



République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la
recherche scientifique

Université Larbi Tébessi - Tébessa

Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie
Département : Mathématiques et Informatique



Mémoire de fin d'étude
Pour l'obtention du diplôme de **MASTER**
Domaine : Mathématiques et Informatique
Filière : Informatique
Option : Systèmes d'information

Thème

**Machine Learning pour un Système d'Authentification
des versets du saint coran online**

Présenté Par :

Bouagal Asma

Devant le jury :

Dr. Bendib Issam	MCB	Université Larbi Tébessi	Président
Dr. Hadjadj Ismail	MAA	Université Larbi Tébessi	Examineur
Pr. Laouar Med Ridda	Prof	Université Larbi Tébessi	Encadreur

Date de soutenance : 15/09/2020

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Résumé

Notre travail décrit un système de classification des textes arabes et coranique en fonction des similarités. Nous avons utilisé des techniques d'apprentissage automatique dans lesquelles nous avons appliqué de nombreux filtres et classificateurs. Les meilleurs résultats ont été obtenus en utilisant l'algorithme LSTM (Long Short-Term Memory), avec une exactitude de l'ordre de 86,67%, une perte de 21,27%, cela sans enlever le TASHKEEL. Et avec le TASHKEEL on obtient une exactitude de l'ordre de 100%, une perte de 17,8%. Nous avons observé que les diacritiques peuvent avoir un impact négatif sur l'exactitude et la perte s'ils sont utilisés avec la technique de "Word Tokenizer" dans la phase de prétraitement.

Mot clés : Classification des textes, Langue arabe, Coran, TASHKEEL, Diacritiques, Word Tokenizer.

Abstract

Our work describes a system for classifying Arabic and Koranic texts according to similarities. We used machine learning techniques in which we applied many filters and classifiers. The best results were obtained using the LSTM (Long Short-Term Memory) algorithm, with an accuracy of around 86.67%, a loss of 21.27%, without removing the TASHKEEL. And with the TASHKEEL we got an accuracy of around 100%, a loss of 17.8%. We have observed that diacritics can have a negative impact on accuracy and loss if used with the "Word Tokenizer" technique in the preprocessing phase.

Keywords: Text classification, Arabic language, Koran, TASHKEEL, Diacritics, Word Tokenizer.

ملخص

يصف عملنا نظاما لتصنيف النصوص العربية والقرآنية وفقا لأوجه التشابه. استخدمنا تقنيات التعلم الآلي التي طبقنا فيها العديد من الفلاتر والمصنفات. تم الحصول على أفضل النتائج باستخدام خوارزمية LSTM (Long Short-Term Memory)، وبدقة بلغت حوالي 86.67%، وبفقد 21.27%، دون إزالة "تشكيل". ومع "تشكيل" حصلنا على دقة حوالي 100%، بخسارة 17.8%. لقد لاحظنا أن علامات التشكيل يمكن أن يكون لها تأثير سلبي على الدقة والخسارة إذا تم استخدامها مع تقنية "Word Tokenizer" في مرحلة المعالجة المسبقة.

الكلمات المفتاحية: تصنيف النص ، اللغة العربية ، القرآن ، التشكيل ، التشكيل ، رمز الكلمات.

Dédicace

Toutes les lettres ne sauraient trouver les mots qu'il faut...

Tous les mots ne sauraient exprimer la gratitude,

L'amour, le respect, la reconnaissance...

Aussi, c'est tout simplement que



Je dédie ce modeste travail ...

À MES CHERS PARENTS

Aucun hommage ne pourrait être à la hauteur de l'amour Dont ils ne cessent de me combler. Que dieu leur procure bonne santé et longue vie.

Mes Chers Frères

mes nièces

mes neveux

à tous mes collègues d'étude

Tous mes amis ...

Et à toute la famille sans exception ...

Bouagal Asma

Remerciement

Avant tout, nous remercions Dieu le tout puissant en qui nous avons trouvé la force, le courage et la volonté pour la réalisation ce modeste travail.

« Celui qui ne remercie pas les gens, ne remercie pas Allah »

[Authentique Hadith]

Je voudrais tout d'abord exprimer mes plus profonds remerciements à mon encadreur Pr.LAOUAR MED RIDDA pour son accord d'être mon directeur de mémoire et de sa disponibilité et son aide pendant toute la préparation de ce travail.

Je tiens aussi à remercier tous les membres de jury : Dr.Bendib Issam et Dr.Hadjadj Ismail, pour leur disponibilité et acceptation d'examiner et de rapporter mon travail.

Je tiens également à remercier nos enseignants qui nous ont dispensés durant deux ans de master, leurs précieux conseils et orientations.

Enfin, Que tous ceux qui directement ou indirectement m'ont apporté leur aide, trouvent ici l'expression de mes sincères remerciements.



TABLES DES MATIERES

INTRODUCTION GENERALE	1
CHAPITRE I : CATEGORISATION ET CLASSIFICATION DES TEXTES	3
1. INTRODUCTION	4
2. DEFINITION	5
3. TYPES DE LA CLASSIFICATION AUTOMATIQUE	5
3.1. La classification supervisée	5
3.2. La classification non supervisée	6
3.3. Avantages et inconvénients	6
4. PROCESSUS DE LA CATEGORISATION AUTOMATIQUE	6
4.1. Représentation des textes	7
4.1.1. Représentation en sac de mots	8
4.1.2. Représentation des textes avec des racines lexicales	8
4.1.3. REPRESENTATION DES TEXTES AVEC DES LEMMES	8
4.1.4. Représentation des textes avec des N-gramme	8
4.1.5. Représentation des textes par des phrases	9
4.2. Choix des classificateurs	9
4.3. EVALUATION DE LA QUALITE DES CLASSIFICATEURS	9
5. Les applications de la catégorisation des textes	12
6. Les problèmes de la classification des textes	12
a. LA REDONDANCE	12
b. L'AMBIGUÏTE	12
c. LA GRAPHIE	12
d. Complexité de l'algorithme d'apprentissage	13

e.	Présence-Absence de terme	13
f.	Les mots composés	13
7.	Conclusion	14
CHAPITRE II : TRAITEMENT AUTOMATIQUE DE LA LANGUE ARABE		15
1.	Introduction	16
2.	traitement Automatique De La Langue (TAL)	17
3.	Niveaux traitement automatique de la langue	18
4.	Traitement Automatique De La Langue Arabe (TALA)	19
5.	La langue arabe	19
5.1.	Particularité de la langue arabe	20
a.	Les voyelles	20
b.	Les agglutinations	21
c.	Irrégularité de l'ordre des mots dans la phrase	21
5.2.	Morphologie arabe	21
5.3.	Structure d'un mot	23
5.4.	Catégorie d'un mot	24
6.	Difficultés du traitement automatique de la langue arabe	26
6.1.	Ambiguïté	26
6.2.	Absence des voyelles	27
6.3.	La segmentation de textes	27
6.4.	Agglutination de mots	27
7.	Quelques travaux sur la classification du saint coran	28
7.1.	Vérification de l'intégrité et de l'authentification en ligne des Versions électroniques du Coran	28
7.2.	Approche basée sur les résidus pour l'authentification modèle de textes arabes diacritiques multi-styles	28
7.3.	Une classification topique	29
7.4.	Utilisation du Deep Learning pour déterminer automatiquement l'application correcte de Règles	

de base de la récitation coranique	30
8. CONCLUSION	32
CHAPITRE III : LES CLASSIFICATEURS	33
1. Introduction	34
2. Algorithmes d'apprentissage	35
2.1. algorithme des k-voisins les plus proches KNN	35
2.1.1. Définition	35
2.1.2. Principes de fonctionnement	36
2.1.3. Critiques de la méthode	37
2.1.4. Les domaines d'application	37
2.2. Les arbres de décision	37
2.2.1. Définition	37
2.2.2. Algorithme	38
2.2.3. Critiques de la méthode	39
2.2.4. Les domaines d'application	39
2.3. Machines à support de vecteurs (ou SVM)	40
2.4. Réseaux de neurones	41
2.5. Classification naïve bayésienne	42
2.5.1. Description du modèle bayésienne	43
2.5.2. Estimation de la valeur des paramètres	45
2.5.3. Construire un classificateur à partir du modèle de probabilités	46
2.5.4. Analyse	47
2.6. Réseau de neurones récurrents et « long short-term memory »	47
3. Conclusion	51
CHAPITRE IV : EXPERIMENTATION ET IMPLEMENTATION	52
1. Introduction	53
2. Présentation du corpus	54
2.1. Qu'est-ce qu'un corpus ?	54
2.2. Le projet TANZIL	54
2.3. Pourquoi TANZIL ?	55
2.4. Fautes de frappe dans certains textes existants	55
3. La mise en œuvre de l'approche proposée	56

3.1.	Le langage utilisé	56
3.2.	Environnement d'exécution	57
3.3.	Les bibliothèques utilisées	59
3.4.	Phase 1 prétraitement	60
	3.4.1. Nettoyage et prétraitement du texte	60
3.5.	Phase 2 Création de modèle	61
3.6.	Phase 3 Prédiction (Résultat)	61
4.	Expérience	62
	4.1. Cadre expérimental	62
	4.2. Mesure d'évaluation	62
	4.3. Bon ajustement	62
5.	Résultats principaux	63
6.	Conclusion	65
	Conclusion Générale et Perspective	66

Liste des figures

Figure 1 : Processus de la catégorisation de textes [Jalam, 2003]	7
Figure 2 : Mesures d'évaluation de Rappel, Précision et Exactitude	10
Figure 3 : Présente La Pluridisciplinaire De TAL	17
Figure 4 : Les niveaux de traitement	18
Figure 5 : Répartition géographique de la langue arabe	20
Figure 6 : Classification des unités lexicales	26
Figure 7 : Architecture globale du Système (Izzat Alsmadi & Mohammad Zarour, 2017)	28
Figure 8 : Le déroulement logique de l'approche proposée. (Saqib Hakak, Amirrudin Kamsin, Shivakumara Palaiahnakote, Omar Tayan, Mohd.Yamani Idna Idris & Khir Zuhaili Abukhir, 2018)	29
Figure 9 : Fonctionnement de l'algorithme KNN	36
Figure 10 : Exemple d'arbre de décision	38
Figure 11 : Fonctionnement de l'algorithme d'arbre de décision	39
Figure 12 : Vecteurs de support machines	40
Figure 13 : Réseau de neurones récurrent [29]	50
Figure 14 : Module répété d'un LSTM, avec les 4 neurones (portes) en jaune [27]	50
Figure 15 : Le site web « TANZIL »	54
Figure 16 : Un extrait du fichier du jeu de données utilisé (data sets)	56
Figure 17 : L'environnement « Google Colab » - Création NOUVEAU NOTEBOOK -	58
Figure 18 : Mode d'exécution	58
Figure 19 : environnement d'exécution « Google Colab »	59
Figure 20 : les différentes bibliothèques utilisées dans notre implémentation	59
Figure 21 : Statistique du Data sets	60

Figure 22 : Prédiction des phrases	61
Figure 23 : les mesures d'évaluation	62
Figure 24 : Les statistiques de notre modèle	63

Liste des Tables

Tableau 1 : Kappa Accord	11
Tableau 2: Dérivation de plusieurs mots à partir de la racine « كتب , écrire »	22
Tableau 3 : Catégories Principales de āyāt arabe quranique (verses)	30
Tableau 4 : Le nombre d'enregistrements audio pour chaque règle (indiquant à la fois l'utilisation correcte et incorrecte de cette règle).	31
Tableau 5 : les statistiques de notre expérience sur les deux jeux de Données (data_with_tashkeel, data_without_tashkeel)	62
Tableau 6 : Les principaux résultats pour l'étude comparative	64

Introduction Générale

Tout d'abord, le terme "intelligence artificielle" ou "IA" a été inventé à la conférence de Dartmouth en 1956. La définition la plus générale est le test de Turing, proposée pour la première fois en 1950, selon lequel une machine peut communiquer en langage naturel via un téléscripteur trompe une personne en lui faisant croire qu'il s'agit d'un être humain. "AGI" ou "intelligence générale artificielle", étend cette idée en exigeant des machines qu'elles fassent tout ce que les humains peuvent faire : comprendre les images, naviguer dans un robot, reconnaître les expressions faciales, y répondre adéquatement, distinguer les genres musicaux...etc (Mahoney, 2015).

