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

## Mémoire

Présenté pour obtenir le diplôme de master académique en

# Informatique

Parcours : **Système d'information optimisation décision (SIOD)**

# Analyse des sentiments arabes en utilisant L'apprentissage en profondeur

**Par :**

**Chenni Rania Wissem**

Soutenu le 2020, devant le jury composé de :

| | | |
|---|---|---|
| **x** | **x** | **Président** |
| **Meadi Med Nadjib** | **x** | **Rapporteur** |
| **x** | **x** | **Examinateur** |

# *Dédicace*

*tout d'abord je remercie Dieu de m'avoir donné la volonté et la force de mener à bien ce travail.*

*Je dédie ce modeste travail*

*À les parents les plus chers au monde,Papa et Maman,que dieu les garde et les protège*

*À mon frère Darradji, ma sœur Hasna*

*À ma petite princesse Rama Itar Al-Nada*

*A ma famille, mes proches Chacun en son nom*

*Mes chers amis, praticuliéremet : Nour, Maria ,Halima*

*à tous les amis*

# *Remerciement*

*Je serais très fier de mes sincères remerciements à mon superviseur le Dr Meadi Mohamed Nadjib pour sa patience, son aide et ses conseils.*

*Merci à tous les professeurs du Département d'informatique l'Université Mohamed Khader Biskra.*

*Merci également au Boudjma Ali pour son soutien.*

*Je remercie également tous les collègues et tous ceux qui ont contribué à ce travail*

# Résumé

Aujourd'hui, l'analyse des sentiments suscite un grand intérêt dans nombreux domaines tels que la politique, les sciences sociales, le marketing et l'économie ... Cela est dû car l'opinion a une grande influence dans plusieurs de ces domaines car elle contribue clairement à la prise de décision.

Dans ce travail, nous visons à réaliser un système d'analyse des sentiments extraites à partir des textes arabes. Contrairement aux techniques utilisées précédemment, nous avons utilisé dans cette étude l'une des techniques d'apprentissage profond pour identifier son potentiel dans ce domaine, ainsi que Bert la nouvelle technique dans le domaine du traitement automatique du langage précisément dans la tâche de représentation des mots.



**Mots clés :** analyse des sentiments, analyse des textes arabe, traitement automatique du langage, l'apprentissage en profondeur, Bert



# Abstract

Today, sentiment analysis is attracting great interest in many fields such as politics, social sciences, marketing and economics ... This is due because opinion has a great influence in many of these areas as it clearly contributes to decision making.

In this work, we aim to realize a system of analysis of feelings extracted from Arabic texts. Unlike the techniques used previously, in this study we used one of the deep learning techniques to identify its potential in this area, as well as Bert the new technique in the area of automatic language processing precisely in the task of representing words.



**Keywords :** sentiment analysis, Arabic text analysis, automatic language processing, deep learning, Bert

# Abréviations

**TALN** Traitement Automatique des Langues Naturelles

**BERT** Bidirectionnal encoder representations from transformers

**CNN** Réseau de neurones convolutif

**RNN** Réseau de neurones récurrents

**LSTM** Long short-term Memory

# Table des matières

0.1 Introduction générale . . . . . 10

1 Analyse de sentiments arabe 12

1.1 Introduction . . . . . 12
1.2 Traitement Automatique des Langues Naturelles(TALN) . . . . . 12
1.2.1 Champs de recherche et applications de TALN . . . . . 13
1.3 Word Embeddings . . . . . 13
1.3.1 TF-IDF . . . . . 14
1.3.2 Word2Vec . . . . . 15
1.3.3 BERT . . . . . 15
1.4 Analyse des sentiments . . . . . 17
1.4.1 Sentiment . . . . . 17
1.4.2 Domaine d'analyse de sentiments . . . . . 18
1.4.3 Les tâches d'analyse des sentiments . . . . . 18
1.4.4 Niveaux d'analyse de sentiments . . . . . 19
1.4.5 Les Méthodes d'analyse sentiments . . . . . 19
1.4.6 Les domaines d'application d'analyse des sentiments . . . . . 20
1.5 Analyse de sentiments en arabe . . . . . 21
1.5.1 COMPLEXITÉ DE LA LANGUE ARABE . . . . . 22
1.5.2 Travaux en relation . . . . . 23
1.6 Conclusion . . . . . 24

2 L'apprentissage en profondeur 25

2.1 Introduction . . . . . 25
2.2 L'apprentissage automatique . . . . . 25
2.2.1 Types d'apprentissage automatique . . . . . 26
2.3 Apprentissage en profondeur . . . . . 27
2.4 Les réseaux de neurones . . . . . 28
2.4.1 Neurone . . . . . 28
2.4.2 Neurone artificiel . . . . . 28
2.5 Les réseaux de neurones . . . . . 29
2.5.1 Architectures réseaux de neurones . . . . . 31
2.6 Long short-term Memory (LSTM) . . . . . 32
2.6.1 Versions de LSTM . . . . . 35
2.7 Conclusion . . . . . 35

3 Réalisation de système 36
3.1 Introduction . . . . . 36
3.2 Conception . . . . . 36
3.2.1 Conception globale du système . . . . . 36
3.2.2 Conception détaillée du système . . . . . 38
3.3 Réalisation de système . . . . . 43
3.3.1 Outils et Environnement de programmation . . . . . 43
3.3.2 Le système développé . . . . . 45
3.3.3 Résultats . . . . . 50
3.4 Conclusion . . . . . 51

**4 Conclusion générale 52**

# Liste des tableaux

1.1 Exemple de différents signification d'un mot . . . . . 22
1.2 Les catégories lexicales seul mot . . . . . 22
1.3 Les Synonymes . . . . . 22
1.4 Exemple de la forme verbale . . . . . 23
1.5 Exemple de stemming arabe . . . . . 23

3.1 Les types de mots vides arabe . . . . . 39
3.2 Exemple de lemmatisation . . . . . 40
3.3 Exemple de prétraitement . . . . . 47
3.4 Les résultats . . . . . 50

# Table des figures


1.1 Les champs de recherche et applications de TALN . . . . . 13
1.2 Les méthodes de calculer le TF . . . . . 14
1.3 Fonctionnement de word2vec . . . . . 15
1.4 Architecture de modèle BERT . . . . . 16
1.5 Mécanisme de classification a l'aide de BERT . . . . . 17
1.6 Exemple d'analyse dans le domaine du marketing . . . . . 20
1.7 Comment la communauté " Jardinerie Animalerie Fleuriste" et fournisseurs réagissent à la 1ère semaine de confinement soumise à l'enquête . . . . . 21

2.1 Les méthodes d'apprentissage supervisé . . . . . 26
2.2 Les méthodes d'apprentissage non supervisé . . . . . 27
2.3 Cellule neuronale . . . . . 28
2.4 Un neurone artificiel . . . . . 29
2.5 Les fonctions d'activation . . . . . 29
2.6 Une couches de réseau de neurones . . . . . 30
2.7 Un réseau de neurones de 3 couches . . . . . 30
2.8 Système de détection des visages avec CNN . . . . . 31
2.9 Une couche RNN . . . . . 32
2.10 Cellule LSTM . . . . . 33
2.11 La porte d'oubli d'une cellule LSTM . . . . . 33
2.12 La porte d'entrée d'une cellule LSTM . . . . . 34
2.13 Mis à jour à l'état de la cellule . . . . . 34
2.14 La porte de sortie d'une cellule LSTM . . . . . 34

3.1 Conception globale du système . . . . . 37
3.2 Conception détaillée du système . . . . . 38
3.3 la structure du mot en langue arabe . . . . . 40
3.4 Les suffixes et préfixes des techniques de la lemmatisation . . . . . 41
3.5 Le procédure d'apprentissage . . . . . 43
3.6 Logo de colab . . . . . 44
3.7 Logo de Python . . . . . 44
3.8 Logo de TensorFlow . . . . . 45
3.9 Logo de Keras . . . . . 45
3.10 Diagramme de nombre d'instances de les classes . . . . . 46
3.11 La réalisation de prétraitement . . . . . 46
3.12 Création des vecteurs . . . . . 47
3.13 Division de données . . . . . 48
3.14 Architecture du modèle . . . . . 48
3.15 Entrainement du modèle . . . . . 49
3.16 Evaluation du modèle . . . . . 49
3.17 La fonction de prédiction . . . . . 49
3.18 Évolution de la précision et du taux d'erreur M-Bert . . . . . 50
3.19 Évolution de la précision et du taux d'erreur Ara-Bert . . . . . 51

## 0.1 Introduction générale

De nos jours, de plus en plus de personnes communiquent et échangent des opinons exprimés sur différentes plate-formes, notamment les réseaux sociaux. Il est devenu nécessaire de prendre en compte ces points de vue pour prendre des décisions sur les sujets abordés dans divers domaines.

Dans ce contexte, on retrouve le domaine de l'analyse des sentiments qui consiste à analyser des opinions exprimées pour extraire des décisions dans des domaines où l'opinion joue un rôle majeur.

Et parce que la pierre angulaire dans le domaine de l'analyse des sentiments est l'opinion qui s'exprime sous nombreuses formes, par exemple, texte, son, image ..., dans notre étude nous appuierons sur la classification des opinions exprimées à travers les textes.

La plupart des techniques utilisées dans ce domaine se heurtaient à des difficultés et à un manque d'efficacité. Pour remédier ces lacunes, une nouvelle technique a été utilisée, représentée par la technologie d'apprentissage en profondeur, qui a montré un grand potentiel dans ce domaine.

Cette technique comprend un large champ de méthodes qui permettent le traitement automatique du langues naturelles, qui est la base de la classification des textes utilisés dans l'analyse des opinions.Dans notre étude, et puisque les sentiments peut être considérées un flux séquentiel des données nous proposons d'utiliser LSTM comme une technique d'apprentissage.

Parce que les textes n'ont pas de structure formelle qui doit être re-représentée à une structure plus formelle avant la classification. Une multitude de techniques dans le domaine de traitement du langage ont traité ce problème en mettant en œuvre une représentation vectorielle des textes. Dans notre étude, nous utiliserons BERT l'une de ces techniques comme méthode de représentation de textes.

Le but de ce travail est de créer un système permettant de classer les opinions sur la base de l'une des méthodes de l'apprentissage en profondeur. Où il sera basé sur des opinions exprimées en langue arabe.

Ce mémoire comporte trois chapitres organisés comme suit :

Le premier chapitre sera consacré à la présentation des définitions de base dans le domaine de l'analyse des sentiments. Nous avons également discuté des difficultés de

ce domaine lors de son application sur la langue arabe,Nous examinerons également les mécanismes de Word Embeddings particulièrement BERT qui dépend dans notre système.

Le second chapitre présente brièvement les définitions de l'apprentissage en profonde, ces architectures et la méthodes LSTM que nous adopterons pour notre système.

Dans le troisième chapitre, Nous présentons notre vision conceptuelle d'un système d'analyse de sentiments extraites à partir des textes arabes, Ensuit nous montrons comment réalisé les différents tâches de ce système.

# Chapitre 1

# Analyse de sentiments arabe

## 1.1 Introduction

Le traitement automatique du langage naturel fait partie des domaines qui ont rencontré une large diffusion car il a fallu un grand défi à travers le traitement du langage qui permet la communication avec les ordinateurs sans avoir recours à des langages de programmation, ce qui a permis à nombreuses personnes d'exprimer leurs opinions sur diverses plate-formes de médias sociaux. Dans ce contexte, nous trouvons l'analyse des sentiments, qui est l'un des plus actifs champs de ce domaine, qui s'intéresse particulièrement à l'analyse des opinions.

Dans ce chapitre, nous expliquerons les notions l'analyse des sentiments, les tâches et les niveaux, aussi les méthodes d'analyse sentiments, Nous étudierons également l'analyse des sentiments en langue arabe et les différentes complications dans ce domaine .

## 1.2 Traitement Automatique des Langues Naturelles(TALN)

Le traitement automatique du langue naturel (TALN) ou encore traitement automatique des langues (TAL) est un domaine multidisciplinaire impliquant la linguistique, l'informatique et l'intelligence artificielle. Qui s'intéresse aux interactions entre les ordinateur et les langages humains. Parmi les défis qui ont été travaillés dans ce domaine on a la reconnaissance vocale, la compréhension du langage naturel et la détection de la langue.

