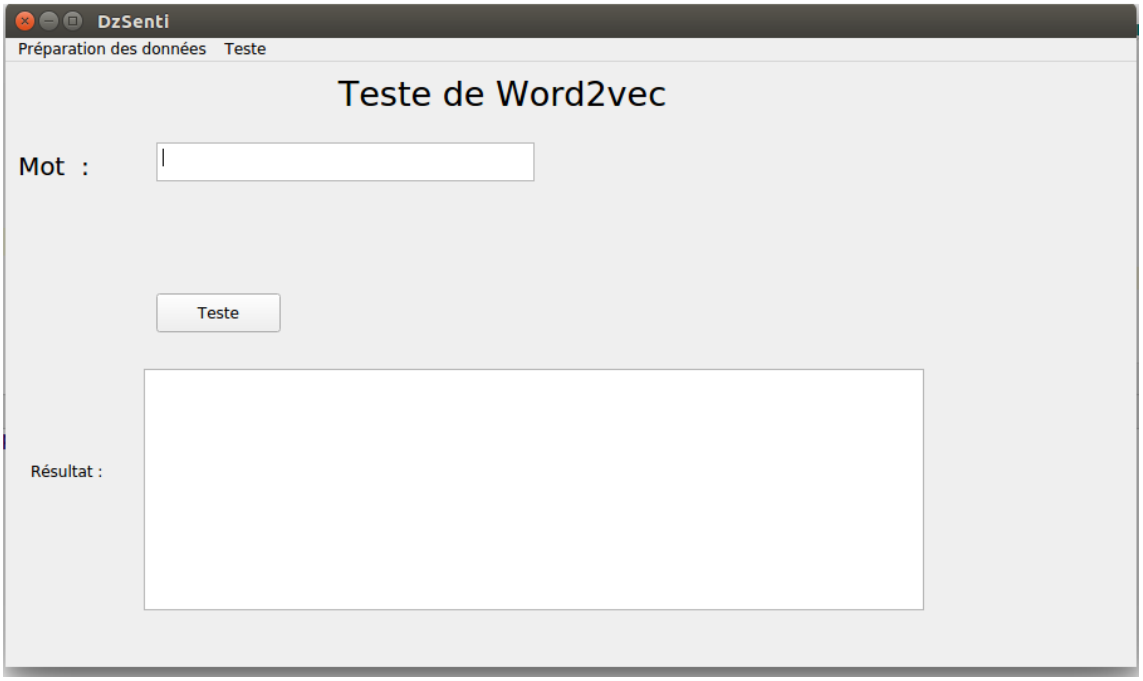


FIGURE 4.14: Interface de teste du Word2vec

Cette interface permet de tester le Word2vec, avec le teste de similarité de certaines mot et la probabilité de cette similarité, si on écrit un mot et ensuite on click sur le button "Teste", les mots similaires du mot entré vont apparaître de la zone de texte.

4.4 Expérimentaions et résultats obtenues

Pour arriver à notre système, nous avons passer par ces expérimentaions :

4.4.1 Première expérimentaion

Dans la première expérimentaion nous avons entraîné notre modèle de catégorisation des sentiments sur une dataset de 25000 items, 12500 positifs et 12500 négatifs,et nous avons divisé le corpus en 2000 commentaires de test(qui représente 8% de dataset) pour tester notre modèle et avoir sa précision, et 23000 commentaires d'entraînement, et dans cette expérimentation nous avons utilisé un Word2vec de Abu Bakr Soliman Mohammad et al (AraVec) [42], et nous avons obtenue comme précision 63%.

Précision	Taille de dataset	Données de teste	Word2vec
62%	25000	8%	AraVec

TABLE 4.1: Résultat de première expérimentation

4.4.2 Deuxième expérimentation

Dans la deuxième expérimentaion nous avons entraîné notre modèle de catégorisation des sentiments sur une dataset de 25000 items, 12500 positifs et 12500 négatifs,et nous avons divisé le corpus en 2000 commentaires de test(qui représente 8% de dataset), et 23000 commentaires d'entraînement, et dans cette expérimentation nous avons utilisé notre Word2vec qui a été entraîné sur un corpus de 600 mégaoctets, et nous avons obtenue comme précision 60%.