L'apprentissage automatique et l'apprentissage en profondeur ce sont deux notions de base dans l'intelligence artificielle. L'apprentissage automatique est la science qui vise à faire en sorte que les machines se comportent comme des humains sans programmation.

En ce qui concerne L'apprentissage en profondeur est un sous-champ de l'apprentissage automatique concernant les algorithmes inspirés de la structure et de la fonction du cerveau, appelé réseau neuronal artificiel.

Traitement du langage naturel (PNL), c'est un domaine qui concerne les interactions entre les ordinateurs et les langages humains (naturels), la plupart des PNL sont basés sur l'apprentissage en profondeur, l'apprentissage automatique classique ainsi que sur d'autres IA moins connues. Au-delà de l'intelligence artificielle, le domaine s'inspire également d'idées tirées de la linguistique informatique.

Notre travail, qui se veut une contribution à ce projet, consiste en la catégorisation automatique des textes arabes, particulièrement des versets coraniques.

Pour ce faire, nous nous appuyons sur un corpus de textes arabes et coraniques mis à notre disposition par le projet TANZIL, dans un but de classification selon leurs différentes.

L'utilisation de Google Colab, prescrit dans le cadre de nos travaux, a abouti à des résultats satisfaisants.

Notre mémoire est ainsi organisé :

- ✓ Un premier chapitre intitulé « **Catégorisation et classification des textes** », nous faisons un bref tour d'horizon sur la classification des textes de manière générale.
- ✓ Dans le deuxième chapitre intitulé « **Traitement automatique de la langue arabe** », nous donnons un aperçu sur la langue arabe et exposons les différents aspects de sa morphologie ainsi que ses caractéristiques.
- ✓ Puis, en troisième chapitre : « **Les Classificateurs** », Nous présentons les principales techniques de classification automatique supervisées.
- ✓ En fin, le quatrième chapitre : « **Expérimentations et Implémentation** », se prêtera à exposer les expérimentations et les résultats obtenus ainsi que l'implémentation de prédiction de catégorisation.

Chapitre 1

Catégorisation et

Classification des

textes

1. Introduction

Dans ce chapitre nous allons exposer la classification automatique de texte, plus en détail la catégorisation de textes. Nous présentons quelques définitions sur la classification et les différents jeux de mots utilisés : classification, catégorisation ou clustering, ensuite les différents objectifs et intérêts de la classification ainsi que les conflits avec d'autres disciplines comme la Recherche d'Informations, puis nous décrivons le processus général de la catégorisation de textes avec toutes ces étapes, et enfin les problèmes spécifiques aux textes lors de l'apprentissage automatique.

2. Définition

La classification de texte (C.T) consiste à trouver /construire un lien fonctionnel entre un ensemble de texte et un ensemble de catégories (étiquettes, classes). Ce lien fonctionnel est estimé par apprentissage automatique (méthode d'apprentissage automatique), également appelé modèle de prédiction. À cette fin, il doit y avoir un ensemble de texte préalablement étiquetés, appelé ensemble d'apprentissage, à partir duquel nous pouvons estimer les paramètres du modèle de prédiction le plus efficace, c'est-à-dire le modèle qui génère le modèle. La moindre erreur de prédiction.

Formellement, la catégorisation de textes consiste à associer une valeur booléenne à chaque paire $(d_j, c_i) \in D \times C$, où "D" est l'ensemble des textes et "C" est l'ensemble des catégories. La valeur V (Vrai) est alors associée au couple (d_j, c_i) si le texte d_j appartient à la classe c_i tandis que la valeur F (Faux) lui sera associée dans le cas contraire. Le but de la catégorisation de textes est de construire une procédure (modèle, classificateur) $\Phi : D \times C \rightarrow \{V, F\}$ qui associe une ou plusieurs catégories à un document d_j tel que la décision donnée par cette procédure « coïncide le plus possible » avec la fonction $\Phi : D \times C \rightarrow \{V, F\}$, la vraie fonction qui retourne pour chaque vecteur d_j une valeur c_i (Sebastiani, 2002) [1].

3. Types de la classification automatique

La classification automatique consiste à regrouper divers objets (les individus) en sous-ensembles d'objets (les classes). Elle peut être supervisée où les classes sont connues a priori, elles ont en général une sémantique associée ou bien non-supervisée (en anglais Clustering) où les classes sont fondées sur la structure des objets, la sémantique associée aux classes est plus difficile à déterminer.

3.1. La classification supervisée (Catégorisation)

La classification est dite supervisée lorsque les données qui entrent dans le processus sont déjà catégorisées et que les algorithmes doivent s'en servir pour prédire un résultat.

Plusieurs techniques sont utilisées. On peut citer Naïve bayes, Machine à vecteur de support, K voisins Proches, Arbre de Décision...

3.2. La classification non supervisé (Clustering)

L'apprentissage non supervisé est principalement utilisé en matière de "clusterisation" procédé destiné à regrouper un ensemble d'éléments hétérogènes sous forme de sous-groupes homogènes ou liés par des caractéristiques communes. La machine fait alors elle-même les rapprochements en fonction de ces caractéristiques qu'elle est en mesure de repérer sans nécessiter d'intervention externe. [2]

3.3. Avantages et inconvénients

Parmi les avantages et inconvénients liés aux deux approches, on peut citer :

- Les groupes ou clusters obtenus par la technique supervisée est de meilleure qualité et plus précise que la technique non-supervisée.
- Dans la technique supervisée, on sait ce qui est attendu favorisant de meilleurs résultats par rapport au non supervisée.
- Un avantage des techniques non supervisées, est qu'elles accomplissent la tâche de similarité sans avoir besoin des données expertisées.
- Un inconvénient des approches supervisées, repose sur le fait qu'il peut être difficile de se procurer des données expertisées.
- L'inconvénient majeur des approches non supervisées qu'elle demande dans l'étape d'évaluation des résultats l'intervention d'un expert.

4. Processus de la catégorisation automatique

Le processus de catégorisation intègre la construction d'un modèle de prédiction qui en entrée, reçoit un texte et, en sortie, lui associe une ou plusieurs étiquettes.

La figure 1 résume le processus de catégorisation des textes qui comporte deux phases : l'apprentissage et le classement.

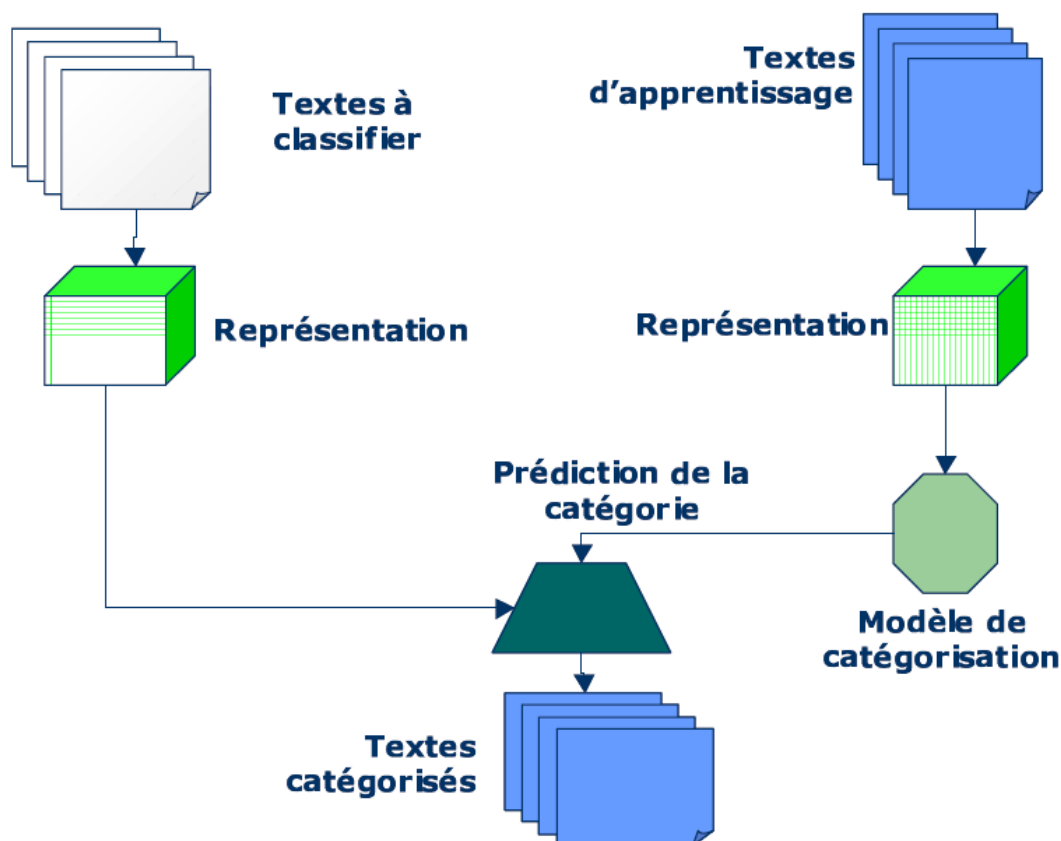


Figure 1 : Processus de la catégorisation de textes [Jalam, 2003]

Pour identifier la catégorie ou la classe à laquelle un texte est associé, un ensemble d'étapes est habituellement suivi. Ces étapes concernent principalement la manière dont un texte est représenté, le choix de l'algorithme d'apprentissage à utiliser et comment évaluer les résultats obtenus pour garantir une bonne généralisation du modèle appris [3].

4.1. Représentation des textes

La représentation des textes est une étape très importante dans le processus de C.T. Pour cela, il est nécessaire d'utiliser une technique de représentation efficace permettant de représenter les textes sous une forme exploitable par la machine.

La représentation la plus couramment utilisée est celle du modèle vectoriel dans laquelle chaque texte est représenté par un vecteur de n termes pondérés.

Les différentes méthodes qui existent pour la représentation des textes sont :

4.1.1. Représentation en sac de mots (Bag of Words)

Cette représentation des textes est la plus simple. Elle a été introduite dans le cadre du modèle vectoriel. Les textes sont transformés simplement en vecteurs dont chaque composante représente un terme.

Dans un premier temps, les termes sont les mots qui constituent un texte. Dans les langues comme le français ou l'anglais, les mots sont séparés par des espaces ou des signes de ponctuations ; ces derniers, tout comme les chiffres, sont supprimés de la représentation. Mais il n'est pas aussi facile de délimiter les mots dans certaines autres langues telles que l'Arabe qui est écrit de droite à gauche ou le Mandarin où les mots ne sont pas séparés par des espaces.

4.1.2. Représentation des textes avec des racines lexicales

Cette méthode consiste à remplacer les mots du document par leurs racines lexicales, et à regrouper les mots de la même racine dans une seule composante. Ainsi, plusieurs mots du document seront remplacés par la même racine. Plusieurs algorithmes ont été proposés. On peut citer l'algorithme de Porter [Porter 1980] et l'algorithme de Khodja pour la langue arabe. Ces algorithmes font la normalisation de mots qui sert à supprimer les affixes de ces derniers pour obtenir une forme canonique. Néanmoins la transformation automatique d'un mot à sa racine lexicale peut engendrer certaines anomalies.

En effet, une racine peut être commune pour des mots qui portent des sens différents tel que les mots jour, journalier, journée ont la même racine « jour » mais font référence à trois notions différentes, cette représentation dépend également de la langue utilisée.

4.1.3. Représentation des textes avec des lemmes

La lemmatisation consiste à utiliser l'analyse grammaticale afin de remplacer les verbes par leur forme infinitive et les noms par leur forme au singulier. La lemmatisation est donc plus compliquée à mettre en œuvre que la recherche de racines, puisqu'elle nécessite une analyse grammaticale des textes.