### 1.2.1 Champs de recherche et applications de TALN

Le domaine du traitement des langues naturelles comprend un grand nombre de disciplines de recherche variés en termes d'objectif ou des méthodes de traitement, ainsi que la forme d'information. Ce domaine est divisé en trois axes principaux : Sémantique, Extraction d'informations et Syntaxe, dont chacun comprend nombreux domaines, nous mentionnons certains dans la figure suivante

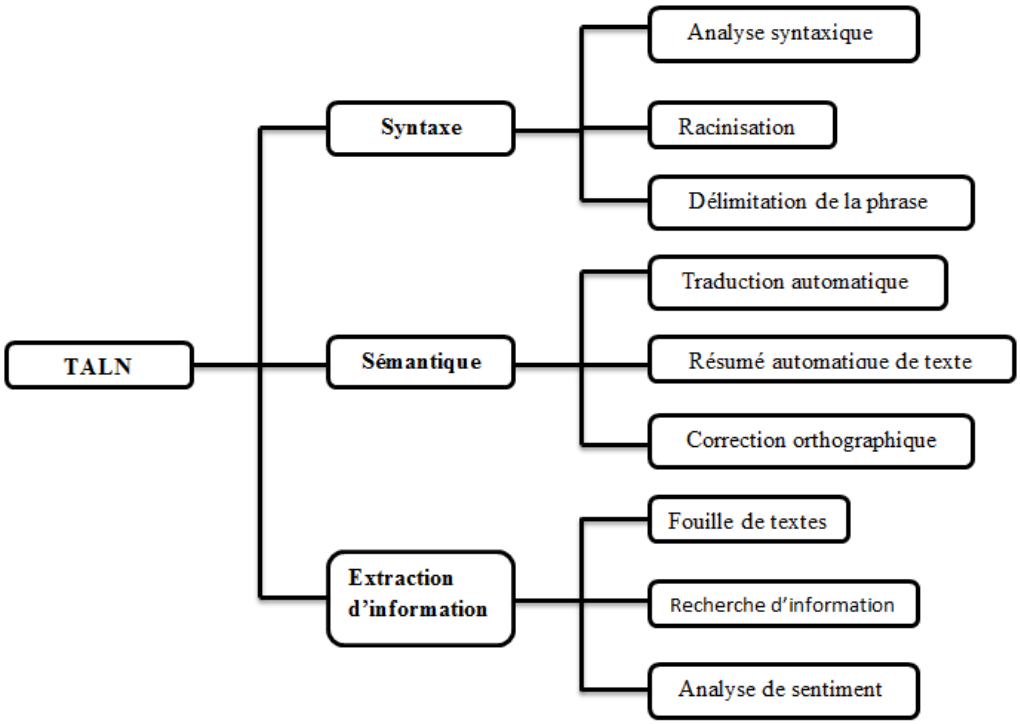



FIGURE 1.1 – Les champs de recherche et applications de TALN

Parmi les processus couramment utilisés dans le domaine du traitement du langues et en particulier lors du travail sur les textes, nous trouvons Word Embeddings , que nous expliquerons dans la section suivante

## 1.3 Word Embeddings

Word Embeddings ou plongement de mots est l'une des méthodes qui ont une grande contribution dans le domaine du traitement automatique du langage naturel. Cette technique cherche à représenter chaque mot du dictionnaire avec un vecteur de nombres réels. Cette représentation a la particularité que les mots qui apparaissent dans des contextes similaires ont des vecteurs convergents relativement proches.

Il existe différents types de plongement de mots, Nous mentionnons trois des techniques les plus utilisées :

### 1.3.1 TF-IDF

Cette méthode est basée sur la représentation de chaque mot en lui donnant un poids spécifique. Son récit est lié à d'occurrences du mot dans le document ainsi qu'à son occurrence dans le corpus. Il s'appuie également sur la réduction de poids des mots courants qui se produisent dans presque tous les documents et donnent plus d'importance aux mots qui apparaissent dans un sous-ensemble de documents.

Le poids de cette méthode est calculé en deux parties : La première partie est TF, qui dépend de la fréquence du mot dans le document. Elle est calculée de plusieurs façons :

| Schéma de pondération | formule du TF |
|---|---|
| binaire | $0, 1$ |
| fréquence brute | $f_{t,d}$ |
| normalisation logarithmique | $1 + \log(f_{t,d})$ |
| normalisation « 0.5 » par le max | $0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$ |
| normalisation par le max | $K + (1 - K)\frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$ |

FIGURE 1.2 – Les méthodes de calculer le TF

La deuxième partie IDF, qui représente la fréquence du mot dans le corpus Calculé comme suit :

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

Dj : nombre de documents où le terme Ti apparaît. Alors, le poids est :

$$\mathrm{tfidf_{i,j}} = \mathrm{tf_{i,j}} \cdot \mathrm{idf_i}$$

Le principal problème avec cette méthode est qu'elle traite le document comme un sac de mots, ce qui signifie que les mots sont indépendants .

### 1.3.2 Word2Vec

Word2vec fournit un vecteur pour chaque jeton / mot et ces vecteurs codent la signification du mot. la signification des vecteurs est compréhensible / interprétable en les comparant à d'autres vecteurs et à diverses équations intéressantes (par exemple roi-hommes + femmes = reine, ce qui prouve à quel point ces vecteurs détiennent la sémantique des mots).

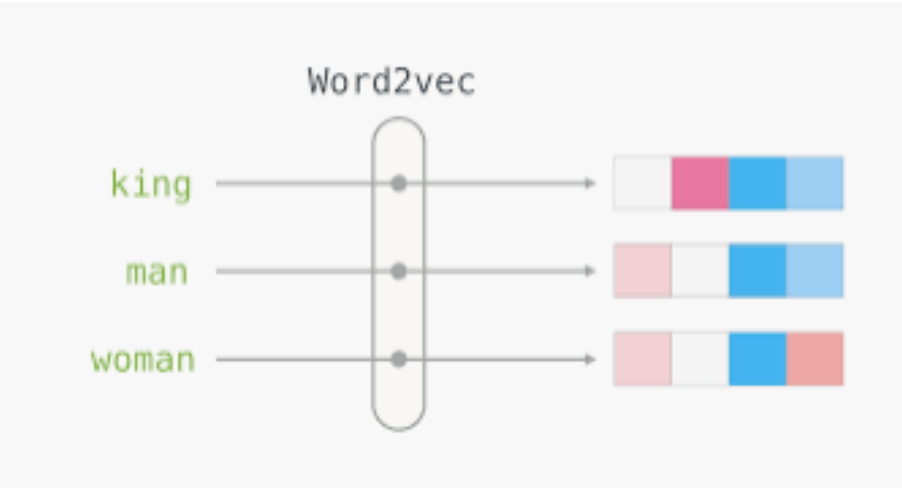



FIGURE 1.3 – Fonctionnement de word2vec

Le problème avec word2vec est que chaque mot n'a qu'un seul vecteur, mais dans le monde réel, nous constatons que le contexte dans lequel le mot est mentionné affecte sa représentation.

### 1.3.3 BERT

BERT est l'acronyme de « Bidirectionnal encoder representations from transformers ». C'est un modèle de langage dit pré-entraîner qui repose sur des réseaux neuronaux. Il a été développé par les équipes Google qui planchent depuis des années sur le traitement de langage naturel (NLP).

C'est un modèle NLP dit bidirectionnel parce qu'il lit un contenu rédactionnel dans sa globalité et dans les deux sens. Et grâce à une architecture faite par Google, le Transformateur, il apprend les relations contextuelles entre les mots. BERT peut donc mieux interpréter les différents types de demande et apporter une réponse très pertinente aux requêtes écrites ou orales.

BERT utilise Transformer,une mécanisme d'attention qui apprend les relations contextuelles entre les mots (ou sous-mots) dans un texte. Sous sa forme vanille, Transformateur comprend deux mécanismes distincts - un encodeur qui lit l'entrée de texte et un décodeur qui produit une prédiction pour la tâche. Le but du BERT étant de générer un modèle de langage, seul le mécanisme de l'encodeur est nécessaire. Le fonctionnement détaillé de Transformer est décrit dans un document de Google.

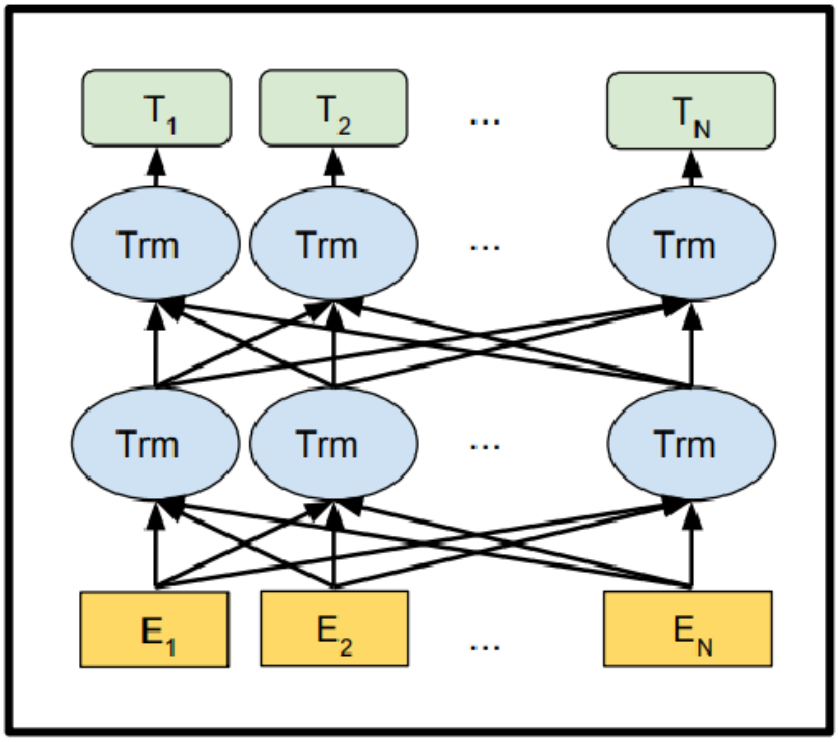



FIGURE 1.4 – Architecture de modèle BERT

Contrairement aux modèles directionnels, qui lisent l'entrée de texte séquentiellement (de gauche à droite ou de droite à gauche), le codeur Transformateur lit la séquence entière de mots à la fois. Par conséquent, il est considéré comme bidirectionnel, mais il serait plus exact de dire qu'il n'est pas directionnel. Cette caractéristique permet au modèle d'apprendre le contexte d'un mot en se basant sur l'ensemble de son environnement (gauche et droite du mot).

**Versions de BERT**

- BERT.
- RoBERTa.
- ALBERT.

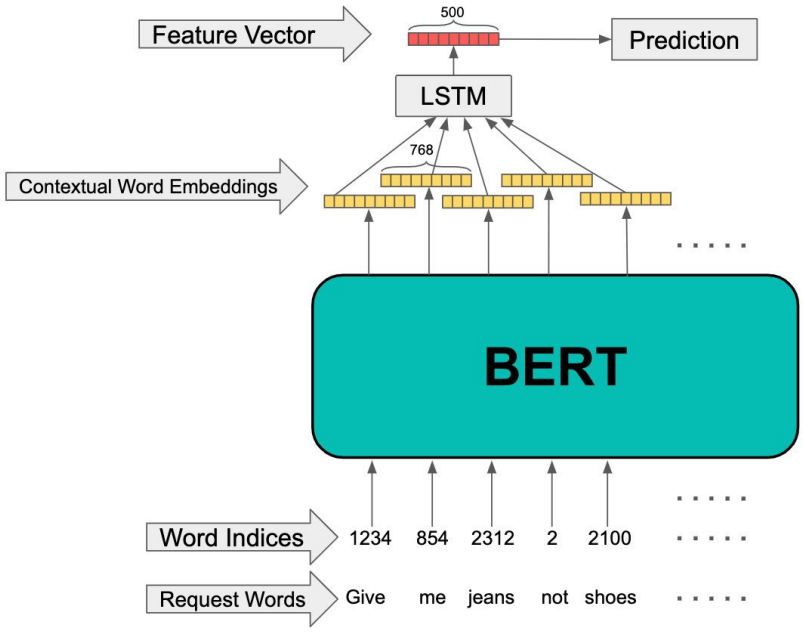



FIGURE 1.5 – Mécanisme de classification a l'aide de BERT

- berto.