Précision	Taille de dataset	Données de teste	Word2vec
60%	25000	8%	Notre word2vec

TABLE 4.2: Résultat de deuxième expérimentation

4.4.3 Troisième expérimentation

Dans la troisième expérimentaion nous avons entraîné notre modèle de catégorisation des sentiments sur une dataset de 25000 items, 12500 positifs et 12500 négatifs,et nous avons divisé le corpus en 2000 commentaires de test(8%), et 23000 commentaires d'entraînement, et dans cette expérimentation nous avons utilisé notre Word2vec qui a été entraîné sur un corpus de 1.4 gigaoctets, et nous avons obtenue comme précision 63%.

Précision	Taille de dataset	Données de teste	Word2vec
63%	25000	8%	Notre word2vec

TABLE 4.3: Résultat de troisième expérimentation

4.4.4 Quatrième expérimentation

Dans la quatrième expérimentaion nous avons entraîné notre modèle de catégorisation des sentiments sur une dataset de 28649 items, 13690 positifs et 14959 négatifs,et nous avons divisé le corpus en 2292 commentaires de test (8%), et 26357 commentaires d'entraînement, et dans cette expérimentation nous avons utilisé notre Word2vec qui a été entraîné sur un corpus de 1.4 gigaoctets, et nous avons obtenue comme précision 75%.

Précision	Taille de dataset	Données de teste	Word2vec
75%	28649	8%	Notre word2vec

TABLE 4.4: Résultat de quatrième expérimentation

4.4.5 Cinquième expérimentation

Dans la cinquième expérimentaion nous avons entraîné notre modèle de catégorisation des sentiments sur une dataset de 49864 items, 24932 positifs et 24932 négatifs,et nous avons divisé le corpus en 7480 commentaires de test (15%), et 42384 commentaires d'entraînement, et dans cette expérimentation nous avons utilisé notre Word2vec qui a été entraîné sur un corpus de 1.4 gigaoctets, et nous avons obtenue comme précision 81%.

Précision	Taille de dataset	Données de teste	Word2vec
81%	28649	15%	Notre word2vec

TABLE 4.5: Résultat de cinquième expérimentation

4.5 Discussion des résultats et comparison

Les résultats obtenus dans les différentes expérimentations montrent l'efficacité de notre proposition et la possibilité de son utilisation dans ce domaine. En effet, notre modèle de catégorisation des sentiments, à permet de catégoriser correctement plus de 81% des données de test de totalité de 7480 items, ça veut dire que notre système a bien catégorisé plus de 5984 items.

Ainsi que, les expérimentations prouvent qu'avec l'augmentation de la taille des données d'entraînement le taux de précision augmente, aussi le pourcentage de dataset de test est important, comme dans les premières expérimentations nous avons utilisé un pourcentage faible 8% la précision aussi étée faible, mais avec le pourcentage 15% qui est un pourcentage standard utiliser dans la majorité des recherches, nous avons eu une meilleure précision.

Et aussi, le Word2vec influence sur la précision, où nous avons obtenue une précision dans la première expérimentation mieux que la deuxième, car dans la deuxième nous avons utilisé notre Word2vec qui est entraîné sur un corpus faible de 600 mégaoctets et de la première est entraîné sur un grand corpus.

Comparé à la méthode existe de M'hamed Metaoui et al[30], notre système a réussi à accrocher une précision plus que la méthode de M'hamed Metaoui et al qui est de 76,68%.

4.6 Conclusion

Dans ce chapitre, nous avons représenté l'implémentation de notre système, où nous avons montré l'environnement et les outils de développement qu'on a utilisé. Ensuite nous avons expliqué l'entraînement de word2vec et du modèle de catégorisation des sentiments et les paramètres utilisé et le test, et aussi nous avons montré l'utilisation de notre modèle, comme nous avons aussi présenté les interfaces graphiques de notre système. Finalement nous avons expliqué les expérimentations et les résultats obtenues.