4.1.4. Représentation des textes avec des N-gramme

Cette méthode consiste à représenter le document par des n-grammes. Le n-gramme est une séquence de n caractères consécutifs. Elle consiste à découper le texte en

plusieurs séquences de n caractères en se déplaçant avec une fenêtre d'un caractère. Un n -gramme de taille 1 est appelé uni-gramme, de taille 2 est un bi-gramme et la taille 3 est un trigramme. Cette technique présente plusieurs avantages. Les n -grammes capturent automatiquement les racines des mots les plus fréquents sans passer par l'étape de recherche des racines lexicales, celles-ci, indépendantes de la langue, les espaces sont prises en considération. En effet, la non-prise en compte de ces dernières introduit du bruit.

4.1.5. Représentation des textes par des phrases

Un certain nombre de chercheurs proposent d'utiliser les phrases comme unité de représentation au lieu des mots comme c'est le cas dans la représentation «sac de mot », puisque les phrases sont plus informatives que les mots seuls, par exemple : « recherche d'information », « world wide web », ont un plus petit degré d'ambiguïté que les mots constitutifs, mais aussi parce que les phrases ont l'avantage de conserver l'information relative à la position du mot dans la phrase [4].

4.2. Choix des classificateurs

La catégorisation de textes comporte un choix de technique d'apprentissage (ou classificateur). Parmi les méthodes d'apprentissage les plus souvent utilisées figurent : "Naïve bayes", "Machine à vecteur de support", "K voisins Proches", "arbre de Décision",...

Généralement, le choix du classificateur se fait en fonction de l'objectif final à atteindre. Si l'objectif final est, par exemple, de fournir une explication ou une justification qui sera ensuite présentée à un décideur ou un expert, alors on préférera les méthodes qui produisent des modèles compréhensibles tels que les arbres de décision.

Mais il demeure difficile de remplacer les tests pour savoir quel classificateur est adéquat à quelle situation.

4.3. Evaluation de la qualité des classificateurs

Il existe de nombreuses mesures pour calculer la performance d'un classificateur.

Les mesures de rappel et précision : Initialement elles ont été conçues pour les systèmes de recherche d'information, que la communauté de classification de textes à

adoptées par la suite. Formellement, pour chaque classe C_i , on calcule deux probabilités qui peuvent être estimées à partir de la matrice de contingence correspondante, ainsi ces deux mesures peuvent être définies de la manière suivante :

- **Le Rappel (R)** ou **Recall** en anglais, est la Proportion des solutions pertinentes trouvées. Il mesure la capacité du système à donner toutes les solutions pertinentes.

$$R = VP / (VP + FN)$$

- **La Précision (P)** est la Proportion de solutions trouvées qui sont pertinentes. Elle mesure la capacité du système à refuser les solutions non-pertinentes

$$P = VP / (VP + FP)$$

- ❖ **VP (True Positive)** : le nombre de documents correctement attribués à la catégorie.
- ❖ **FP: (False Positive)** : le nombre de documents incorrectement attribués à la catégorie.
- ❖ **FN (False Negative)**: le nombre de documents qui auraient dû lui être attribués mais qui ne l'ont pas été.

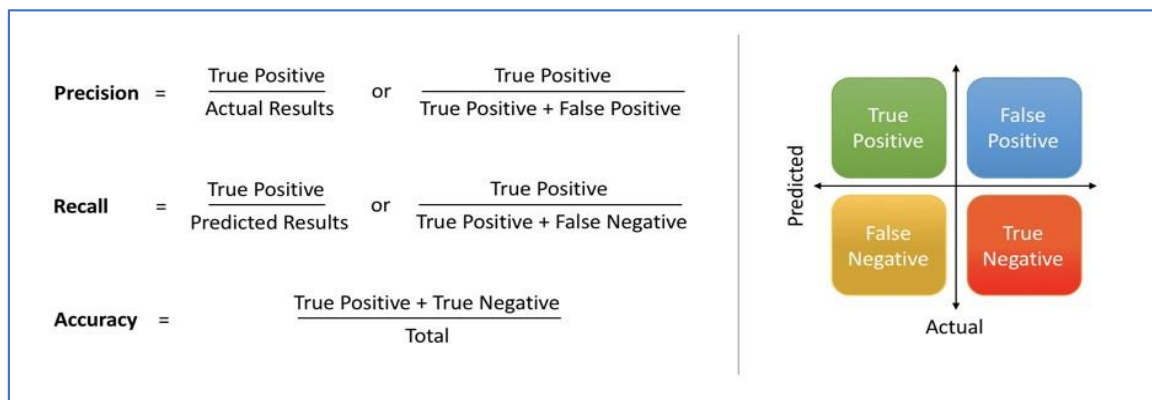


Figure 2 : Mesures d'évaluation de Rappel, Précision et Exactitude

D'autres mesures sont également couramment utilisées. On y a fait appel dans nos expériences afin d'évaluer la performance des différents classificateurs utilisés dans notre tâche de classification des Poèmes arabes selon leurs époques d'apparition. On peut alors citer :

- **Exactitude (Accuracy) :**

$$\text{Exactitude} = (\text{VP} + \text{TN}) / (\text{VP} + \text{VN} + \text{FP} + \text{FN})$$

- **F-Mesure (F) :** la moyenne harmonique entre le Rappel et la précision :

$$F = 2 * (\text{P} * \text{R}) / (\text{P} + \text{R})$$

- Le coefficient de **KAPPA (K) :** est un indice statistique variant entre 0 et 1 interprété comme suit :

Kappa (K)	Interprétation
< 0	Désaccord
0.0 — 0.20	Accord faible
0.21 — 0.40	Accord juste
0.41 — 0.60	Accord modéré
0.61 — 0.80	Accord fort
0.81 — 1.00	Accord Presque parfait

Tableau 1 : Kappa Accord

- **Matrice de confusion:** En apprentissage automatique supervisé, la matrice de confusion est une matrice qui mesure la qualité d'un système de classification. Chaque ligne correspond à une classe réelle, chaque colonne correspond à une classe estimée. La cellule ligne L, colonne C contient le nombre d'éléments de la classe réelle L qui ont été estimés comme appartenant à la classe C.

L'un des intérêts de la matrice de confusion est qu'elle montre rapidement si un système de classification parvient à classifier correctement [5].

5. Les applications de la catégorisation des textes

La catégorisation de textes est utilisée dans de nombreuses applications. Parmi lesquelles figurent : l'identification de la langue, la catégorisation de documents multimédia, la classification et la reconnaissance d'écrivains et des poèmes...

Notre travail traite de la catégorisation des textes arabes c'est de coran ou non.

6. Les problèmes de la classification des textes

Plusieurs difficultés peuvent contrarier le processus de catégorisation de textes, les principales sont les suivantes :

- a. **La redondance** : La redondance et la synonymie permettent d'exprimer le même concept par des expressions différentes, soit plusieurs façons d'exprimer la même chose. Cette difficulté est liée à la nature des documents traités exprimés en langage naturel contrairement aux données numériques. "Lefèvre" illustre cette difficulté dans l'exemple du chat et l'oiseau : mon chat mange un oiseau, mon gros matou croque un piaf et mon félin préféré dévore une petite bête à plumes (Lefèvre, 2000). La même idée est représentée de trois manières différentes, différents termes sont utilisés d'une expression à une autre mais en fin de compte c'est bien le malheureux oiseau qui est dévoré par le chat.
- b. **L'ambiguïté** : A la différence des données numériques, les données textuelles sont sémantiquement riches, du fait qu'elles sont conçues et raisonnées par la pensée humaine. À cause de l'ambiguïté, les mots sont parfois de mauvais descripteurs ; par exemple le mot avocat peut désigner le fruit, le juriste, ou même au sens figuré, la personne qui défend une cause.
- c. **La graphie** : Un terme peut comporter des fautes d'orthographe ou de frappe comme il peut s'écrire de plusieurs manières ou s'écrire avec une majuscule. Ce qui va peser sur la qualité des résultats. Car si un terme est orthographié de deux manières dans le même document (Ghelizane/Relizane, Oignon/Ognon, Clé/Clef, feignant/fainéant), la simple recherche de ce terme avec une seule forme graphique omet la présence du même terme sous d'autres graphies.

- d. **Complexité de l'algorithme d'apprentissage** : Un texte est représenté généralement sous forme de vecteur contenant les nombres d'apparitions des termes dans ce texte. Or, le nombre de textes à traiter est très important. A cela, s'ajoute le nombre de termes composant le même texte. On peut dès lors se faire une idée de la dimension du tableau (textes * termes) à traiter qui ne peut que considérablement compliquer la tâche de classification en diminuant la performance du système.
- e. **Présence-Absence de terme** : La présence d'un mot dans le texte indique un propos que l'auteur a voulu exprimer. Il y a donc une relation impliquant le mot et le concept associé, sachant très bien qu'il y a plusieurs façons d'exprimer la même chose. Dès lors l'absence d'un mot n'implique pas obligatoirement que le concept qui lui est associé est absent du document. Cette réflexion pointue nous amène à être attentifs quant à l'utilisation des techniques d'apprentissage se basant sur l'exclusion d'un mot particulier.
- f. **Les mots composés** : La non-prise en charge des mots composés tels que Arc-en-ciel, peut-être, sauve-qui-peut, etc..., dont le nombre est très important dans toutes les langues, et traiter le mot Arc-en-ciel par exemple en étant 3 termes séparés réduit considérablement la performance d'un système de classification néanmoins l'utilisation de la technique des n-grammes pour le codage des textes atténue considérablement ce problème des mots composés [6].

7. Conclusion

Dans ce chapitre nous avons fait un bref tour d'horizon sur la classification des textes de manière générale tout en citant les différents types et le processus détaillé de la catégorisation automatique. Nous avons également introduit les différents moyens d'évaluation d'un classificateur ainsi que les problèmes majeurs et difficultés qui peuvent s'opposer à cette dite classification. La catégorisation de texte a essentiellement progressé ces dix dernières années grâce à l'introduction des techniques héritées de l'apprentissage automatique qui ont amélioré de manière significative les taux de bonne classification.

Le chapitre suivant présente le domaine du Traitement automatique des langages naturels et plus précisément celui de la langue arabe.

Chapitre 2

Traitement

Automatique de la

Langue Arabe

1. Introduction

La langue est un outil central dans notre vie sociale et professionnelle.

Il s'agit d'un support pour véhiculer, entre autres, des idées, des opinions et des sentiments ainsi que pour persuader, demander des informations, donner des ordres, etc.

L'intérêt pour la langue du point de vue de l'informatique a débuté au début même de l'informatique, notamment dans le cadre des travaux dans le domaine de l'intelligence artificielle ; on assiste alors à la naissance du TAL.

La vague d'Internet entre le milieu des années 1990 et le début des années 2000 a été un moteur très important pour le TAL et les domaines dérivés, notamment celui de la recherche d'information (RI) et de la classification qui sont passés d'un domaine marginal et limité au seul domaine de la grande entreprise, à la recherche d'information à l'échelle d'Internet, dont le contenu ne cesse de s'élargir [7].

Au cours de ce chapitre nous présenterons d'une manière brève le TAL et le TALA tout en décrivant les particularités de la langue arabe ainsi que certaines de ses propriétés morphologiques et syntaxiques.

2. Traitement Automatique De La Langue (TAL)

Le Traitement Automatique de la langue naturelle (TALN) ou des langues (TAL) est une discipline à la frontière de la linguistique, de l'informatique et de l'intelligence artificielle.

Elle concerne la conception de systèmes et techniques informatiques permettant de manipuler le langage humain, dont le principal objectif est la conception et le développement de programmes capables de traiter de manière automatique des données linguistiques c'est-à-dire des données exprimées dans une langue dite naturelle.

Ces dernières décennies le traitement automatique des langues a connu une véritable ascension que ce soit sur le plan scientifique mais aussi socio-économique et cela par l'émergence de plusieurs firmes et de produits spécialisés. On parle aujourd'hui : de Traduction automatique, de correction automatique d'orthographe, de résumé automatique, d'interrogation de base de données en langues naturelle,etc.