## 1.4 Analyse des sentiments

Dans cette section, nous expliquons le domaine d'analyse des sentiments qui est considéré comme un axe de TALN

### 1.4.1 Sentiment

**Définition :**Connaissance plus ou moins claire, donnée d'une manière immédiate[1].

Il est également défini comme un élément émotionnel qui implique les fonctions cognitives du corps et un moyen d'appréciation. Le sentiment est à l'origine de la connaissance directe ou d'une simple impression. Il indique la perception de l'état physiologique du moment[2].

Bien qu'il existe de nombreux types de sensations, chacune a une façon différente de l'exprimer, mais elle est essentiellement divisée en deux catégories principales, les sentiments positifs et négatifs. C'est de cela que dépend l'analyse des sentiments, où vous classez l'opinion ou le sentiment à ces deux catégories.

### 1.4.2 Domaine d'analyse de sentiments

Les systèmes d'analyse des sentiments sont appliquées dans presque tous les domaines commerciaux (les sciences de gestion, Le marketing) et sociaux (les sciences sociales) même en politique. Ces raisons ont fait ce domaine l'un des plus actifs dans la recherche d'informations.

D'autre part, nous constatons que les médias sociaux ont constitué une grande base de données pour l'analyse des sentiments, qui est devenue une source d'opinions très importante, ce qui a grandement contribué à faciliter et à accélérer le processus d'analyse des sentiments.

En informatique, l'analyse des sentiments ou l'opinion mining (aussi appelé sentiment analysis) est l'analyse des sentiments à partir de sources textuelles dématérialisées sur de grandes quantités de données [3]. Le but est de classer des phrases ou des documents en fonction des sentiments.

### 1.4.3 Les tâches d'analyse des sentiments

Il existe différents différentes tâches dans l'analyse des sentiments :

- Catégorisation des sentiments.
- Identification de sujet d'opinion .
- détection de l'opinion

### 1.4.4 Niveaux d'analyse de sentiments

L'analyse des sentiments peut se produire de différents niveaux :

1. **Classification au niveau du document :**

   La classification des sentiments au niveau du document vise à automatiser la tâche de classification d'une révision textuelle, qui est donnée sur un seul sujet(par exemple Produit spécifique) [4], L'objectif est de classer un avis comme positif, négatif, ou neutre. Le problème de ce niveau est qu'il ne peut pas être appliqué aux documents qui contiennent plus d'un sujet (Par exemple de nombreux produits pour la même marque)

2. **Classification au niveau de phrase :**

   Dans ce niveau, l'objectif est déterminé si l'opinion exprimée dans une phrase est positive, négative ou neutre. Par exemple :

   - phrase positive : « Ce téléphone est très bon »
   - phrase négative : « Je rencontre de nombreux problèmes lors de l'utilisation de ce produit»
   - phrase neutre : « Je vais travailler tous les jours »

### 1.4.5 Les Méthodes d'analyse sentiments

Il existe deux grandes catégories d'analyse : l'analyse lexicale et l'analyse par apprentissage automatique :

#### L'analyse lexicale

L'approche à base de règles (ou l'approche lexique) définit un ensemble de règles dans un type de langage de programmation (script) qui identifie la subjectivité, la polarité ou le sujet d'une opinion. Cette approche peut utiliser diverses entrées, telles que[5] :

- Techniques classiques de TALN, telles que la racinisation, tokenisation, POS –tagging et Chunking.
- Autre opérations basées sur le lexique, ils utilisent le dictionnaire des sentiments avec des mots d'opinion et les faire correspondre avec les données pour déterminer la polarité.. phrase neutre : « Je vais travailler tous les jours »

**Apprentissage automatique**

Les approches automatiques reposent sur des techniques d'apprentissage automatique (Machine learning). La tâche d'analyse des sentiments est généralement modélisée comme un problème de classification dans lequel un classificateur est alimenté avec un texte et renvoie la catégorie correspondante, par ex. positif, négatif ou neutre (en cas d'analyse de polarité)[5].

### 1.4.6 Les domaines d'application d'analyse des sentiments

L'analyse des expressions d'opinion sur internet est utilisée dans nombreux domaines. Nous mentionnons les plus importants :

Dans le domaine social, cette technologie est utilisée pour identifier et analyser les opinions exprimées dans les publications en ligne et les réseaux sociaux, et pour définir leur contexte (neutre, positif, négatif) dans le but d'aborder les phénomènes sociaux.

L'analyse des sentiments a grandement aidé dans le domaine du marketing car elle cherche à déterminer les réactions des clients vers un nouveau produit ou une nouvelle campagne commerciale. Cela permet à l'entreprise de connaître ses faiblesses du point de vue du client et de travailler à les améliorer.

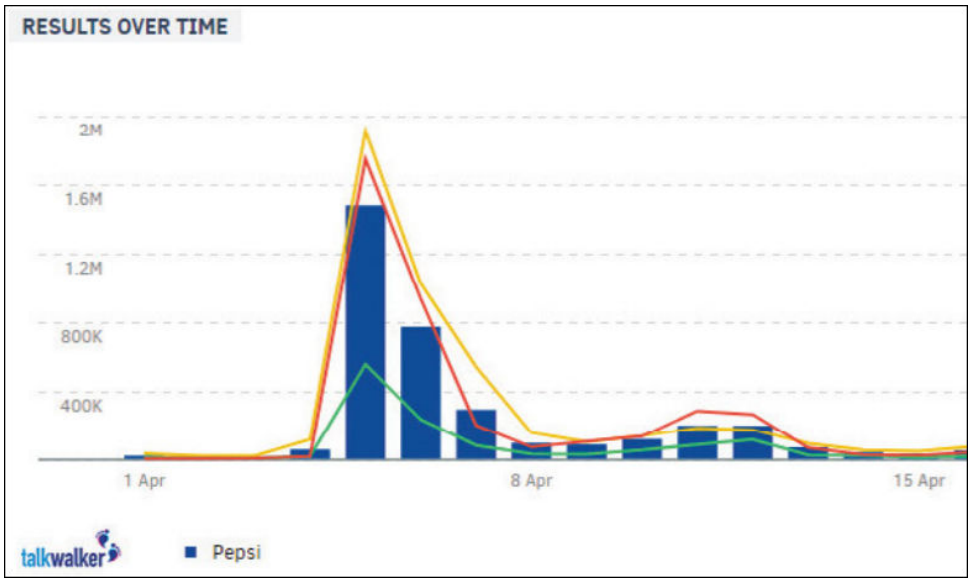



FIGURE 1.6 – Exemple d'analyse dans le domaine du marketing

L'analyse d'opinion interfère également avec l'aspect politique, car elle est généralement utilisée pour aider à prendre des décisions, en particulier celles qui nécessitent l'avis

de la société.

Il peut également être utilisé pour extraire des opinions sur certains des phénomènes les plus courants à l'heure actuelle et de nombreuses opinions ont été proposées à ces sujets, qui doivent être analysées pour déterminer le point de vue des utilisateurs, par exemple : racisme, virus Corona .....

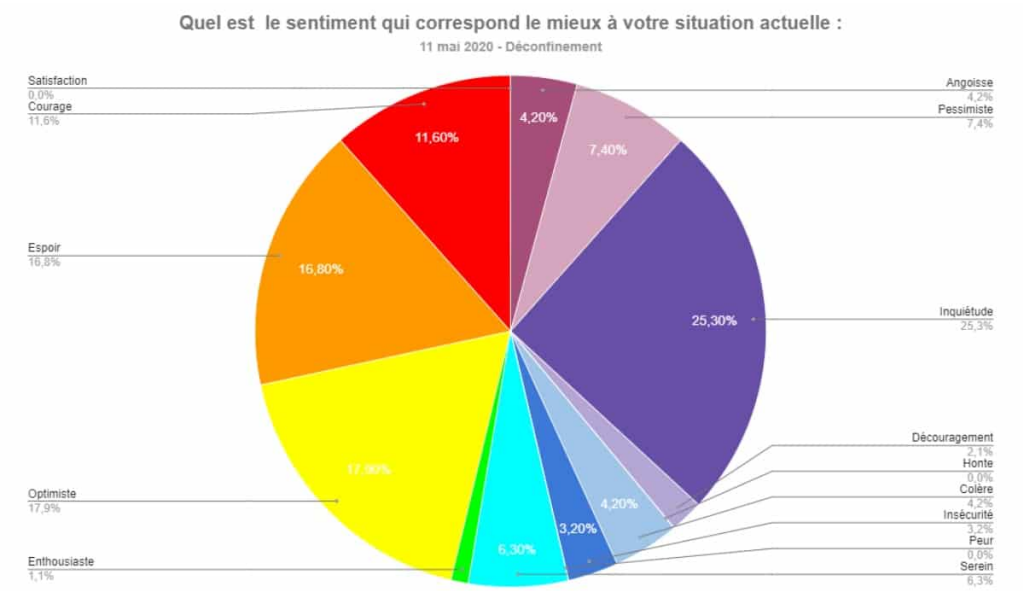



FIGURE 1.7 – Comment la communauté " Jardinerie Animalerie Fleuriste" et fournisseurs réagissent à la 1ère semaine de confinement soumise à l'enquête

## 1.5 Analyse de sentiments en arabe

La plupart des travaux de recherche effectués dans ce domaine ont été menés sur des langues européennes (surtout en anglais) et asiatiques (japonais et chinois). Néanmoins, très peu de travaux ont été réalisés sur le plan des langues qui sont morphologiquement riches (comme l'arabe et tchèque)[6].

La langue arabe est l'une des langues les plus parlées dans le monde, Elle est parlée par plus de 467 millions des gens comme première langue. elle occupe la quatrième ou la cinquième place en termes de langues les plus répandues dans le monde, et la quatrième langue en termes de nombre d'utilisateurs sur Internet[3].

### 1.5.1 COMPLEXITÉ DE LA LANGUE ARABE

- **Signification des mots :**il est possible d'identifier différentes significations associées à un mot, en raison d'un mot peut avoir plusieurs sens dans différents contextes.

| **Mot signifiant** | **Phrase** |
|---|---|
| Cœur | في قلب الحدث |
| Cœur | اجرى عملية قلب مفتوح |
| milieu, | الكرة في قلب الملعب |

TABLE 1.1 – Exemple de différents signification d'un mot

- **Variations dans la catégorie lexicale :** un mot peut avoir plus qu'une catégorie lexicale (nom, verbe, adjectif, etc.) dans différents contextes, comme le montre le tableau suivant :

| **Catégorie de mot** | **Phrase** |
|---|---|
| Nom propre | عين جالوت |
| Nom | عين الانسان |
| Verbe | عين وزير للخارجية |

TABLE 1.2 – Les catégories lexicales seul mot

- **Synonymes :** les langues ont beaucoup de mots considérés comme synonyme. À travers un corpus donné, les chercheurs peuvent utiliser des outils d'analyse morphologique pour connaître les synonymes d'un mot, la fréquence de chacun mot de ces synonymes et lequel d'entre eux est plus commun[7].

| **mot** | **Synonymes** |
|---|---|
| donner | بذل منح اعطى |
| famille | اسرة عائلة |

TABLE 1.3 – Les Synonymes

- **La forme verbale selon son cas :** La forme de certains mots arabes peut changer en fonction de leurs modes de cas (nominatif, accusatif ou génitif)[7].

| mot | pluriel | pluriel en arabe |
|---|---|---|
| voyageur | voyageurs | مسافرون (مرفوعة)<br>مسافرين (منصوبة |

TABLE 1.4 – Exemple de la forme verbale

- **Tachkil :** On retrouve également tachkil qui contribue à changer le sens du mot d'une phrase à l'autre.
- en arabe à partir d'une même racine, nous pouvons générer diverses formes de mots qui n'ont pas une signification similaire [8].

| mot | racine |
|---|---|
| merveilleux رائع<br>terrible مروع | روع |

TABLE 1.5 – Exemple de stemming arabe

### 1.5.2 Travaux en relation

Malgré les nombreuses difficultés que la langue arabe contient, de nombreux défis dans le domaine de l'analyse des sentiments à l'égard de cette langue ont été observés ces dernières années. Dans cette section, nous mentionnons quelque travaux :

M. Elhawary et M. Elfeky [9] expliquent comment extraire les avis commerciaux dispersés sur le web écrit en langue arabe. Les critiques extraites sont analysés pour exprimer également leurs sentiments (positifs, négatifs ou neutre). De cette façon, ils ont fourné à leurs utilisateurs les informations qu'ils besoin sur les entreprises locales dans la langue comprendre et donc offrir une meilleure expérience de recherche pour la région du Moyen-Orient, qui parle principalement l'arabe.