Conclusion générale

Le web est devenu une plateforme de lecture-écriture où les utilisateurs ne sont plus strictement des consommateurs d'informations mais aussi des producteurs. Le contenu généré par l'utilisateur, sous forme de texte libre non structuré, devient partie intégrante du web principalement en raison de l'augmentation spectaculaire des sites de réseaux sociaux, des sites de partage de vidéos, des nouvelles en ligne, des sites de critiques en ligne, des forums en ligne et des blogs. En raison de cette prolifération de contenu généré par les utilisateurs, l'exploration de contenu web suscite une attention considérable en raison de son importance pour de nombreuses entreprises, agences gouvernementales et institutions, où l'analyse des sentiments est un sous-domaine important de l'exploration de contenu web.

Pour effectuer l'analyse des sentiments sur un texte Arabe Algérien, 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 mémoire court et long terme(LSTM : Long short-term memory, en Anglais) qui est l'une des architectures de réseaux de neurones récurrents. Dans cette méthode, un dataset de presque 50000 éléments est générée, où nous avons entrainé notre modèle de catégorisation des sentiments sur cette dataset et on a atteint une précision de 81%, qui est un résultat excellent dans ce domaine, et qui nous encourage à améliorer nos recherches de plus en plus.

Pour les perspectives et les travaux de futur, nous proposons des idées qui peuvent améliorer et généraliser notre système de catégorisation des sentiments, telles que :

- Détection et élimination des commentaires sarcastiques pour ne pas perturber notre système.
- Détection et élimination des commentaires de spam car ils contiennent des textes objectifs avec aucune opinion.
- Ajouter une autre étiquette "neutre" pour les commentaires neutre.
- Marquer les commentaires avec certaine dégrée de polarité comme(+1, +2, ..., +5 / -1, -2, ..., -5).

Bibliographie

[1] *Dictionnaire Larousse*.

[2] *Dictionnaire Larousse*.

[3] Le site web officiel de jupyter. `http://jupyter.org/`. consulté le 26/04/2018.

[4] merriam-webster, definition of feelin,. `https://www.merriam-webster.com/dictionary/feeling`. consulté le 10/12/2017.

[5] Site web de facebook developers. `www.developers.facebook.com`. consulté le 26/04/2018.

[6] Site web de la société jetbrain. `www.jetbrains.com`. consulté le 25/04/2018.

[7] Site web de la société non lucratif python software foundation. `www.python.org`. consulté le 25/04/2018.

[8] Site web de tensorflow. `www.tensorflow.org`. consulté le 26/04/2018.

[9] Site web officiel de pyqt. `https://www.riverbankcomputing.com`. consulté le 10/05/2018.

[10] Wikipedia, beautiful soup (html parser). `https://en.wikipedia.org/wiki/Beautiful_Soup_(HTML_parser)`. consulté le 26/04/2018.

[11] Asa Ben-Hur, David Horn, Hava T Siegelmann, and Vladimir Vapnik. Support vector clustering. *Journal of machine learning research*, 2(Dec) :125–137, 2001.

[12] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds. ; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3) :542–542, 2009.

[13] T Edward Damer. *Attacking faulty reasoning*. Cengage Learning, 2008.

[14] Sanjiv Das and Mike Chen. Yahoo ! for amazon : Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific finance association annual conference (APFA)*, volume 35, page 43. Bangkok, Thailand, 2001.

[15] Kushal Dave, Steve Lawrence, and David M Pennock. Mining the peanut gallery : Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM, 2003.

[16] Hady ElSahar and Samhaa R El-Beltagy. Building large arabic multi-domain resources for sentiment analysis. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 23–34. Springer, 2015.

[17] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. Learning precise timing with lstm recurrent networks. *Journal of machine learning research*, 3(Aug) :115–143, 2002.

[18] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[19] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. arxiv. org, 2014.

[20] Vasileios Hatzivassiloglou and Kathleen R McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the european chapter of the association for computational linguistics*, pages 174–181. Association for Computational Linguistics, 1997.