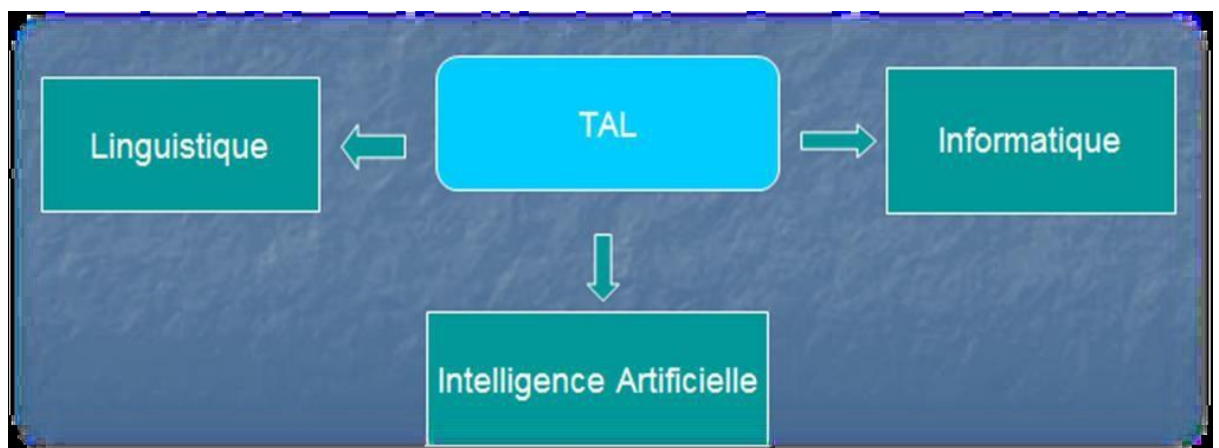


Figure 3 : Présente La Pluridisciplinaire De TAL

La réalisation de n'importe quelle application parmi celles citées précédemment passe principalement par différents niveaux (lexicale, morphologique, syntaxique, sémantique et pragmatique) mais aussi par le développement de plusieurs modules importants, où la réussite de l'application dépend pleinement de la performance de ces modules [8].

3. Niveaux Traitement Automatique de la Langue

On va essayer de citer brièvement dans cette section les différents niveaux de traitement nécessaires pour parvenir à une compréhension complète d'un énoncé en langage naturel.

La figure 4 schématise ces différents niveaux de traitements. Ces niveaux se superposent ; chacun apportant des problèmes spécifiques à résoudre relatifs à un niveau donné. En s'appuyant sur un découpage méthodologique classique dans le domaine de la linguistique cela nous donne la hiérarchie suivante :

- ✓ **La phonétique** concerne l'étude des sons et prosodies (variations).
- ✓ **La phonologie** concerne l'étude de Phonèmes.
- ✓ **La morphologie** concerne l'étude de la formation des mots et de leurs variations de forme.
- ✓ **La syntaxe** consistant à extraire les relations grammaticales que les mots et groupes de mots entretiennent entre eux.
- ✓ **La sémantique** se consacre au sens des énoncés.
- ✓ **La pragmatique** prend en compte le contexte d'énonciation.

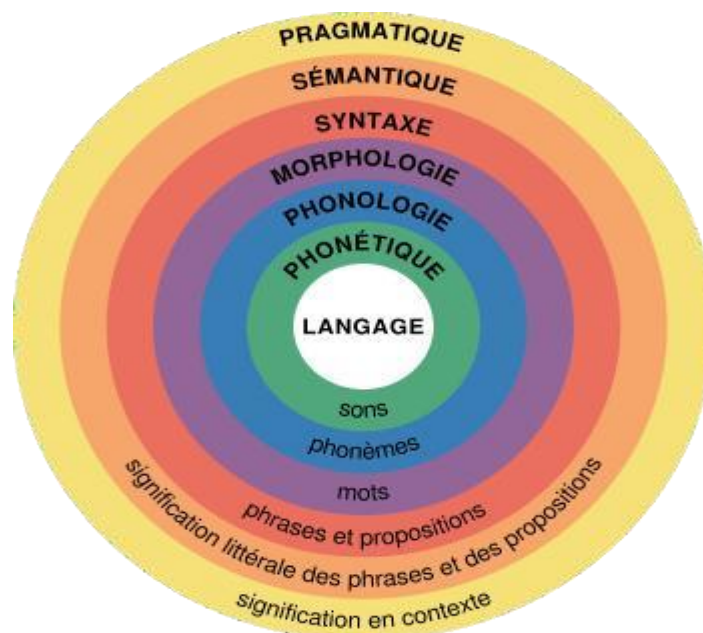


Figure 4 : Les niveaux de traitement

4. Traitement Automatique De La Langue Arabe (TALA):

Le traitement automatique de la langue arabe est une discipline en pleine expansion, et dans laquelle on constate de plus en plus de recherches et de technologies qui portent un intérêt aux spécificités de cette langue et proposent des outils nécessaires au développement de son traitement automatique.

Le traitement automatique de l'arabe est un domaine de recherche stimulant. Il combine en effet plusieurs défis intéressants, parmi lesquels on peut citer la complexité morphologique de la langue, son haut degré d'ambiguïté et l'existence de nombreux dialectes présentant des variantes significatives.

Par ses propriétés morphologiques et syntaxiques la langue arabe est considérée Comme une langue difficile à maîtriser dans le domaine du traitement automatique de la langue.

Les recherches pour le traitement automatique de l'arabe ont débuté vers les années 1970. Les premiers travaux concernaient notamment les lexiques et la morphologie [9].

Avec la diffusion de la langue arabe sur le Web et la disponibilité des moyens de manipulation de textes arabes, les travaux de recherche ont abordé des problématiques plus variées comme la syntaxe, la traduction automatique, l'indexation automatique des documents, la recherche d'information, la catégorisation des textes etc.

Le domaine du Traitement Automatique des Langues (TAL) appliqué à l'arabe a fait ces 15 dernières années des progrès considérables, mais il reste un grand chemin à faire pour pouvoir rivaliser avec d'autres langues comme le français et l'anglais.

5. La langue arabe :

L'arabe (arabe: العربية, al-'arabīya ') est une langue asiatique et africaine de la famille des langues sémitiques. Dans le monde arabe et la diaspora arabe, on estime que 315 à 375 millions de personnes parlent. L'arabe est de loin la langue sémitique la plus utilisée.

L'arabe est originaire de la péninsule arabique et est devenu la langue du Coran et la langue rituelle de l'islam au septième siècle. L'expansion territoriale de l'Empire arabe au Moyen Âge a conduit à une arabisation au moins partielle à plus ou moins long terme au Moyen-Orient, en Afrique du Nord et dans certaines régions d'Europe

(péninsule ibérique, Sicile). , La Crète, Chypre (une zone disparue) et Malte (Malte est une extension).

Cette langue a été parlée pour la première fois par les Arabes. Cette langue est géographiquement répartie sur de nombreux continents, d'un point de vue sociologique pour les non-arabes, et est aujourd'hui devenue l'une des langues les plus répandues au monde. C'est la langue officielle de plus de vingt pays et de plusieurs organisations internationales, dont l'une des six langues officielles des Nations Unies. [10].

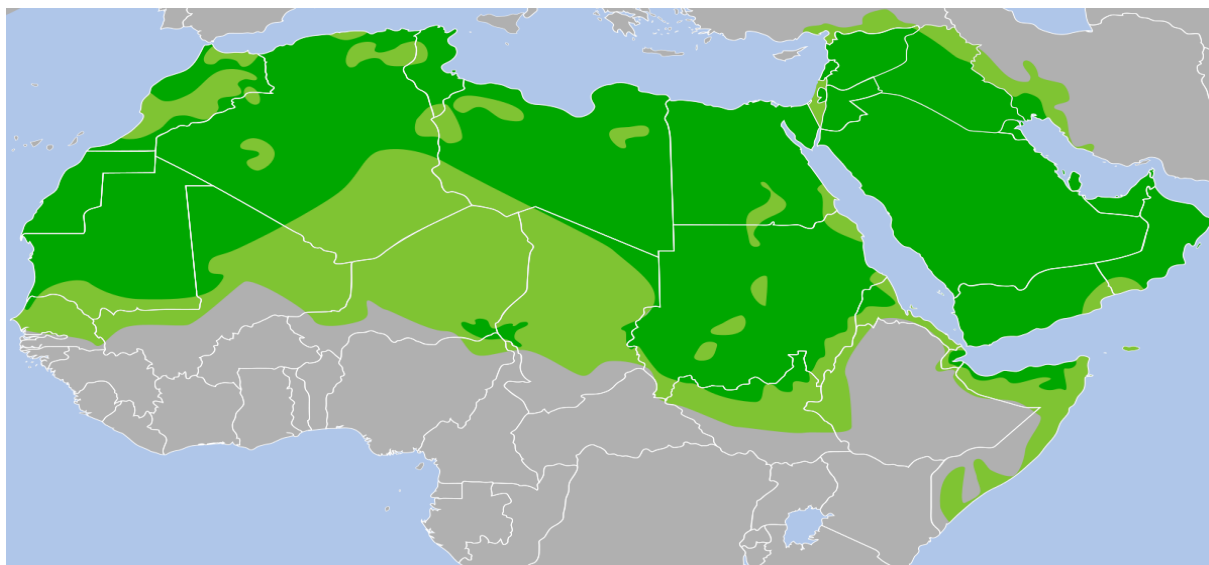


Figure 5 : Répartition géographique de la langue arabe

5.1. Particularité de la langue arabe:

L'alphabet de la langue arabe compte 28 consonnes. L'arabe s'écrit et se lit de droite à gauche, les lettres changent de forme de présentation selon leur position (au début, au milieu ou à la fin du mot) [11]. L'arabe est assez complexe en raison de la variation morphologique et du phénomène d'agglutinement; on peut citer brièvement quelques particularités de cette langue :

a. Les voyelles :

En arabe, la notion de voyelles n'existe pas sous sa forme classique : en effet elles ne sont pas des lettres de l'alphabet, mais représentées par des signes diacritiques placés facultativement au-dessus ou au-dessous des consonnes et qui jouent le même rôle que les voyelles dans les autres langues. Les voyelles en arabe sont généralement utilisées

pour faciliter la lecture ou pour rendre un texte beaucoup moins ambigu, elles permettent de distinguer des traits flexionnels tels que le genre, le nombre, la personne, etc.

Pour cette raison les textes religieux, les ouvrages pédagogiques ainsi que les textes juridiques sont entièrement en diacritiques.

b. Les agglutinations :

Contrairement aux langues latines, en arabe, les articles, les prépositions, les pronoms, etc. collent aux adjectifs, noms, verbes et particules auxquels ils se rapportent. Comparé au français, un mot arabe peut parfois correspondre à une phrase en française.

Exemple : le mot arabe « أتتذكروننا » correspond en Français à la phrase "Est-ce que vous vous souvenez de nous ?".

c. Irrégularité de l'ordre des mots dans la phrase :

L'ordre des mots en arabe est relativement libre. D'une manière générale, on met au début de la phrase le mot sur lequel on veut attirer l'attention et l'on termine sur le terme le plus long ou le plus riche en sens ou en sonorité. Cet ordre provoque des ambiguïtés syntaxiques artificielles dans la mesure où il faut prévoir dans la grammaire toutes les règles de combinaisons possibles d'inversion de l'ordre des mots dans la phrase.

Exemple, on peut changer l'ordre des mots dans la phrase suivante :

فازت الجزائر بكأس افريقيا

الجزائر فازت بكأس افريقيا

بكأس افريقيا فازت الجزائر

On peut constater que les trois phrases ont le même sens, ce malgré le changement dans l'ordre de ces mots.

5.2.Morphologie arabe

La langue arabe, par rapport aux autres langues, à une morphologie riche et différente L'analyse morphologique d'un mot, consiste principalement à déterminer la structure générale de ce mot, les éléments essentiels utilisés pour construire ce mot sont :

✓ Les racines :

Les racines sont des verbes formés souvent de trois consonnes (Mustafa et al. 2008)[12]. Elles sont à l'origine de la plupart des mots arabes. A partir d'une racine, on peut générer jusqu'à 30 mots.