Le prétraitement du texte est une étape importante dans l'exploration de texte , Motaz K. Saad et W. Ashour [10] ont étudié l'influence de la phase de prétraitement du texte, la racinisation ou désuffixation (stemming) , normalisation de texte et la pondération des termes sur la classification des textes arabe.

Mountassir, H.Benbrahim et I.Berrada [11] présentent une étude de classification su-

pervisée des sentiments dans un contexte arabe.Ils ont utilisé deux corpus arabes qui sont différents à bien des égards. Pour qu'ils utilisent trois classificateurs communs connus pour leur efficacité, à savoir Naïve Bayes, Support Vector Machines et k-Nearest Neighbour. Ils ont étudié certains paramètres pour identifier ceux qui permettent d'obtenir les meilleurs résultats. Ces paramètres concernent le type de radical, le seuil de fréquence des termes, la pondération des termes et les mots de n grammes. qui montrons que les Bayes naïfs et les machines à vecteurs de support sont efficaces sur le plan de la concurrence ; cependant k-nn dépend du corpus.. leurs résultats montrent que les performances de classification peuvent être influencées par la longueur des documents, l'homogénéité des documents et la nature des auteurs des documents. Cependant, la taille des ensembles de données n'a pas d'impact sur les résultats de la classification.

A. Ziani, Y. Tlili Guiassa et N. Azizi [12] ont proposé un système qui opère en trois phases, la première consiste à la construction et le prétraitement manuel du corpus recueillis à partir des journaux arabes algériens. La seconde phase est le choix des caractéristiques pour la représentation des commentaires. Enfin la troisième phase est la réalisation du module de classification combinant quatre classificateurs SVM avec des fonctions noyaux différents. Ils ont utilisé deux stratégies nommé un contre un et un contre tous dont les résultats ont prouvé que la première stratégie est meilleure que la deuxième avec les commentaires des journaux en langue arabe.

Nora Al-Twairesh ,Hend Al-Khalifa , AbdulMalikAlsalman et Yousef Al-Ohaliont [13] sont décidé de relever le défi et de travailler sur le dialecte arabe, où une méthode hybride a été conçue pour analyser les sentiments dans le dialecte soudanais.

## 1.6 Conclusion

Dans ce chapitre, nous avons vu des définitions de base dans le domaine de l'analyse des sentiments. Nous avons également discuté des difficultés de ce domaine lors de son application sur la langue arabe

Dans le chapitre suivant,Nous présentons une vue générale sur le deep learning,qui est la technologie que nous adopterons dans notre travail.

# Chapitre 2

# L'apprentissage en profondeur

## 2.1 Introduction

Les applications d'intelligence artificielle existent aujourd'hui capable de reconnaître et d'analyser les commandes vocales, les images automatiquement, aide les experts pour prendre des décisions dans des environnements complexes et sophistiqués.

Pour réaliser ces tâches, ils sont dotés de module d'apprentissage leur permettant d'adapter leur comportement à des situations jamais rencontrées ou d'extraire des lois à partir de bases de données.C'est exactement ce qui intéresse le domaine d'apprentissage automatique. Parmi les sous-domaines de l'apprentissage automatique, nous trouvons l'apprentissage en profondeur.

Dans ce chapitre, nous verrons brièvement l'apprentissage automatique. Nous expliquerons également l'apprentissage en profondeur, les réseaux de neurones qui sont à la base de la construction de l'apprentissage en profondeur. Enfin, nous verrons comment les mots sont représentés afin de les préparer à l'apprentissage.

## 2.2 L'apprentissage automatique

C'est un champ d'étude de l'intelligence artificielle qui concerne le développement des algorithmes qui fait largement appel à des outils et des concepts statistique et mathématique pour donner à l'ordinateur la capacité d'apprendre des données sans être explicitement programmé.

**Domaines d'application de l'apprentissage automatique**

Le domaine de l'apprentissage automatique ne se limitait pas aux technologies de l'information uniquement en traitement de l'information, mais aidait également dans de nombreux domaines par exemple :

- **la santé :** trouver des médicaments pour guérir les maladies incurables.
- **Le marketing.**
- **La sécurité :** Sécurité de l'information.
- **Robotique :** Contribution au développement de robots capables d'exécuter des tâches similaires à l'homme et parfois de résoudre des problèmes humains insolubles

### 2.2.1 Types d'apprentissage automatique

Les algorithmes d'apprentissage automatique sont divisés en deux parties principales :

1. **Apprentissage supervisé**

   L'apprentissage est dit supervisé lorsque les données qui entrent dans le processus sont déjà catégorisées et que les algorithmes doivent s'en servir pour prédire un résultat en vue de pouvoir le faire plus tard lorsque les données ne seront plus catégorisées [14].

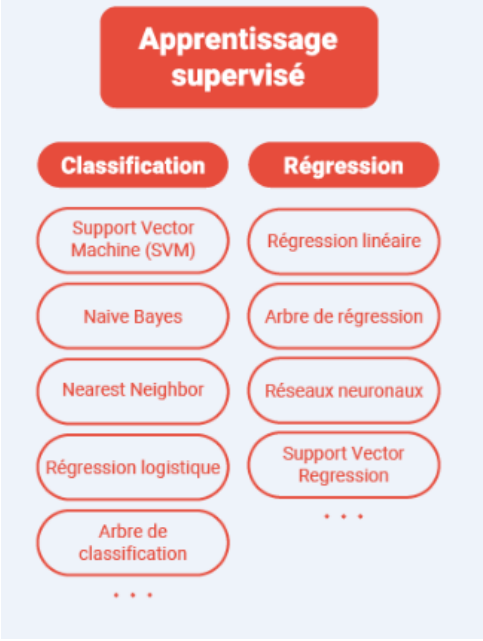



FIGURE 2.1 – Les méthodes d'apprentissage supervisé

2. **Apprentissage non supervisé** L'apprentissage non supervisé est beaucoup plus complexe puisqu'ici le système va détecter les similarités dans les données qu'il reçoit et les organiser [14]. Également ce type contient deux tâches principales, la clustering et l'association


FIGURE 2.2 – Les méthodes d'apprentissage non supervisé

## 2.3 Apprentissage en profondeur

L'apprentissage en profondeur est un sous-domaine d'apprentissage automatique qui tente de modéliser des données avec architectures complexes combinant différentes transformations non linéaires. Qui permet à la machine de construire des concepts complexe à partir de concepts simple (Ces concepts sont organisés en couches). La « profonde » dans « l'apprentissage en profondeur » fait référence au nombre de couches à travers lesquelles les données sont transformées.

Par exemple, Dans une application de reconnaissance d'image, la première entrée peut être une matrice de pixels, la première couche de représentation le codage de pixels et, la deuxième couche peut composer et arrangements de codage de bords, la troisième couche peut coder pour un nez et les yeux, et la quatrième couche peut reconnaître que l'image contient un visage.

## 2.4 Les réseaux de neurones

Les briques élémentaires de l'apprentissage en profondeur sont les réseaux de neurones, qui sont combinés pour former les réseaux de neurones profonds [15] . alors dans cette section nous définirons le neurone, le neurone artificiel et ce qu'est un réseau de neurones.

### 2.4.1 Neurone

Cellule de base du tissu nerveux, capable de recevoir, d'analyser et de produire des informations. (La partie principale, ou corps cellulaire du neurone, est munie de prolongements, les dendrites et l'axone) [1] .

Les neurones reçoivent des signaux (impulsions électriques non linéaire) par les dendrites et envoient l'information par les axones. Les contacts entre deux neurones se font par l'intermédiaire des synapses ( terminaison neuronale ). La figure suivante montre les composants de la cellule nerveuse :

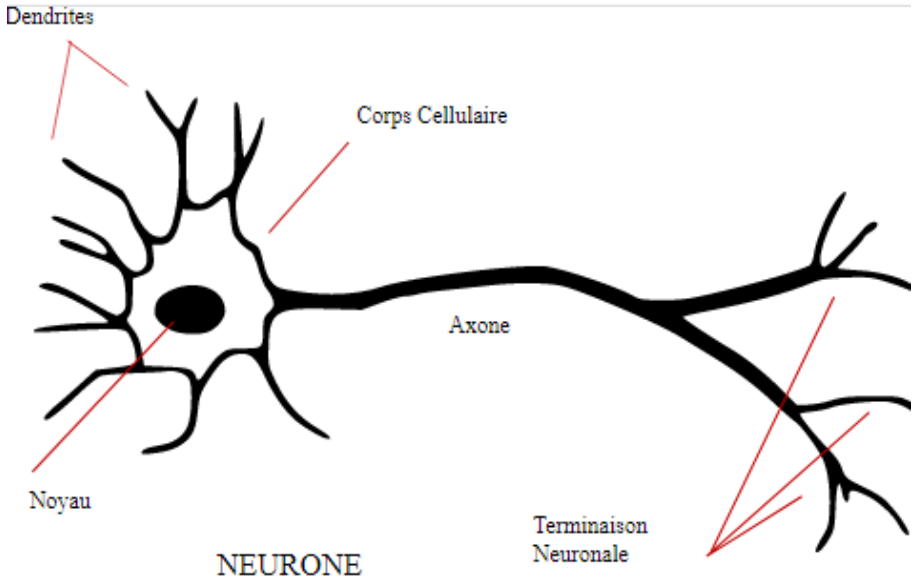



FIGURE 2.3 – Cellule neuronale

### 2.4.2 Neurone artificiel

Un neurone artificiel est une fonction $f_j$ de l'entrée $x = (x_1,..., x_d)$ pondérée par un vecteur de poids de connexion $w_j = (w_j, 1,..., w_j, d)$, complété par un biais neuronal $b_j$, et associé à une fonction d'activation , à savoir : $y_j = f_j(x) = (hw_j, x_i + b_j)$ [?].

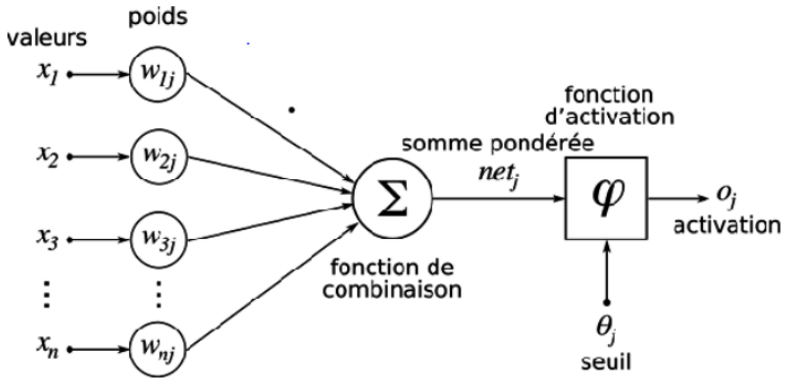



FIGURE 2.4 – Un neurone artificiel

Il existe plusieurs fonctions d'activation. Le tableau suivant présente quelques-unes :

| Nom de la fonction | Relation d'entrée/sortie | Icône | Nom Matlab |
|---|---|---|---|
| seuil | $a = 0$ si $n < 0$<br>$a = 1$ si $n \geq 0$ | | hardlim |
| seuil symétrique | $a = -1$ si $n < 0$<br>$a = 1$ si $n \geq 0$ | | hardlims |
| linéaire | $a = n$ | | purelin |
| linéaire saturée | $a = 0$ si $n < 0$<br>$a = n$ si $0 \leq n \leq 1$<br>$a = 1$ si $n > 1$ | | satlin |
| linéaire saturée symétrique | $a = -1$ si $n < -1$<br>$a = n$ si $-1 \leq n \leq 1$<br>$a = 1$ si $n > 1$ | | satlins |
| linéaire positive | $a = 0$ si $n < 0$<br>$a = n$ si $n \geq 0$ | | poslin |
| sigmoïde | $a = \frac{1}{1+\exp^{-n}}$ | | logsig |
| tangente hyperbolique | $a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$ | | tansig |
| compétitive | $a = 1$ si $n$ maximum<br>$a = 0$ autrement | C | compet |

FIGURE 2.5 – Les fonctions d'activation

## 2.5 Les réseaux de neurones

Un réseau de neurones est un modélisation mathématique du cerveau humain composé d'un ensemble d'algorithmes inspirés de la structure et de la fonction des neurones humains

Un réseau de neurones est un ensemble séquentiel des couches Chacune contient des unités appelées neurones artificiels. Chaque unité correspond à une des variables d'entrée($x_1$, $x_2$,...$x_n$). (On peut rajouter une unité de biais qui est toujours activée).