[21] Vasileios Hatzivassiloglou and Janyce M Wiebe. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 299–305. Association for Computational Linguistics, 2000.

[22] Marti A Hearst. Direction-based text interpretation as an information access refinement. *Text-based intelligent systems : Current research and practice in information extraction and retrieval*, pages 257–274, 1992.

[23] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition : The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6) :82–97, 2012.

[24] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8) :1735–1780, 1997.

[25] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[26] Nitin Jindal and Bing Liu. Identifying comparative sentences in text documents. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 244–251. ACM, 2006.

[27] Nitin Jindal and Bing Liu. Mining comparative sentences and relations. In *AAAI*, volume 22, pages 1331–1336, 2006.

[28] Bing Liu. *Web data mining : exploring hyperlinks, contents, and usage data*. Springer Science & Business Media, 2007.

[29] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1) :1–167, 2012.

[30] M'hamed Mataoui, Omar Zelmati, and Madiha Boumechache. A proposed lexicon-based sentiment analysis approach for the vernacular algerian arabic. *Research in Computing Science*, 110 :55–70, 2016.

[31] Karima Meftouh, Nadjette Bouchemal, and Kamel Smaïli. A study of a non-resourced language : an algerian dialect. In *Spoken Language Technologies for Under-Resourced Languages*, 2012.

[32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*, 2013.

[33] Glyn Moody. *Digital code of life : how bioinformatics is revolutionizing science, medicine, and business*. John Wiley & Sons, 2004.

[34] Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. Mining product reputations on the web. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 341–349. ACM, 2002.

[35] Tim Morris. *Computer Vision and Image Processing (Cornerstones of Computing)*. Palgrave Macmillan Limited, 2004.

[36] Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis : Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77. ACM, 2003.

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

[38] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up ? : sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[39] Radim Rehurek and Petr Sojka. Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer, 2010.

[40] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.

[41] Houda Saadane and Nizar Habash. A conventional orthography for algerian arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 69–79, 2015.

[42] Abu Bakr Soliman, Kareem Eissa, and Samhaa R El-Beltagy. Aravec : A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117 :256–265, 2017.

[43] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning :An Introduction*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[44] Richard M Tong. An operational system for detecting and tracking opinions in on-line discussion. In *Working Notes of the ACM SIGIR 2001 Workshop on Operational Text Classification*, volume 1, 2001.

[45] Peter D Turney. Thumbs up or thumbs down ? : semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.

[46] Gary R VandenBos. *APA dictionary of psychology.* American Psychological Association, 2007.

[47] M vanGerven and SM Bohte. Artificial neural networks as models of neural information processing : Editorial on the research topic artificial neural networks as models of neural information processing. 2017.

[48] Wiebke Wagner. Steven bird, ewan klein and edward loper : Natural language processing with python, analyzing text with the natural language toolkit. *Language Resources and Evaluation*, 44(4) :421–424, 2010.

[49] Janyce Wiebe. Learning subjective adjectives from corpora. *Aaai/iaai*, 20(0) :0, 2000.

[50] Janyce M Wiebe. Identifying subjective characters in narrative. In *Proceedings of the 13th conference on Computational linguistics-Volume 2*, pages 401–406. Association for Computational Linguistics, 1990.

[51] Janyce M Wiebe. Tracking point of view in narrative. *Computational Linguistics*, 20(2) :233–287, 1994.

[52] Janyce M Wiebe, Rebecca F Bruce, and Thomas P O'Hara. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 246–253. Association for Computational Linguistics, 1999.

[53] Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Opinionfinder : A system for subjectivity analysis. In *Proceedings of hlt/emnlp on interactive demonstrations*, pages 34–35. Association for Computational Linguistics, 2005.

[54] Theresa Wilson, Janyce Wiebe, and Rebecca Hwa. Just how mad are you ? finding strong and weak opinion clauses. In *aaai*, volume 4, pages 761–769, 2004.

[55] Taha Zerrouki and Amar Balla. Tashkeela : Novel corpus of arabic vocalized texts, data for auto-diacritization systems. *Data in brief*, 11 :147, 2017.