Considérons l'exemple de la racine trigramme

(«كتب» «écrire») où l'on peut produire plusieurs nominaux et verbaux

Ecrire	Ecrivain	Livre	Petit livre	Ecrit
كتب	كاتب	كتاب	كتيب	مكتوب

Tableau 2: Dérivation de plusieurs mots à partir de la racine «كتب, écrire»

Dans cet exemple, nous remarquons qu'à partir d'une racine trilittérale «كتب», on peut générer plusieurs mots dans lesquels les trois lettres (ك, ت, ب) figurent, ainsi que d'autres lettres représentant les patrons insérées au début, au milieu ou à la fin du mot.

✓ Les patrons :

Les patrons (ou modèles) sont des déclinaisons du mot «فعل» qui sont obtenus en ajoutant des affixes ou en utilisant des diacritiques. Par exemple, le modèle «مستفعل» est obtenu en y ajoutant les préfixes, par contre le modèle «فعل» est obtenu en utilisant les diacritiques. Les patrons servent à extraire la racine d'un mot ou inversement à produire des stems à partir d'une racine (Khoja et al. 2001) [13].

✓ Les affixes :

Les affixes sont des morphèmes qui s'ajoutent au début ou à la fin des mots arabes. En général, Ils permettent de former, à partir d'une même racine, de nouveaux lemmes. Les affixes peuvent être subdivisés en deux types : préfixes et suffixes. Les préfixes se placent avant le radical, et dépendent des mots auxquels ils s'attachent. Il y a trois types des préfixes: les préfixes nominaux qui sont réservés pour les noms et les adjectifs, les préfixes verbaux qui sont réservés aux verbes et les préfixes généraux qui sont indépendants du type des mots.

Les suffixes sont des morphèmes placés après le radical. Il existe deux types des suffixes: les suffixes verbaux qui dépendent de la transitivité, et les suffixes nominaux indiquant la flexion du nom, du nombre et du genre, etc.

✓ **Les stems :**

Un stem (ou lemme) est obtenu par troncature sur les deux extrémités du mot sans modification interne sur le mot. C'est la dérivation obtenue à partir d'une racine donnée selon un patron. Par exemple, le lemme « مدرس, enseignant », il est obtenu à partir de la racine « درس, il a étudié » selon le patron « مفعَل ».

✓ **Les mots dérivés :**

La plupart des mots arabes sont considérés comme des mots dérivés, puisqu'ils sont construits à partir des racines. Ainsi, les mots qui dérivent d'une même racine ont des sens différents. En effet, les mots dérivés sont construits à partir d'un stem en y ajoutant des affixes comme c'est le cas du nom « أتطلبون, est ce que vous demandez ? »

5.3. Structure d'un mot

Les mots peuvent avoir une structure composée, résultat d'une agglutination de morphèmes lexicaux et grammaticaux. En arabe un mot peut représenter toute une proposition. La représentation suivante schématise une structure possible de mot complexe.



- Les proclitiques sont des prépositions ou des conjonctions.
- Les préfixes et suffixes expriment des traits grammaticaux, tels que les fonctions de noms, le mode du verbe, le nombre, le genre, la personne.....
- Les enclitiques sont des pronoms personnels.
- Le corps schématique représente la base de mot.

Exemple :

أتتذكروننا → Ce mot en arabe représente en français la phrase suivante :

« Est-ce que vous vous souvenez de nous ? »

- Proclitique : أ conjonctions d'interrogation.
- Préfixe : "ت" préfixe verbal exprimant l'aspect inaccompli.
- Corps schématique : تتذكر dérivé de la racine (ذ ك ر) selon le schème تفعل

- Suffixe : "ون" suffixe verbal exprimant le pluriel.
- Enclitique : "نا" pronom suffixe.

Cet exemple montre la richesse morphologique de la langue arabe [14]. Pour identifier les différentes formes soudées par ces phénomènes d'agglutination, et envisager un traitement automatique, il va donc falloir mettre en œuvre une phase spécifique de segmentation.

5.4. Catégorie d'un mot

En langue arabe, le mot peut être divisé en trois catégories : le nom, le verbe et les particules. La figure 06 résume toutes ces catégories :

- **Verbe :**

Un verbe est une entité exprimant un sens dépendant du temps, c'est un élément fondamental auquel se rattachent directement ou indirectement les divers mots qui constituent l'ensemble.

La plupart des mots en arabe, dérivent d'un verbe de trois lettres. Chaque verbe est donc la racine d'une famille de mots. Comme en français, le mot en arabe se déduit de la racine en rajoutant des suffixes ou des préfixes [15].

La conjugaison des verbes dépend de plusieurs facteurs :

- Le temps (accompli, inaccompli).
- Le nombre du sujet (singulier, duel, pluriel).
- Le genre du sujet (masculin, féminin).
- La personne (première, deuxième et troisième).
- Le mode (actif, passif).

- **Nom :**

L'élément désignant un être ou un objet qui exprime un sens indépendant du temps.

Les noms arabes sont de deux catégories, ceux qui sont dérivés de la racine verbale et ceux qui ne le sont pas comme les noms propres, les noms communs. La déclinaison des noms se fait selon les règles suivantes :

- Le féminin singulier: On ajoute le ة, exemple كبير grand devient كبيرة grande.

- Le féminin pluriel : On ajoute pour le pluriel les deux lettres ات. Exemple : كبير grand devient كبيرات .grandes
- Le masculin pluriel : Pour le pluriel masculin on rajoute les deux lettres ين ou ون dépendamment de la position du mot dans la phrase.

Exemple : الراجعين ou الراجعون revenants devient الراجع الراجع

- Le Pluriel brisé: Il suit une diversité de règles complexes et dépend du nom.

Exemple: طفل un enfant devient أطفال des enfants.

- **Les particules :**

Entités qui servent à situer les événements et les objets par rapport au temps et l'espace, et permettent un enchaînement cohérent du texte.

Ce sont principalement les mots vides comme les prépositions (حتى، من ، في، على) et les conjonctions (بل، أم، أو، ثم) qui sont utilisés pour lier des noms, des verbes ou des phrases (Taani et al. 2009) [16].

Par exemple :

- La particule « حتى » est employée pour indiquer une action progressive et sa finalité.

Exemple "قرأت الكتاب حتى الخاتمة" : j'ai lu le livre jusqu'à la fin".

Généralement, on dit que les particules sont des termes à ne pas prendre en considération lors du calcul de fréquence de distribution des mots.

Dans notre travail on va démontrer que ce n'est pas toujours le cas. En effet, ces particules peuvent être très utiles, surtout dans les travaux de classifications.

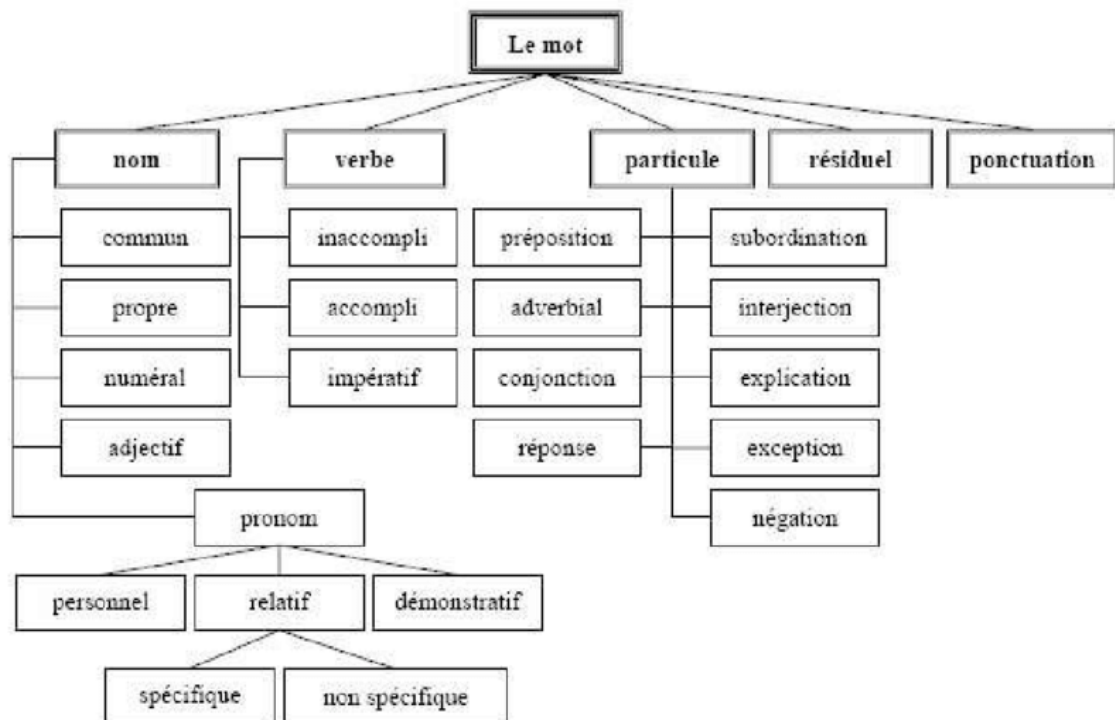


Figure 6 : Classification des unités lexicales

6. Difficultés du traitement automatique de la langue arabe :

Parmi les problèmes spécifiques à la langue arabe et à certaines autres langues sémitiques, citons l'ambiguïté, l'absence des voyelles, la segmentation de textes, les problèmes de flexion et d'agglutination.

6.1. Ambiguïté :

Les mots arabes peuvent être ambigus aux niveaux lexical et grammatical.

Exemples : Le mot « ذهب » est ambigu lexicalement. Il peut désigner l'or en français ou encore le verbe aller [17].

Le mot « كتب » quant à lui, est ambigu grammaticalement. Il peut appartenir à plusieurs catégories grammaticales différentes : verbe ou nom. Le sens de ce mot sera très différent selon sa catégorie : nom = "livres", verbe = "écrit". Il existe aussi des ambiguïtés qui relèvent du niveau syntaxique. Une même phrase peut avoir plusieurs sens possibles en fonction de ses interprétations syntaxique.

6.2. Absence des voyelles :

Le problème de la voyellation réside dans le fait qu'elle est absente dans les textes arabes. En effet, comme déjà expliqué précédemment, les signes de voyellation sont des signes diacritiques placés au-dessus ou au-dessous des lettres, qui apparaissent dans certains ouvrages scolaires pour débutants et dans le Coran. La non-voyellation génère plusieurs cas d'ambiguïté et des problèmes lors de l'analyse automatique.

6.3. La segmentation de textes :

La segmentation d'un texte est une étape fondamentale pour son traitement automatique. Son rôle est de découper le texte en unités d'un certain type qu'on aura défini et préalablement repéré. En effet, l'opération de segmentation d'un texte consiste à délimiter les segments de ses éléments de base qui sont les caractères, en éléments constituants différents niveaux structurels tels que : paragraphe, phrase, syntagme, mot graphique, mot-forme, morphème, etc.

Toutefois, les particularités de la langue arabe, rendent la segmentation arabe toujours différente, il n'y a pas de majuscules qui marquent le début d'une nouvelle phrase. De plus, les signes de ponctuation, ne sont pas utilisés de façon régulière. D'après l'étude réalisée par Belguith (Belguith et al. 2005) [18], certaines particules comme " و | et ", " ف | donc ", etc. jouent un rôle principal dans la séparation de phrases et peuvent être déterminantes pour guider la segmentation.

6.4. Agglutination de mots :

Contrairement à la plupart des langues latines, en arabe, les articles, les prépositions, les pronoms se collent aux adjectifs, noms, verbes et particules auxquels ils se rapportent. Un mot arabe peut parfois correspondre à toute une phrase. Par exemple, le mot arabe « أتستعملونها », est-ce que vous l'utilisez ? ». Cette caractéristique engendre des ambiguïtés morphologiques au cours de l'analyse. En effet, il est parfois difficile de distinguer entre un proclitique/enclitique et un caractère du mot en question. Par exemple, le caractère " و " dans le mot " وجمع " est un caractère qui fait partie de ce mot alors que dans le mot " وحصل ", il s'agit d'un proclitique.