Ces unités sont reliées à une seule et unique unité de sortie, qui reçoit la somme des unités qui lui sont reliées, pondérée par des poids de connexion (appelés aussi coefficient synaptique) ($w_1$ ,$w_2$.....,$w_n$).

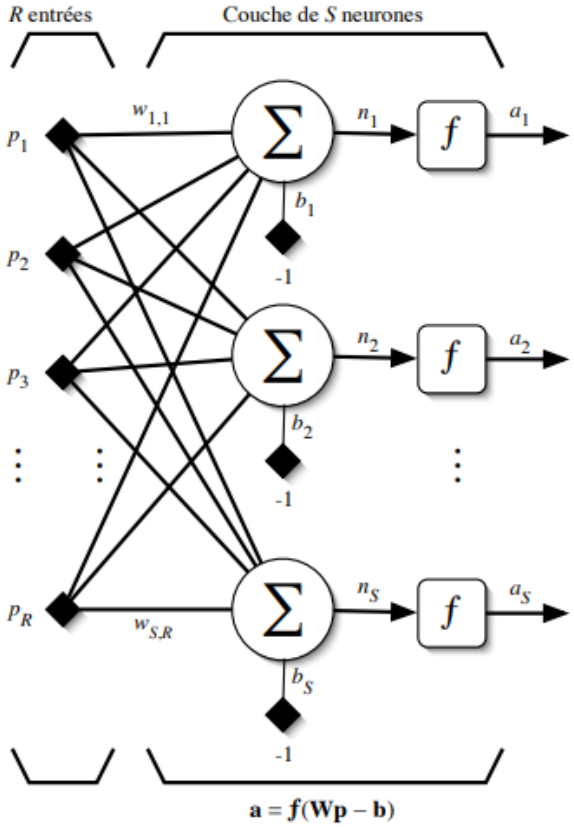



FIGURE 2.6 – Une couches de réseau de neurones

La sortie reçoit donc w0+wjxj. L'unité de sortie applique alors une fonction d'activation à cette sortie. Cette dernière, à son tour est une variable d'entrée pour les unités des couches suivent

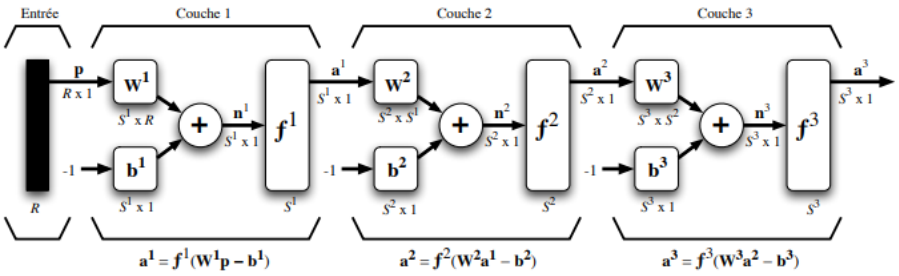



FIGURE 2.7 – Un réseau de neurones de 3 couches

### 2.5.1 Architectures réseaux de neurones

#### Réseau de neurones convolutif

Un réseau de neurones convolutif est un réseau neuronal multicouche inspiré biologiquement du cortex visuel humain. Est très utilisés pour l'analyse d'images dans les applications graphique qui concerne le traitement et reconnaissance d'images ou vidéo. Les deux caractéristiques principales du réseau convolutif sont qu'ils utilisent des filtres (kernel) et mettent en œuvre du pooling.

Les filtres analysent les images zones par zones. Chaque filtre se spécialise de façon à reconnaître des motifs (patterns). Un filtre peut par exemple se spécialiser dans la détection des contours, tandis qu'un autre reconnaîtra certaines formes. La convolution a pour effet d'augmenter la profondeur de la matrice correspondant à l'image, puisque chaque filtre lui ajoutera une couche. Une image qui a une profondeur de 3 couches (le nombre 3 correspondant aux 3 canaux RGB) pourra ainsi résulter en une matrice d'une profondeur de 5, si le réseau convolutif est constitué de 5 filtres.

Le pooling permet quant à lui de réduire la taille d'une image en n'en conservant que les pixels les plus importants. Cela a pour effet de déformer l'image en perdant le positionnement précis des pixels. Cet effet est en fait bénéfique, puisqu'il permet de limiter les risques de sur apprentissage. A titre d'exemple, un système de détection des visages aura tout intérêt à apprendre qu'un visage est constitué de deux yeux, d'un nez et d'une bouche

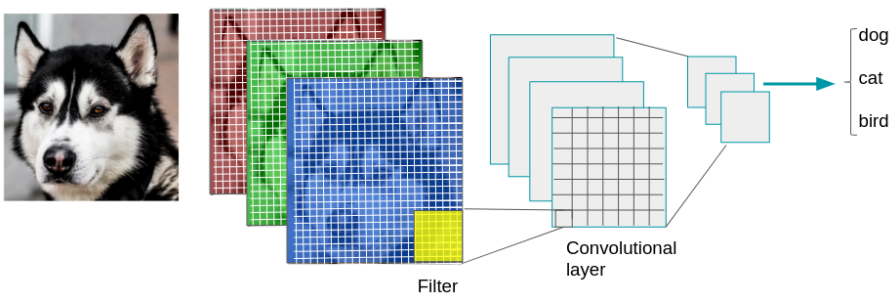



FIGURE 2.8 – Système de détection des visages avec CNN

**Réseau de neurones récurrents (RNN)**

Les réseaux de neurones récurrents ont contribué à de nombreuses améliorations fondamentales dans divers domaines tels que la reconnaissance vocale, la composition automatique de la musique, l'analyse des sentiments, le séquençage de l'ADN et la traduction automatique.

Contrairement aux réseaux de neurones traditionnels qui supposent que chaque couche a sa propre poids et biais ,RNN convertit les activations indépendantes en activations dépendantes en fournissant les mêmes poids et biais à toutes les couches, réduisant ainsi la complexité de l'augmentation des paramètres et en mémorisant chaque sortie précédente en donnant chaque sortie en entrée à la couche cachée suivante.

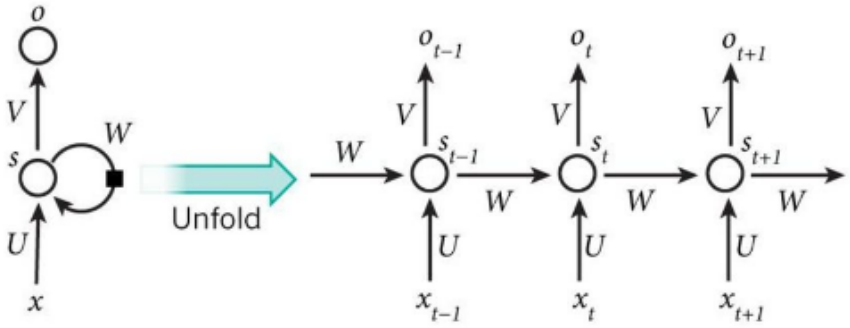



FIGURE 2.9 – Une couche RNN

L'une des plus grands inconvénients des réseaux de neurones récurrents traditionnels est la sensibilité à la longueur de la séquence, de sorte que si nous voulons prédire la séquence après 1000 virgules, par exemple, nous constaterons que le modèle a oublié son point de départ(les points de connaissance). Dans ce contexte, nous constatons que LSTM « long short-term memory», qui est un type de réseau de neurones récurrent,Viser à améliorer cet aspect.

## 2.6 Long short-term Memory (LSTM)

Un réseau LSTM est un réseau neuronal récurrent qui a des blocs de cellules LSTM à la place de couches de réseau neuronal standard.

Ces cellules ont divers composants appelés la porte d'entrée, la porte d'oubli et la porte de sortie. Ces portes à la capacité de supprimer ou d'ajouter des informations pour

protéger et contrôler l'état des cellules, Chacun d'eux se compose d'une couche de réseau neuronal sigmoïde et d'un processus de multiplication ponctuelle.

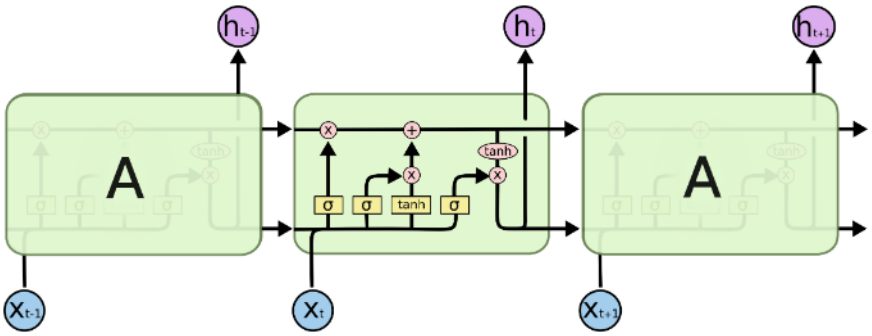
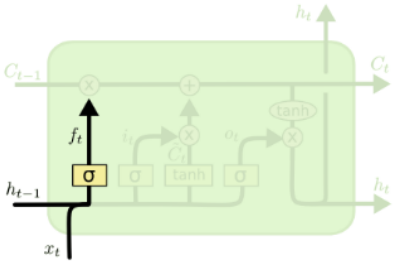



FIGURE 2.10 – Cellule LSTM

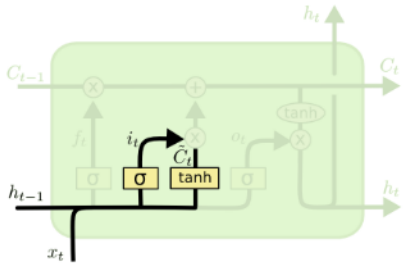
La porte d'oubli est la première étape par laquelle les informations qui seront exclues de la cellule sont déterminées. Cela se fait par des calculs qui sont appliqués à Xt qui représente la nouvelle entrée et ht-1 et est la sortie de la couche précédente. Il produit un nombre compris entre 0 et 1. Pour chaque nombre dans le cas de la cellule Ct - 1. Le nombre 1 représente "retenu" tandis que le nombre 0 représente "élimination".



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

FIGURE 2.11 – La porte d'oubli d'une cellule LSTM

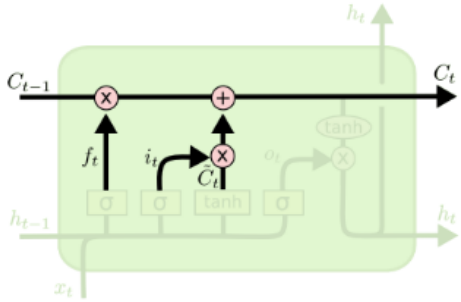
Dans l'étape suivante, les nouvelles informations qui seront stockées dans la cellule sont déterminées par la porte d'entrée, qui est définie les valeurs à mettre à jour. D'autre part, la couche tanh crée un vecteur de nouvelles valeurs candidates, C' t, qui pourraient être ajoutées à l'état.




$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

FIGURE 2.12 – La porte d'entrée d'une cellule LSTM

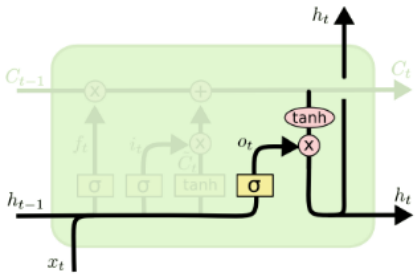
Ensuite, l'état de l'ancienne cellule Ct-1 est mis à jour a état nouvelle de la cellule Ct, c'est en multipliant l'ancien état par ft en y ajoutant également les nouvelles valeurs de candidates.




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

FIGURE 2.13 – Mis à jour à l'état de la cellule

Enfin, ce que la cellule va éjecter, c'est un filtrage de son état, et c'est via la "porte de sortie" ce qui multiplie l'état de la cellule dont elle est extraite. Aux valeurs productives avec la fonction tanh




$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

FIGURE 2.14 – La porte de sortie d'une cellule LSTM

### 2.6.1 Versions de LSTM

L'architecteur évoquée précédemment explique la généralité de LSTM, mais on retrouve également un ensemble de versions contenant quelques légères différences, nous citerons quelques-unes :

Une version LSTM populaire, introduite par Gers et Schmidhuber (2000), ajoute des «connexions de judas». Cela signifie que nous laissons les couches de porte examiner l'état des cellules [16].

$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \; + \; b_o\right)$$

Une autre version consiste à utiliser des portes couplées d'oubli et d'entrée. Au lieu de décider séparément ce qu'il faut oublier et ce à quoi nous devons ajouter de nouvelles informations, nous prenons ces décisions ensemble. Nous n'oublions que lorsque nous allons saisir quelque chose à sa place. Nous n'introduisons de nouvelles valeurs dans l'état que lorsque nous oublions quelque chose de plus ancien [16].

$$C_t = f_t * C_{t-1} + \boldsymbol{(1 - f_t)} * \tilde{C}_t$$

Une variation légèrement plus spectaculaire du LSTM est le « Gated Recurrent Unit » ou GRU, introduit par Cho, et al. (2014). Il combine les portes d'oubli et d'entrée dans une seule «porte de mise à jour». Il fusionne également l'état de cellule et l'état caché et apporte d'autres modifications. Le modèle résultant est plus simple que les modèles LSTM standard et est devenu de plus en plus populaire [16].

## 2.7 Conclusion

Dans ce chapitre, nous avons présenté les définitions de l'apprentissage en profonde ,ces architectures et LSTM que nous adopterons pour notre système.

Dans le chapitre suivant. Nous avons montré la réalisation de notre système.

# Chapitre 3

# Réalisation de système

## 3.1 Introduction

Dans ce chapitre, Nous présentons notre vision conceptuelle d'un système d'analyse de sentiments extraites à partir des textes arabes, dont nous proposons d'utiliser le modèle Bert pour transformer les mots composants les sentiments textuels en vecteurs numérique, et puisque les sentiments peut être considérées un flux séquentiel des données nus proposons d'utiliser LSTM comme une technique d'apprentissage.

Ensuite, nous montrerons comment mettre en œuvre les différentes phases mentionnées ci-dessus avec les outilles les plus importantes utilisées pour cela, ainsi que les résultats obtenus.

## 3.2 Conception

Dans cette section , nous présenterons la conception globale et détaillée de notre système

### 3.2.1 Conception globale du système

Dans cette partie , nous présentons une vision globale de notre système qui est l'analyse des sentiments à l'aide de l'apprentissage en profondeur. Notre système est divisé en trois étapes de base comme le montre figure suivante :

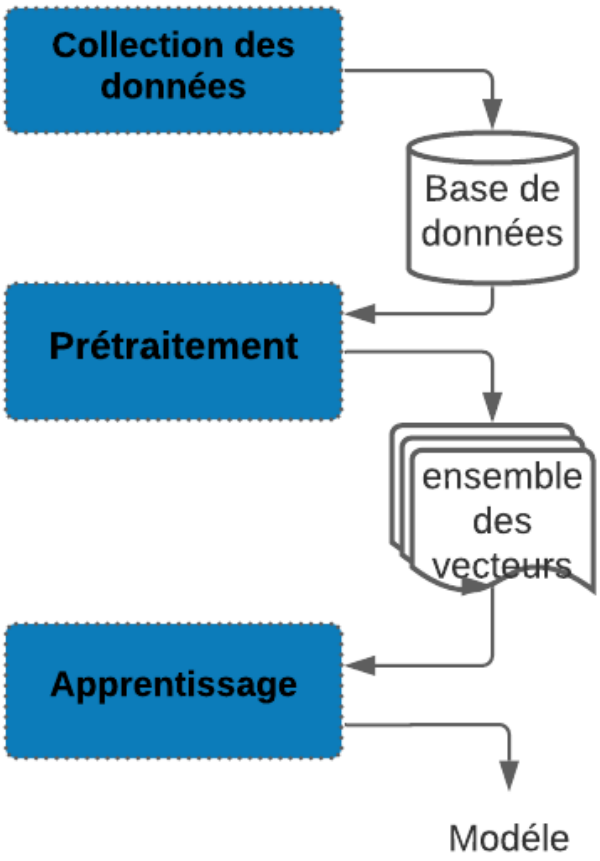



FIGURE 3.1 – Conception globale du système

La première étape consiste à collecter les données sur lesquelles le système travaillera, qui sont représentées par un ensemble des sentiments recueillies à partir de diverses sources.

La deuxième étape est divisée en groupe de sous étapes, qui est le pré-traitement dont la phrase est convertie en un vecteur numérique à travers un ensemble de processus, suivi d'un processus d'apprentissage sur ces vecteurs numériques à travers lequel un modèle de classification est obtenu.

### 3.2.2 Conception détaillée du système

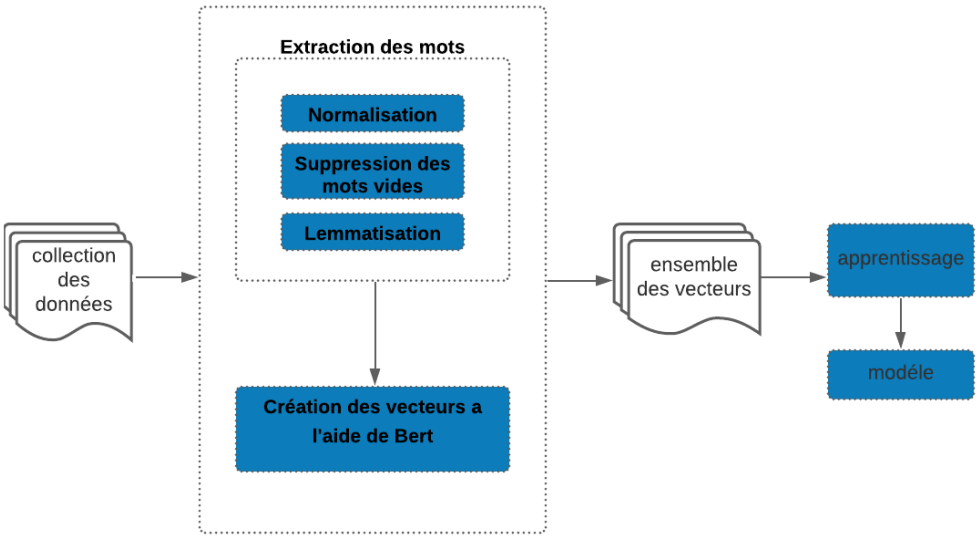



FIGURE 3.2 – Conception détaillée du système

#### Collection de données

Les réseaux sociaux sont l'une des sources de données les plus importantes, notamment en ce qui concerne les opinions ou les sentiments, car cela fait appel à un grand nombre d'internautes pour émettre leurs opinions et exprimer leurs sentiments.

Dans ce travail, nous appuierons sur un ensemble de bases de données contenant des avis et opinions exprimés en langue arabe centrés sur les hôtels, les livres, les films, les produits et quelques compagnies aériennes [17].

Où nous baserons dans notre système sur un échantillon de ces données pour que chaque ligne contienne une phrase et une classe (négative ou positive) qui lui correspond.

#### Prétraitement

Cette étape est considérée comme l'une des étapes les plus importantes qui précèdent le processus d'apprentissage, car il pour extraire uniquement des données importantes. Il se compose de deux étapes de base, qui sont l'extraction de mots et création des vecteurs.

**A. Extraction des mots :**

1. Normalisation : également connu sous le nom de processus de standardisation, qui consiste à convertir le document dans un format standard facile à manipuler.

Cette étape se fait à travers un ensemble d'étapes, la plus importante d'entre elles :

- Suppression de caractères : /, ? ,*, !,. . . . . .
- Suppression de nombres.
- Suppression des mots et les caractères non arabes.

2. Suppression des mots vides :Nous pouvons définir ces mots comme des mots qui n'ont aucun effet sur la classification.

   cette étape consiste à supprimer tous les mots vides en comparant chaque mot connu avec les éléments de la liste de mots vides. Nous pouvons définir ces mots comme des mots qui n'ont aucun effet sur la classification.

   La liste des mots vides en langue arabe comprend environ 162 mots, dont certains sont mentionnés dans le tableau suivant :

| | |
|---|---|
| **Les Outils de liaison** | و ،ثم ،ف ،ثم ،او ،ام ..... |
| **Pronoms** | انا، نحن، انتم ،هن ،..... |
| **Les Outils conjonctions** | في ،ب ،..... |

TABLE 3.1 – Les types de mots vides arabe

On considère également les mots fréquemment utilisés, et on retrouve dans la plupart des phrases des mots vides. Par exemple :

اكثر ، ، امام ، ايضا ، بعض

3. Lemmatisation : L'une des étapes les plus difficiles, elle nous montre que l'arabe est une langue inflexible, cela est dû aux difficultés que contient ce langage, qui ont été mentionnées dans le premier chapitre. Il s'agit de trouver la racine du mot.

Premièrement, la structure du mot doit être connue en langue arabe, comme suit :

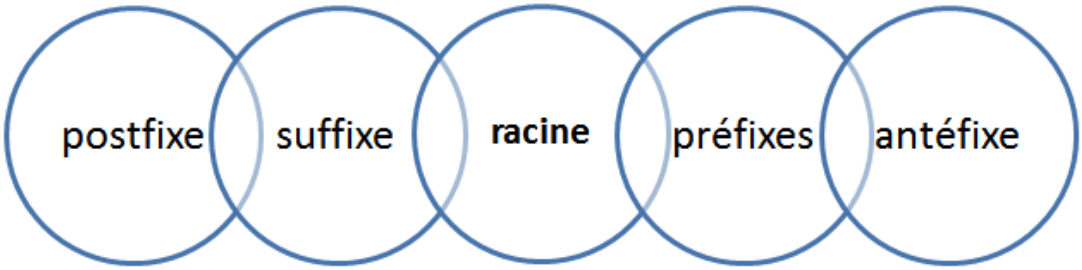



FIGURE 3.3 – la structure du mot en langue arabe

**Exemple :**

اتتذكروننا

| | |
|---|---|
| **Antéfixe** | أ |
| **Préfixe** | ت |
| **La racine** | ذكر |
| **Suffixe** | ون |
| **Post fixe** | نا |

TABLE 3.2 – Exemple de lemmatisation

Les techniques de racinisation reposent principalement sur l'utilisation d'une liste de suffixes et une autre de préfixes pour réduire les mots fléchis a leurs stems. Parmi ces techniques on a :

- Aljlayl Frieder [2002]
- Larkey Connell [2002] (light10)
- Chen Gey [2002]
- Darwish Orad [2003]
- Kadri Nie [2008]
- Abdelali et al. [2016]

Dans le tableau suivant, nous avons une liste de suffixes et une autre de préfixes pour chacune des techniques précédentes :

| Méthode | Préfixes | Suffixes |
|---|---|---|
| Aljlayl | و ، ال ، وال ، كال ، ست ، سي ، ل، ب ، ت ، ي ، لل ، ال | ين ، ون ، ات ، ة ، ان ، ي ، هم ، هن |
| Light10 | لل، ال ، وال، بال، كال ، فال، و | ها ، ان ، ات ، ون ، ين ، يه ، يةه، ة، ي |
| AL-Stem | وال، بال، فال، بت، يت، لت، ، مت، وت، ست، لم، بم نت، وم، كم، فم، ال، لل، وي، لي، في، وا، فا، لا، با | ات، وا ، ونهوه، ان، ، تي، ته، تم، كم، هم هن ، ها، ية، تك، نا ، ين، يه، ة، ه، ي، ا |
| Chen | وال، بال، فال، كال ، ولل، مالها ال، لال، فا ، كا، ول، وي، وس ، سي، سال لا، وب ، وت، وم، لل، با، و ، ب، ل | ها، ية ،هم، ن، ما، و، يا ، ني كن، تم، تن، ين، يا، ه، كم ان، ات، ون، ة، ه، ي، ت |

FIGURE 3.4 – Les suffixes et préfixes des techniques de la lemmatisation

Dans notre travail, nous appliquerons la segmentation à nos données en utilisant Farasa segmenteur qui basé sur SVM qui utilise une variété de fonctionnalités et de lexiques pour classer les segmentations possibles d'un mot. Les caractéristiques comprennent : la probabilité des tiges, des préfixes, des suffixes, leurs combinaisons ; présence dans des lexiques contenant des racines valides ou entités nommées [18].