7. Quelques travaux sur la classification du saint coran

Dans cette section nous présentons quelques approches introduites par les chercheurs pour résoudre les problèmes de la classification de coran.

7.1. Vérification de l'intégrité et de l'authentification en ligne des Versions électroniques du Coran

(Izzat Alsmadi & Mohammad Zarour, 2017) Ont introduit un système d'authentification qui peut vérifier tout éventuel «changement» ou «modification» de Versets du Coran. Puisque les versets du Coran sont littéralement identiques, Le système d'authentification sera capable de rechercher fréquemment sur Internet des pages Web susceptibles inclure ces versets de fraude et les indexer pour la surveillance, l'alerte, journalisation et vérification.

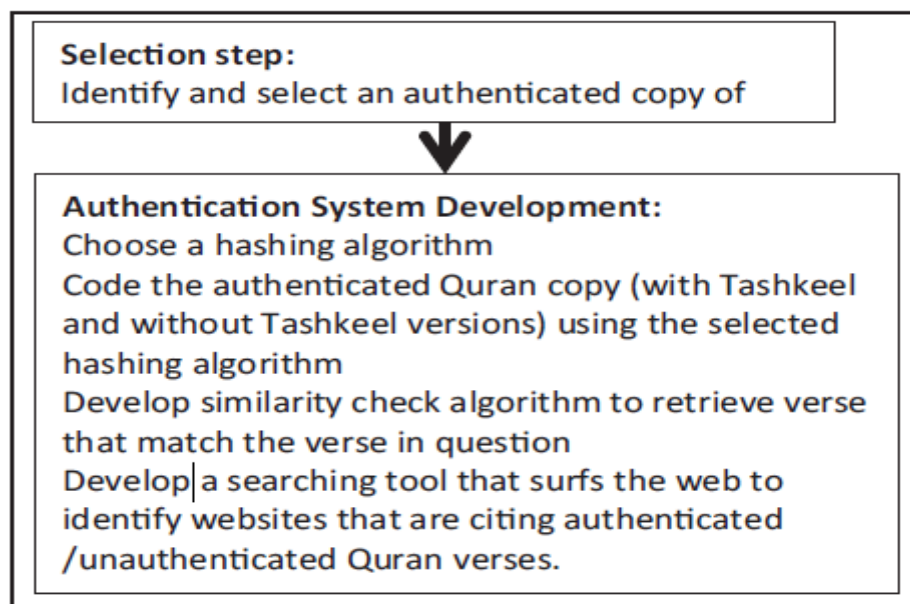


Figure 7 : Architecture globale du Système (Izzat Alsmadi & Mohammad Zarour, 2017)

7.2. Approche basée sur les résidus pour l'authentification modèle de textes arabes diacritiques multi-styles

(Saqib Hakak, Amirrudin Kamsin, Shivakumara Palaiahnakote, Omar Tayan, Mohd.Yamani Idna Idris & Khir Zuhaili Abukhir, 2018) authentifier les styles d'écriture uthmani et coranique puisque les deux sont largement utilisés pour la communication via le Web ou le courrier électronique. Pour chaque couplet dans le style Uthmani, la

méthode proposée trouve le résidu en effectuant une opération XOR au niveau du bit avec le vérité au sol. Le résidu a été étudié pour trouver une lettre appropriée à remplacer de telle sorte que le des versets uthmani donnés peuvent être convertis en texte clair du Coran.

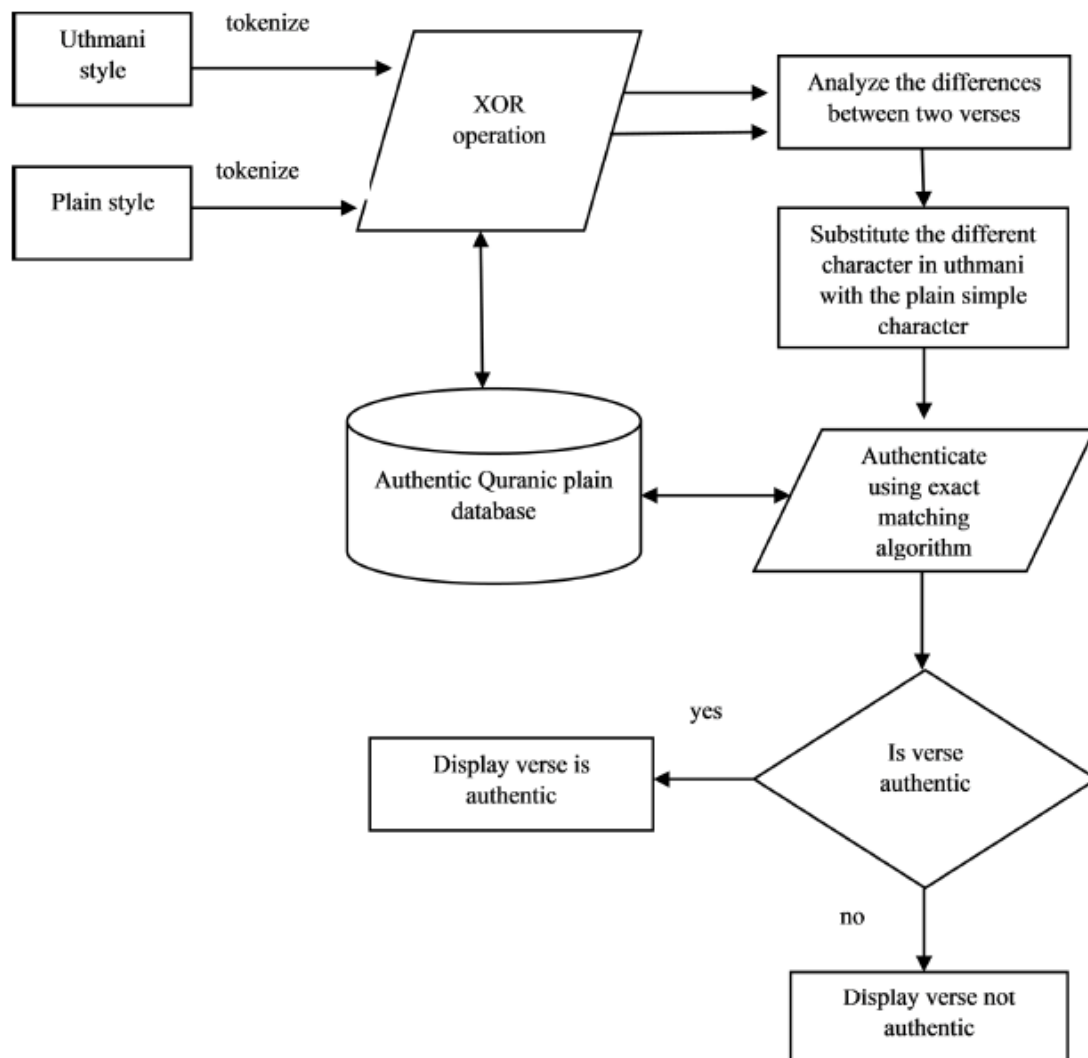


Figure 8 : Le déroulement logique de l'approche proposée. (Saqib Hakak, Amirrudin Kamsin, Shivakumara Palaiahnakote, Omar Tayan, Mohd.Yamani Idna Idris & Khir Zuhaili Abukhir, 2018)

7.3. Une classification topique

(Mohammed N. Al-Kabi, Belal M. Abu Ata, Heider A. Wahsheh & Izzat M. Alsmadi, 2013) Cette étude vise à évaluer l'efficacité d'un quatre algorithmes de classification bien connus (Decision Tree, KNearest Neighbor (K-NN), Support Vector

Machine (SVM) & Naïve Bayes (NB)) pour classer différents āyāt coraniques (versets) selon leurs thèmes. Le cadre comprend les procédures suivantes:

1. Suppression des signes diacritiques arabes ('حركات')
2. Suppression des symboles coraniques (tels que : 'صلى', 'ج')
3. ils ont utilisé la classification topique humaine manuelle des Āyāt coranique (versets) par [14] pour former et évaluer les quatre classificateurs considérés.

De nombreux livres classaient différents āyāt coraniques (versets) écrit par des érudits musulmans. Ces livres sont différents largement dans leurs tailles. Certains de ces livres se composent d'un nombre de volumes, et certains se composent de quelques pages. Cette étude est basée sur le lexique des sujets du Coran par [14].

i	Arabic Main Categories	English Translation
1	الباب الأول: أركان الإسلام	Part I: Pillars of Islam
2	الباب الثاني: الإيمان	Part II: Faith
3	الباب الثالث: العلوم	Part III: Science
4	الباب الرابع: العمل (أس الحياة)	Part IV: Working (MSI life)
5	الباب الخامس: الدعوة إلى الله	Part V: Call to God
6	الباب السادس: الجهاد	PART VI: Jihad
7	الباب السابع: الإنسان والعلاقات الإجتماعية	Chapter VII: Human and social relations
8	الباب الثامن: العلاقات الأخلاقية	Part VIII: Ethical relations
9	الباب التاسع: تنظيم العلاقات المالية	Part IX: Regulation of financial relations
10	الباب العاشر: العلاقات القضائية	Part X: Judicial relations
11	الباب الحادي عشر: العلاقات السياسية	Part XI: Political relations
12	الباب الثاني عشر: القصص القرآني	Chapter XII: Quranic stories
13	الباب الثالث عشر: الديانات السابقة	Chapter XIII: Previous religions
14	الباب الرابع عشر: تنوع الخطاب الإلهي	Chapter XIV: The diversity of divine discourse

Tableau 3 : Catégories Principales de āyāt arabe quranique (verses)

7.4. Utilisation du Deep Learning pour déterminer automatiquement l'application correcte de Règles de base de la récitation coranique

(Mahmoud Al-Ayyoub, Ismail Hmeidi & Nour Alhuda Mohamad Ali Damer, 2018) L'objectif est de construire un système capable de déterminer lequel des Ahkam Al-Tajweed est utilisé dans un enregistrement audio spécifique d'une récitation

coranique. Les Ahkam Al-Tajweed qu'ils ont considérés sont huit: «EdgamMeem» (une règle), «EkhfaaMeem» (une règle), «Ahkam Lam» dans le terme «Allah» (deux règles) et «Edgam Noon» (quatre règles). De plus, ils considèrent à la fois l'utilisation correcte et incorrecte de chaque règle. Par conséquent, les problèmes de classification impliquent 16 classes. Enfin, contrairement aux travaux précédents, le système couvre tout le Saint Coran. Ils commencent la discussion de l'approche avec la description de l'ensemble de données. Ensuite, ils discutent des étapes d'extraction et de classification des caractéristiques à l'aide de techniques populaires de la littérature sur le traitement de la parole.

Rule	Correct?	Recordings by males	Recordings by females
R1: EdgamMeem	Correct	60	60
	Incorrect	30	-
R2: EkhfaaMeem	Correct	60	60
	Incorrect	30	-
R3: Tafkheem Lam	Correct	88	512
	Incorrect	55	407
R4: Tarqeeq Lam	Correct	60	195
	Incorrect	30	134
R5: Edgam Noon (Noon)	Correct	138	138
	Incorrect	-	68
R6: Edgam Noon (Meem)	Correct	107	106
	Incorrect	-	53
R7: Edgam Noon (Waw)	Correct	52	52
	Incorrect	-	26
R8: Edgam Noon (Ya')	Correct	221	219
	Incorrect	-	110

Tableau 4 : Le nombre d'enregistrements audio pour chaque règle (indiquant à la fois l'utilisation correcte et incorrecte de cette règle).

8. Conclusion

Dans ce chapitre, nous avons présenté le domaine du Traitement Automatique du Langage Naturel et ses différents niveaux d'analyses. Ainsi, nous avons illustré les particularités de la langue arabe moderne à savoir : la morphologie, de même que la structure et la catégorie d'un mot. De plus, nous avons décrit les principaux problèmes d'analyse automatique de la langue arabe inhérents à certains phénomènes tels que la non-voyellation, l'agglutination, l'absence de ponctuation régulière, l'ambiguïté, etc.