**B. Création des vecteurs**

Étant donné que les textes ne sont pas formels. Par conséquent, avant le processus d'apprentissage, ils doivent être convertis en vecteurs numériques qui sont caractérisés par leurs formalité.

Alors la prochaine étape est le processus de représentation des mots, qui conduira à un groupe de vecteurs prêts à apprendre sur lesquels nous appuierons deux différent modèles :

M-BERT, qui est une variante multilingue de BERT, avec exactement la même architecture et les mêmes API. Les variantes de modèles linguistiques multilingues et monolingues sont pré-entraînées, de manière non supervisée, en utilisant les mêmes approches de modélisation de langage masqué (MLM) et d'inférence de langage naturel (NLI) .Ce-

pendant, alors que BERT monolingue original a été formé au corpus anglais de Wikipedia et de BooksCorpus, m-BERT est formé à une concaténation de Corpus Wikipedia de 104 langues, possédant ainsi une compréhension de texte multilingue [19]. Il existe actuellement deux versions du modèle modèles multilingues :

- BERT-Base, Multilingual Uncased : version originelle Support 102 langues
- BERT-Base, Multilingual Cased : Support 104 langues

Nous utilisons Le modèle Multilingual Cased qui est corrige également les problèmes de normalisation dans de nombreuses langues, il est donc recommandé dans les langues avec des alphabets non latins (il est souvent meilleur pour la plupart des langues avec des alphabets latins) [20].

Le deuxième modèle est AraBERT qui est un modèle de langue arabe pré-formé basé sur l'architecture BERT de Google .il utilise la même configuration BERT-Base. Le modèle a été formé sur 70 millions de phrases ou 23 Go de texte arabe avec 3 milliards de mots [21]. Il existe deux versions du modèle :

- AraBERTv0.1
- AraBERTv1

**Apprentissage**

Dans cette phase ,nous compterons sur l'une des techniques de réseau de neurones récurrent qui est LSTM, Avant de commencer le processus d'apprentissage, nous devons d'abord avoir deux bases de données disponibles, la première est utilisée pour entraîner le modèle, la seconde est pour le test, de sorte que nous allons diviser la grande base de données qui y était précédemment mentionnée en ces deux bases (90 pourcent entraînement, 10 pourcent test).

Ensuite, le réseau de neurones est formé sur une base d'entraînement afin d'obtenir un modèle classification, et ce dernier sera validé à partir de la base de test.

Si le modèle entraîné n'est pas valide, un processus de revue des paramètres sera mis en œuvre au cours duquel les paramètres du réseau neuronal sont modifiés afin d'obtenir de meilleurs résultats.

La figure suivante montre un diagramme de la façon dont le processus d'apprentissage se déroule :

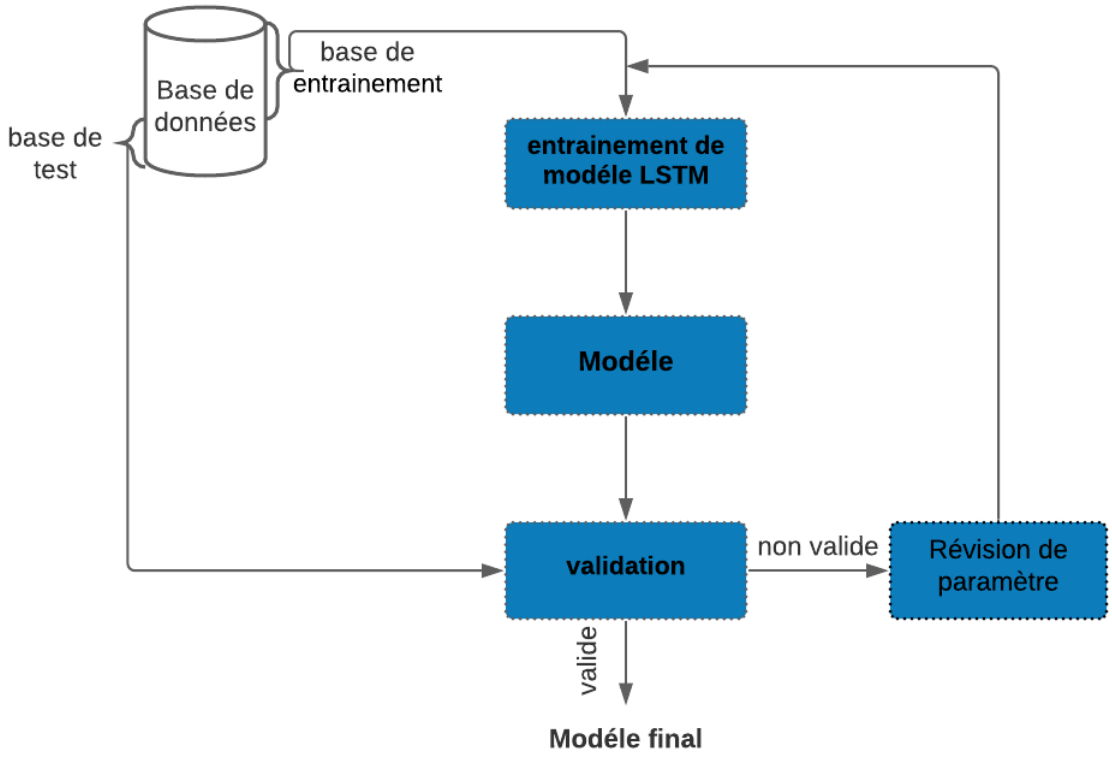



FIGURE 3.5 – Le procédure d'apprentissage

## 3.3 Réalisation de système

Après avoir présenté la conception de notre système et mentionné ses étapes les plus importantes. Dans cette section, nous montrerons comment mettre en œuvre ces étapes et présenterons les résultats les plus importants obtenus.

### 3.3.1 Outils et Environnement de programmation

Afin de réaliser notre système, nous avons utilisé un ensemble d'outils et d'environnements de programmation, Nous citerons les plus importants d'entre eux :

**Google Colab** ou **Colaboratory** est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de l'apprentissage

automatique directement dans le cloud. Sans donc avoir besoin de l'installer sur notre ordinateur à l'exception d'un navigateur [22].

FIGURE 3.6 – Logo de colab

Pour valider notre application nous adopterons sur **Python** qui est un puissant langage de programmation polyvalent. Il est utilisé dans le développement Web, la science des données, la création de prototypes de logiciels, etc. Python a une syntaxe simple et facile à utiliser [23].

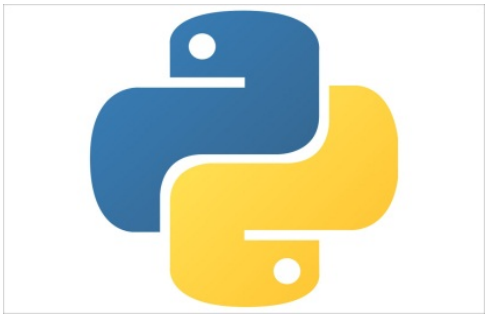


FIGURE 3.7 – Logo de Python

Parmi les caractéristiques les plus importantes de Python est qu'il est riche dans de nombreuses bibliothèques, ce qui a un langage facile à utiliser. Cela nous a permis dans notre système d'utiliser un groupe de bibliothèques, dont le plus important est :

**TensorFlow** qui est une plate-forme open source de bout en bout pour l'apprentissage automatique. Il dispose d'un écosystème complet et flexible de bibliothèques, d'outils, et de ressources communautaires qui permet aux chercheurs de pousser les avancées de pointe en matière d'apprentissage automatique et aux développeurs de créer et de déployer facilement des applications basées sur l'apprentissage automatique. [24].


FIGURE 3.8 – Logo de TensorFlow

**Keras** est une API de réseaux de neurones de haut niveau, écrite en Python et interfaçable avec TensorFlow, CNTK et Theano. Elle a été développée avec pour objectif de permettre des expérimentations rapides. Être capable d'aller de l'idée au résultat avec le plus faible délai possible étant la clef d'une recherche efficace [25].


FIGURE 3.9 – Logo de Keras

Nous avons également utilisé la bibliothèque **Pandas** la manipulation des données, **Scikit-Learn**, **matplotlib** pour visualiser des données sous formes graphiques...

### 3.3.2 Le système développé

Dans cette section, nous montrerons comment réaliser les différentes étapes du système qui ont été expliquées précédement

#### Prétraitement

Avant de démarrer le processus de prétraitement des données, nous devons d'abord vérifier que la base de données est équilibrée afin que le nombre des instances pour chaque classe soit convergent. Le diagramme suivant montre que la base de données que nous avons utilisée est équilibrée de sorte que chaque classe contient environ 3700 instances

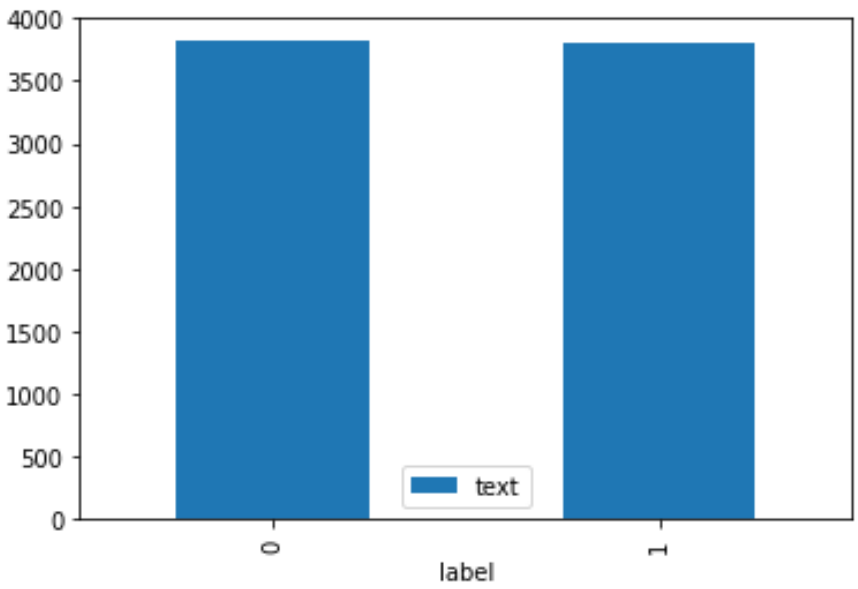



FIGURE 3.10 – Diagramme de nombre d'instances de les classes

Pour entraîner notre modèle, nous devons d'abord traiter notre ensemble de données en suivant ces étapes :

- Premièrement, le processus de normalisation, à cette étape, les mots vides sont d'abord supprimés, en fonction de la liste « arabic.text », qui contient une liste de mots vides, suivi de l'étape de suppression des mots non arabes et des caractères spéciaux.
- Deuxièmement, le processus de lemmatisation, que nous avons mis en œuvre sur la base de la bibliothèque «Al Farasa».