Le chapitre suivant est entièrement dédié à la présentation des principales techniques de classification, leurs avantages et inconvénients ainsi que leurs domaines d'applications.

Chapitre 3

Les

Classificateurs

1. Introduction

Ces dernières années, la recherche a attaché une grande importance au traitement des données textuelles. Il y a plusieurs raisons à cela : de plus en plus de collections sont mises en réseau et distribuées à l'international, et le développement des infrastructures de communication et d'Internet. Le traitement manuel de ces données est très coûteux en temps et en personnel, il n'est pas très flexible, et il est presque impossible de les généraliser à d'autres domaines, c'est pourquoi nous avons essayé de développer des méthodes automatiques.

Actuellement, il existe de nombreux logiciels de classification de texte, ils ont été publiés et leurs champs d'application se développent. Généralement, ces systèmes sont basés sur des algorithmes d'apprentissage automatique, nous proposons donc une méthode d'apprentissage qui peut classer de nouveaux documents sur la base de documents déjà classés.

2. Algorithmes d'apprentissage

Dans l'apprentissage automatique, différents types de classificateurs ont été développés dans le but d'atteindre une précision et une efficacité maximales, et chaque classificateur a ses avantages et ses inconvénients. Cependant, ils partagent des caractéristiques communes.

Dans les pages qui suivent, nous allons exposer en détail quelques algorithmes d'apprentissage

Il existe de nombreux algorithmes d'apprentissage supervisé, notamment :

- L'algorithme des K plus proches voisins (ou KNN)
- Les arbres de décision
- Machines à support de vecteurs (ou SVM).
- Les réseaux de neurones (RNA).
- L'algorithme de Naïve Bayes.
- Multinomial Naïve Bayes.

2.1 Algorithme des K-voisins les plus proches KNN

2.1.1. Définition

L'algorithme des k-voisins les plus proches («k-nearest neighbors» ou KNN) connu dans WEKA sous le nom de IBK (Instance Based Learner) de la famille des « LAZY » classificateurs est une méthode d'apprentissage à base d'instances.

La méthode ne nécessite pas de phase d'apprentissage ; c'est l'échantillon d'apprentissage, associé à une fonction de distance et à une fonction de choix de la classe en fonction des classes des voisins les plus proches, qui constitue le modèle.

Lorsqu'un nouveau document à classer arrive, il est comparé aux documents d'entraînement à l'aide d'une mesure de similarité. Ses k plus proches voisins sont alors considérés : on observe leur catégorie et celle qui revient le plus parmi les voisins est assignée au document à classer. C'est là une version de base de l'algorithme que l'on peut raffiner. Souvent, on pondère les voisins par la distance qui les sépare du nouveau texte [19].

2.1.2. Principe de fonctionnement

L'algorithme de KNN comparé avec ceux déjà classés en cherchant ses K plus proches voisins. Une fois ces derniers déterminés, le nouveau document est classé dans la catégorie qui inclut le maximum de voisins parmi les K trouvés.

Deux paramètres sont utilisés : le nombre K et la fonction de similarité pour comparer le nouveau document à ceux déjà classés telle que la distance euclidienne par exemple qui est donnée par l'équation suivante :

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

L'algorithme suivant illustre le fonctionnement de l'algorithme KNN :

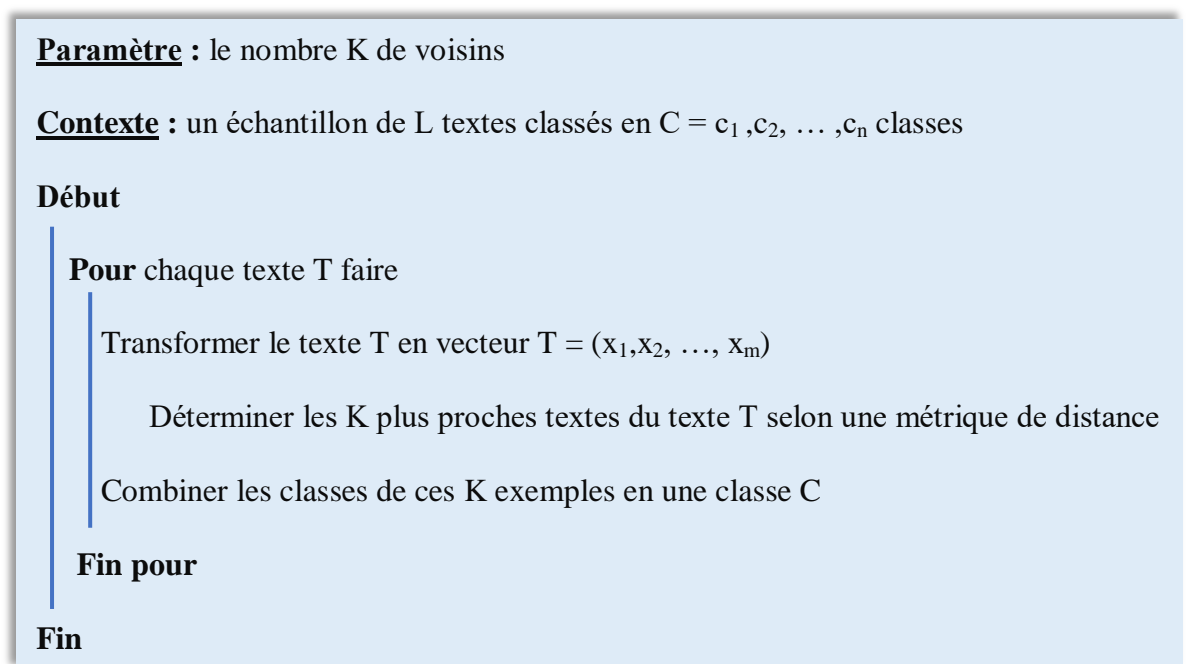


Figure 9 : Fonctionnement de l'algorithme KNN

La distance entre un texte et ses voisins se fait via une métrique de distance. Cette métrique peut être comme suit :

- **Mesure Cosinus** qui consiste à calculer le produit scalaire entre deux vecteurs a et b, que nous divisons par le produit de la norme de ces deux vecteurs. La formule de la mesure Cosinus est :

$$\cos(a, b) = \frac{\Sigma(a * b)}{\sqrt{\Sigma a^2 * \Sigma b^2}}$$

- **Mesure de Distance euclidienne** La formule de la mesure de Distance est comme suivante :

$$(a, b) = \sum |a - b|^2$$

- **Mesure de Jaccard** La formule de la mesure de Jaccard est :

$$(a, b) = \frac{\Sigma(a * b)}{\Sigma a^2 + \Sigma b^2 - \Sigma ab}$$

2.1.3. Critiques de la méthode

L'avantage que présente cette méthode est sa simplicité et son efficacité qui fait d'elle une méthode très utilisée ; toutefois, on peut lui reprocher le fait qu'elle utilise un nombre important d'objets pour calculer la similarité avec un nouvel objet à classer et plus le nombre d'objets est grand plus le temps d'exécution est très important.

2.1.4. Les domaines d'application

La méthode peut s'appliquer dès qu'il est possible de définir une distance sur les champs. Or, il est possible de définir des distances sur des champs complexes tels que des informations géographiques, des textes, des images, et du son. C'est parfois un critère de choix de la méthode K-PPV car les autres méthodes traitent difficilement les données complexes. On peut noter, également, que la méthode est robuste au bruit.

2.2. Les arbres de décisions

2.2.1. Définition

Les arbres de décision sont les plus populaires des méthodes d'apprentissage. Les Algorithmes connus sont ID3 (Quinlan 1986) [20] et C4.5 (Quinlan 1993) [21]. Ils sont également populaires pour la classification de document.

Comme toute méthode d'apprentissage supervisée, les arbres de décision utilisent des exemples. Si l'on doit classer des documents dans des catégories, il faut construire un arbre de décision par catégorie. Pour déterminer à quelle(s) catégorie(s) appartient un

nouveau document, on utilise l'arbre de décision de chaque catégorie auquel on soumet le document à classer. Chaque arbre répond Oui ou Non (il prend une décision).

Concrètement, chaque nœud d'un arbre de décision contient un test (un IF...THEN) et les feuilles ont les valeurs Oui ou Non. Chaque test regarde la valeur d'un attribut de chaque exemple. En effet, on suppose qu'un exemple est un ensemble d'attributs/valeurs. Pour des documents, chaque attribut peut être un mot et la valeur sera par exemple 0 ou 1 selon que ce mot appartient ou non au document.

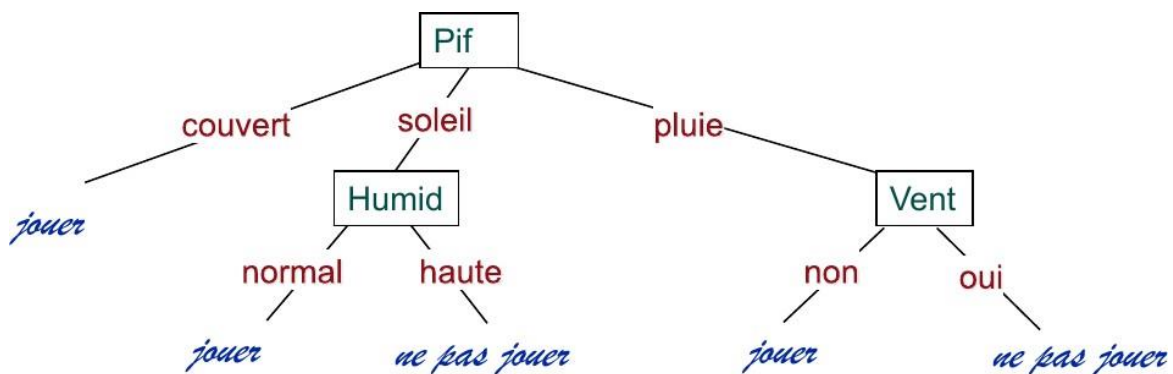


Figure 10 : Exemple d'arbre de décision

Pour construire l'arbre de décision, il faut trouver quel attribut tester à chaque nœud. C'est un processus récursif. Pour déterminer quel attribut tester à chaque étape, on utilise un calcul statistique qui détermine dans quelle mesure cet attribut sépare bien les exemples Oui/Non.

On crée alors un nœud contenant ce test, et on crée autant de descendants que de valeurs possibles pour ce test.

Exemple : si on teste la présence d'un mot, les valeurs possibles sont Présent/Absent. A chaque fois, on aura donc deux descendants pour chaque nœud ...

2.2.2. Algorithme

En général, l'algorithme d'arbre de décision se présente de la façon suivante :

Arbre ← arbre vide ; nœud_courantracine

Répéter

Décider si le nœud courant est terminal

Si le nœud terminal alors lui affecter une classe

Sinon sélectionner un test et créer autant de nœuds fils qu'il y a de réponse au test

Passer au nœud suivant (s'il existe)

Jusqu'à obtenir un arbre de décision

Figure 11 : Fonctionnement de l'algorithme d'arbre de décision

2.2.3. Critiques de la méthode

L'arbre de décision est une méthode très utilisée pour des raisons d'efficacité et de simplicité par rapport aux autres méthodes existantes ; en effet, elle est bien compréhensible pour tous les utilisateurs puisque ses règles sont de type « Si...Alors... ». Elle repose sur l'utilisation simultanée de variables qualitatives et quantitatives (discrètes ou continues). Sa classification est rapide : pour classer un nouvel objet, nous parcourons un seul chemin de l'arbre de la racine jusqu'à la feuille qui correspond à sa classe. Par contre, ses performances sont moins bonnes lorsque les classes sont nombreux, les arbres peuvent être très complexes et ne sont pas nécessairement optimaux. La construction des arbres de décisions nécessite généralement beaucoup de temps car il faut trouver le bon choix des attributs. Si les données évoluent dans le temps, il est nécessaire de relancer la phase d'apprentissage sur un échantillon complet qui contient les nouveaux et les anciens exemples.