  La figure suivante montre comment nous avons implémenté ces étapes :

```
with open('arabic.txt',"r",encoding="utf-8") as f2:
    stopword = f2.readlines()
st=[]
for stop in stopword:
    stop=re.sub(r'[^\u0600-\u06ff\u0750-\u077f\ufb50-\ufbc1\ufbd3-\ufd3f\ufd50-\ufd8f\ufd50-\ufd8f ]',
                '',
               stop)
    st.append(stop)
for stop in st:
    stop=' '+stop+' '
    df['text']=df['text'].replace(stop,' ', regex=True)
df=df.replace('_', ' ', regex=True)
df['text']=df['text'].replace(r'[^\u0600-\u06ff\u0750-\u077f\ufb50-\ufbc1\ufbd3-\ufd3f\ufd50-\ufd8f\ufd50-\u
                              '',
                             regex=True)
farasa_segmenter = FarasaSegmenter(interactive=True)
df['text'] = df['text'].apply(lambda row: preprocess(row, do_farasa_tokenization=True,
                                                     farasa=farasa_segmenter,
                                                    use_farasapy = True))
```

FIGURE 3.11 – La réalisation de prétraitement

Dans le tableau suivant, nous avons un exemple sélectionné au hasard dans la base de données pour laquelle différentes étapes de prétraitement ont été appliquées :

| | |
|---|---|
| **Phrase** | تعاون احتواء نظافة . . ازعاج اطفال النزلاء طوال اليوم وعدم استجابتهم او تعاونهم مع الطاقم ـ |
| **Phrase sans mots vides** | تعاون احتواء نظافة . . ازعاج اطفال النزلاء طوال وعدم استجابتهم تعاونهم الطاقم |
| **Phrase normalisé** | تعاون احتواء نظافة ازعاج اطفال النزلاء طوال وعدم استجابتهم تعاونهم الطاقم |
| **Phrase segmenté** | تعاون احتواء نظاف +ة ازعاج أطفال ال+ نزلاء طوال و+ عدم استجاب +ت +هم تعاون +هم ال+ طاقم |

TABLE 3.3 – Exemple de prétraitement

- L'étape suivante est le processus de création des vecteurs, dans lequel nous sommes appuyés sur deux versions de BERT , La figure suivante montre la fonction qui a été utilisée pour obtenir le vecteurs numériques :

```
1 def tokenize(text):
2     return tokenizer.convert_tokens_to_ids(tokenizer.tokenize(text))
```

FIGURE 3.12 – Création des vecteurs

La figure suivante montre le dernier processus de cette étape, au cours duquel la base de données a été divisée (entraînement/test) :

```
import math
TOTAL_BATCHES = math.ceil(len(sorted_sent_labels) / BATCH_SIZE)
train_size = int(0.8 * TOTAL_BATCHES)
test_size = int(0.1 * TOTAL_BATCHES)
val_size = int(0.1 * TOTAL_BATCHES)
full_dataset = batched_dataset.shuffle(TOTAL_BATCHES)
train_data = full_dataset.take(train_size)
test_data = full_dataset.skip(train_size)
val_data = test_data.skip(test_size)
```

FIGURE 3.13 – Division de données

### Entrainement de modèle

Puisque notre ensemble de données est prêt, nous allons maintenant concevoir une architecture de modèle de réseau neuronal récurrent à l'aide du modèle séquentiel Keras avec des configurations de réglage comme indiqué dans la figure suivant :

```
model=Sequential()
model.add(layers.Embedding(VOCAB_LENGTH, output_dim=128 ))
model.add(layers.LSTM(units=128,return_sequences=True))
model.add(layers.LSTM(units=32))
model.add(layers.Flatten())
model.add(layers.Dense(8))
model.add(layers.Dropout(0.25))
model.add(layers.Dense(1, activation="sigmoid"))
```

FIGURE 3.14 – Architecture du modèle

Notre modèle a été entraîné sur la base de 60 époques. Nous dépendions également de l'optimiseur Adam comme indiqué dans la figure suivante :

### Evaluation de modèle

L'évaluation est l'étape au cours de laquelle nous vérifions si le modèle est le mieux adapté aux données de test. Comme le montre la figure suivante, dans cette étape, nous utilisons la fonction evaluate qui approuvée par keras :

```
opt = keras.optimizers.Adam(learning_rate=0.01)
model.compile(optimizer=opt, loss="binary_crossentropy",
     metrics=['accuracy'])
hist=model.fit(train_data , epochs=60,validation_data=val_data)
loss, acc = model.evaluate(train_data)
print("Training Accuracy: ", acc)
```

FIGURE 3.15 – Entrainement du modèle

```
loss, acc = model.evaluate(test_data)
print("Test Accuracy: ", acc)
```

FIGURE 3.16 – Evaluation du modèle

Après avoir vérifié la précision et le taux d'erreur sur la base du test. Nous concluons si le modèle est bien formé ou nécessite un processus de révision des paramètres au cours duquel les données du modèle sont réinitialisées.

### La prédiction

À ce stade, notre modèle est prêt pour le processus de classification, qui est basé sur la fonction predict de Keras , comme le montre la figure suivante :

```
seq=[tokenize(text)]
pred = model.predict(seq)
pred[pred<0.5]=0
pred[pred>=0.5]=1
if pred==0 :
  print('negative')
else :
  print('positive')
```

FIGURE 3.17 – La fonction de prédiction

### 3.3.3 Résultats

Dans cette partie, nous présenterons dans le tableau ci-dessous les résultats les plus importants obtenus grâce aux deux versions m-bert et ara-bert, tout en conservant les mêmes paramètres du modèle de classification, ainsi que les mêmes opérations de prétraitement :

| | **M-BERT** | **Ara-BERT** |
|---|---|---|
| Précision d'entraînement | 0.81 | 0.98 |
| taux d'erreur d'entraînement | 0.38 | 0.02 |
| Précision de test | 0.78 | 0.99 |
| taux d'erreur de test | 0.37 | 0.02 |
| Temps d'entrainement | 23 min 13 s | 25 min |

TABLE 3.4 – Les résultats

Afin de suivre de plus près les résultats obtenus, il devient plus clair de les suivre à travers des graphiques. Les graphiques suivants nous permettent de suivre l'évolution de la précision et du taux d'erreur lors de l'utilisation les versions m-bert et ara-bert

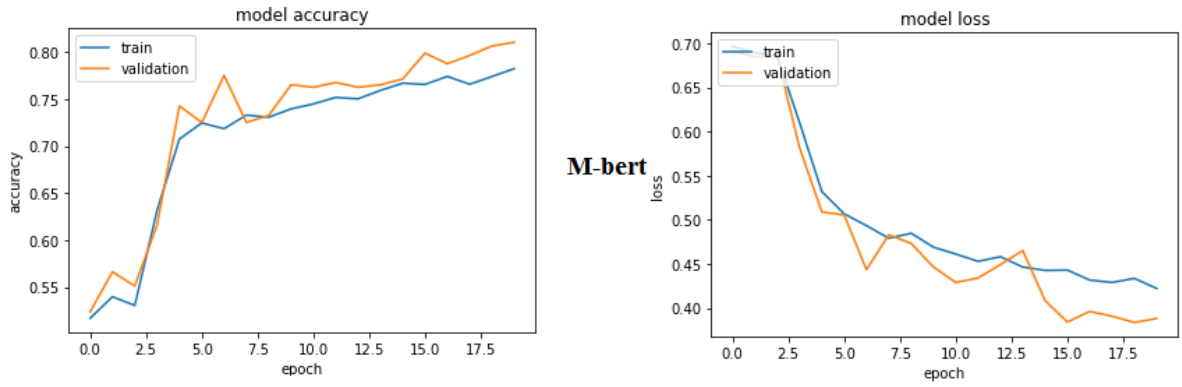



FIGURE 3.18 – Évolution de la précision et du taux d'erreur M-Bert

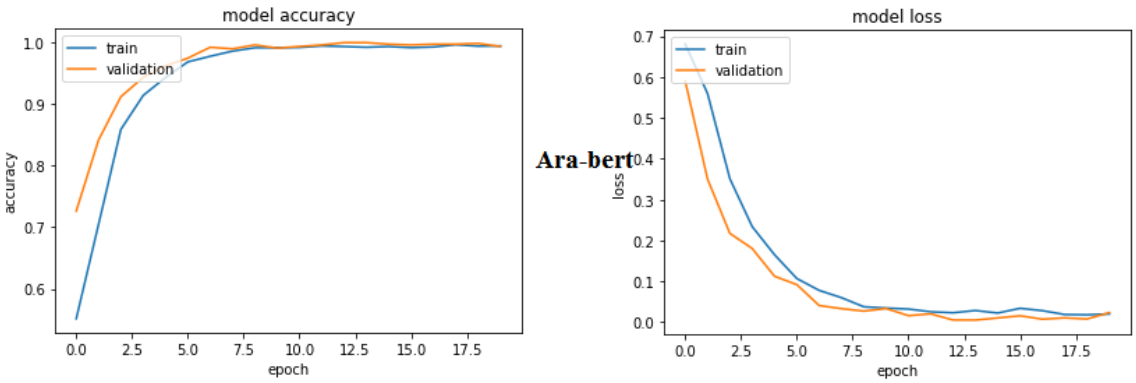



FIGURE 3.19 – Évolution de la précision et du taux d'erreur Ara-Bert

## 3.4 Conclusion

Dans ce chapitre, nous avons présenté comment mettre en œuvre les différentes étapes pour créer un modèle d'analyse des sentiments , où nous avons précédemment expliqué en détail la conception des ces étapes.

Nous avons également fait une simple comparaison entre deux versions de BERT, où nous constatons que la version qui pré-entraîné sur la langue arabe a donné des résultats relativement mielleux que celle qui a été pré-entraîné sur multilingue.

# Chapitre 4

# Conclusion générale

Dans ce travaille , nous avons créé un système d'analyse des sentiments basé sur les capacités du deep learning, en particulier LSTM, qui est l'une de ses techniques qui a connu un grand succès dans le domaine du traitement du langage.Nous avons également utilisé le modèle de BERT pour convertir les mots qui forment des sentiments textuels en vecteurs numériques afin de aligner avec la propriété de LSTM, qui traite les données comme un flux séquentiel.

Nous espérons au future de former des systèmes d'analyse des sentiments sur les textes arabes exprimés en lettres latines ainsi que sur ceux qui contiennent des mots non arabes, qui sont considérés comme une grande quantité de données qui n'ont pas encore été étudiées.

# Bibliographie


[1] Larousse.fr : encyclopédie et dictionnaires gratuits en ligne.

[2] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. page 94.

[3] Wikipedia.

[4] Rodrigo Moraes, João Francisco Valiati, and Wilson P. Gavião Neto. Document-level sentiment classification : An empirical comparison between SVM and ANN. *Expert Systems with Applications*, 40(2) :621–633, February 2013.

[5] Mehdi Hadji. Analyse des sentiments : Généralités, August 2019.

[6] Behnam Roshanfekr, Shahram Khadivi, and Mohammad Rahmati. Sentiment analysis using deep learning on Persian texts. In *2017 Iranian Conference on Electrical Engineering (ICEE)*, pages 1503–1508, May 2017.

[7] (PDF) Arabic Morphological Tools for Text Mining.

[8] Mohamed Ali Sghaier, Houssem Abdellaoui, Rami Ayadi, and Mounir Zrigui. Analyse de sentiments et extraction des opinions pour les sites e-commerce : application sur la langue arabe. page 7.

[9] Mohamed Elhawary and Mohamed Elfeky. Mining Arabic Business Reviews. In *2010 IEEE International Conference on Data Mining Workshops*, pages 1108–1113, Sydney, TBD, Australia, December 2010. IEEE.

[10] Motaz K Saad and Wesam Ashour. Arabic Text Classification Using Decision Trees. page 5, 2010.

[11] A. Mountassir, H. Benbrahim, and I. Berrada. A cross-study of Sentiment Classification on Arabic corpora. In Max Bramer and Miltos Petridis, editors, *Research and Development in Intelligent Systems XXIX*, pages 259–272, London, 2012. Springer.

[12] ACL Anthology - ACL Anthology.

[13] Nora Al-Twairesh, Hend Al-Khalifa, AbdulMalik Alsalman, and Yousef Al-Ohali. Sentiment Analysis of Arabic Tweets : Feature Engineering and A Hybrid Approach. *arXiv :1805.08533 [cs]*, May 2018. arXiv : 1805.08533.

[14] Apprentissage Supervisé Vs. Non Supervisé, January 2019.

[15] Neural networks and introduction to deep learning,https ://www.math.univ-toulouse.fr/.

[16] Understanding LSTM Networks – colah's blog.

[17] Kaggle : Your Machine Learning and Data Science Community.

[18] Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. Farasa : A fast and furious segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Demonstrations*, pages 11–16, San Diego, California, June 2016. Association for Computational Linguistics.

[19] Dan Garrette Telmo Pires, Eva Schlinger. How multilingual is multilingual bert ?, 2019.

[20] google-research/bert,https ://github.com/google-research/bert,visité juin 2020.

[21] (PDF) AraBERT : Transformer-based Model for Arabic Language Understanding.

[22] Google Colab : Le guide Ultime, May 2019. Section : Data Science.

[23] Learn Python Programming,https ://www.programiz.com/python-programming, visité 2020-09-19.

[24] tensorflow/tensorflow, September 2020. original-date : 2015-11-07T01 :19 :20Z.

[25] Débuter avec Keras - Documentation en français,https ://www.actuia.com/keras/.