2.2.4. Les domaines d'application

Cette méthode peut être utilisée dans plusieurs domaines tels que : Les études (pour comprendre les critères prépondérants dans l'achat d'un produit, l'impact des dépenses publicitaires), les ventes (pour analyser les performances par région, par enseigne, par vendeur), l'analyse de risques (pour détecter les facteurs prédictifs d'un comportement

de non- paiement), Le domaine médical (pour étudier les rapports existant entre certaines maladies et des particularités physiologiques ou sociologiques).

2.3. Machines à Support de Vecteurs (ou SVM)

Les machines à support de vecteurs (SVM) sont à l'origine de nouvelles méthodes de catégorisation, bien que les premières publications sur le sujet datent des années 60. [22]

Avant d'aborder le principe de fonctionnement général des SVM voici quelques notions de base :

- **Hyperplan** : est un séparateur d'objets des classes. De cette notion, nous pouvons dire qu'il est évident de trouver une multitude d'hyperplans mais la propriété délicate des SVM est d'avoir l'hyperplan dont la distance minimale aux exemples d'apprentissage est maximale, cet hyperplan est appelé l'hyperplan optimal, et la distance appelée marge.
- **Vecteurs Support** : ce sont les points qui déterminent l'hyperplan tels qu'ils soient les plus proches de ce dernier.

Voici un schéma représentatif de ces notions :

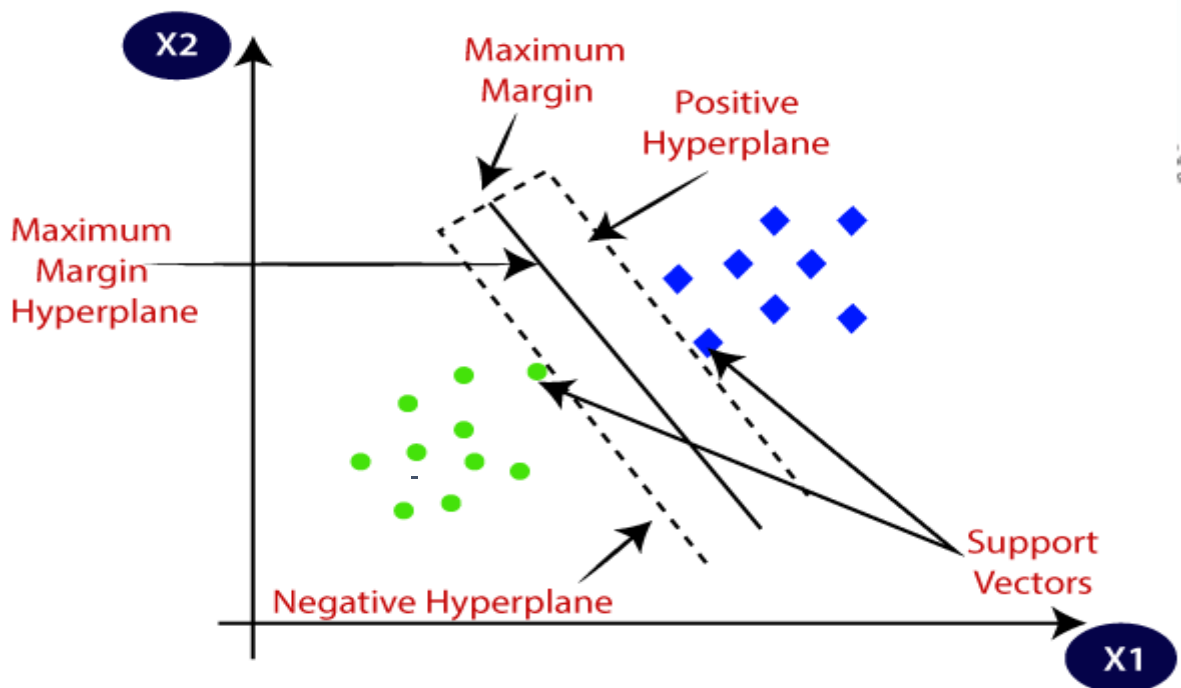


Figure 12 : Vecteurs de support machines

Le principe des SVM consiste en une stratégie de minimisation structurelle du risque mais le problème revient à trouver une frontière de décision qui sépare l'espace en deux régions, à trouver l'hyperplan qui classe correctement les données et qui se trouve le plus loin possible de tous les exemples. On dit qu'on veut maximiser la marge qui veut dire la distance du point le plus proche de l'hyperplan.

Dans le cas de la catégorisation des textes, les entrées sont des documents et les sorties sont des catégories. En considérant un classificateur binaire, on voudra lui faire apprendre l'hyperplan qui sépare les documents appartenant à la catégorie et ceux qui n'en font pas partie [9].

Les SVM conviennent bien pour la classification de textes parce qu'une dimension élevée ne les affecte pas puisqu'ils se protègent contre le sur apprentissage. Autrement dit, il affirme que peu d'attributs sont totalement inutiles à la tâche de classification et que les SVM permettent d'éviter une sélection agressive qui aurait comme résultat une perte d'information. On peut se permettre de conserver plus d'attributs. Également, une caractéristique des documents textuels est que lorsqu'ils sont représentés par des vecteurs, une majorité des entrées sont nulles.

Or, les SVM conviennent bien à des vecteurs dits clairsemés. Un autre aspect positif des SVM est qu'aucun ajustement de paramètres manuel n'est requis, car ils ont l'habileté de trouver automatiquement des paramètres adéquats.

2.4. Réseaux de neurones

Un réseau de neurones est un modèle de calcul dont la conception est très schématiquement inspirée du fonctionnement de vrais neurones (humains ou non). Les réseaux de neurones sont généralement optimisés par des méthodes d'apprentissage de type statistique grâce à leur capacité de classification et de généralisation, tels que la classification automatique de codes postaux ou la prise de décision concernant un achat boursier en fonction de l'évolution des cours. Ils enrichissent avec un ensemble de paradigmes permettant de générer de vastes espaces fonctionnels, souples et partiellement structurés. Il appartient d'autre part à la famille des méthodes de l'intelligence artificielle qu'ils enrichissent en permettant de prendre des décisions s'appuyant davantage sur la perception que sur le raisonnement logique forme.[23]

Un réseau de neurone est en général composé d'une succession de couches dont chacune prend ses entrées sur les sorties de la précédente. Chaque couche (i) est composée de N_i neurones, prenant leurs entrées sur les N_{i-1} neurones de la couche précédente. À chaque synapse est associé un poids synaptique, de sorte que les N_{i-1} sont multipliés par ce poids, puis additionnés par les neurones de niveau i, ce qui est équivalent à multiplier le vecteur d'entrée par une matrice de transformation.

Mettre l'une derrière l'autre, les différentes couches d'un réseau de neurones reviendrait à mettre en cascade plusieurs matrices de transformation et pourrait se ramener à une seule matrice, produit des autres, s'il n'y avait à chaque couche, la fonction de sortie qui introduit une non-linéarité à chaque étape. Ceci montre l'importance du choix judicieux d'une bonne fonction de sortie : un réseau de neurones dont les sorties seraient linéaires n'aurait aucun intérêt.

Les tentatives d'effectuer des classifications à l'aide de cet algorithme n'ont pas permis d'aboutir à des résultats et ce en raison de ressources importantes exigées.

2.5. Classification Naïve Bayésienne

La classification naïve bayésienne est un type de classification Bayésienne probabiliste simple basée sur le théorème de Bayes avec une forte indépendance (dite naïve) des hypothèses. Elle met en œuvre un classificateur bayésienne naïf, ou classificateur naïf de Bayes, appartenant à la famille des classificateurs Linéaires.

Un terme plus approprié pour le modèle probabiliste sous-jacent pourrait être « modèle à Caractéristiques statistiquement indépendantes ». [24]

En termes simples, un classificateur bayésien naïf suppose que l'existence d'une caractéristique pour une classe, est indépendante de l'existence d'autres caractéristiques. Un fruit peut être considéré comme une pomme s'il est rouge, arrondi, et fait une dizaine de centimètres. Même si ces caractéristiques sont liées dans la réalité, un classificateur bayésien naïf déterminera que le fruit est une pomme en considérant indépendamment ces caractéristiques de couleur, de forme et de taille.

Selon la nature de chaque modèle probabiliste, les classificateurs bayésiens naïfs peuvent être entraînés efficacement dans un contexte d'apprentissage supervisé.

Dans beaucoup d'applications pratiques, l'estimation des paramètres pour les modèles bayésiennes naïfs repose sur le maximum de vraisemblance. Autrement dit, il est possible de travailler avec le modèle bayésienne naïf sans se préoccuper de probabilité bayésienne ou utiliser les méthodes bayésiennes.

Malgré leur modèle de conception « naïf » et ses hypothèses de base extrêmement simplistes, les classificateurs bayésienne naïfs ont fait preuve d'une efficacité plus que suffisante dans beaucoup de situations réelles complexes. En 2004, un article a montré qu'il existe des raisons théoriques derrière cette efficacité inattendue. Toutefois, une autre étude de 2006 montre que des approches plus récentes (arbres renforcés, forêts aléatoires) permettent d'obtenir de meilleurs résultats. L'avantage du classificateur bayésienne naïf est qu'il requiert relativement peu de données d'entraînement pour estimer les paramètres nécessaires à la classification, à savoir moyennes et variances des différentes variables. En effet, l'hypothèse d'indépendance des variables permet de se contenter de la variance de chacune d'entre elle pour chaque classe, sans avoir à calculer de matrice de covariance.

2.5.1. Description du modèle Bayésienne

Le modèle probabiliste pour un classificateur est le modèle conditionne $(C|F_1, \dots, F_n)$ où "C" est une variable de classe dépendante dont les instances ou classes sont peu nombreuses, conditionnée par plusieurs variables caractéristiques F_1, \dots, F_n .

Lorsque le nombre de caractéristiques n est grand, ou lorsque ces caractéristiques peuvent prendre un grand nombre de valeurs, baser ce modèle sur des tableaux de probabilités devient impossible. [25]

Par conséquent, nous le dérivons pour qu'il soit plus facilement soluble. À l'aide du théorème de Bayes, nous écrivons

$$P(C|F_1, \dots, F_n) = \frac{P(C)P(F_1, \dots, F_n|C)}{P(F_1, \dots, F_n)}$$

En langage courant, cela signifie :

$$Postérieure = \frac{Antérieure * Vraisemblance}{Evidence}$$

En pratique, seul le numérateur nous intéresse, puisque le dénominateur ne dépend pas de C et les valeurs des caractéristiques F_i sont données. Le dénominateur est donc en réalité constant. Le numérateur est soumis à la loi de probabilité à plusieurs variables.

$$P(C, F_1, \dots, F_n)$$

et peut-être factorisé de la façon suivante, en utilisant plusieurs fois la définition de la probabilité conditionnelle :

$$\begin{aligned} p(C, F_1, \dots, F_n) &= p(C) p(F_1, \dots, F_n | C) \\ &= p(C) p(F_1 | C) p(F_2, \dots, F_n | C, F_1) \\ &= p(C) p(F_1 | C) p(F_2 | C, F_1) p(F_3, \dots, F_n | C, F_1, F_2) \\ &= p(C) p(F_1 | C) p(F_2 | C, F_1) p(F_3 | C, F_1, F_2) p(F_4, \dots, F_n | C, F_1, F_2, F_3, \dots) \\ &= p(C) p(F_1 | C) p(F_2 | C, F_1) p(F_3 | C, F_1, F_2) \dots p(F_n | C, F_1, F_2, F_3, \dots) \end{aligned}$$

C'est là que nous faisons intervenir l'hypothèse naïve : si chaque F_i est indépendant des autres caractéristiques $F_j \neq i$ alors

Pour tout $i \neq j \in \mathbb{R}$, par conséquent la probabilité conditionnelle peut s'écrire

$$\begin{aligned} p(F_i | C, F_j) &= p(F_i | C) \\ p(C, F_1, \dots, F_n) &= p(C) p(F_1 | C) p(F_2 | C) p(F_3 | C) \dots \\ &= p(C) \prod_{i=1}^n p(F_i | C). \end{aligned}$$

Par conséquent, en tenant compte de l'hypothèse d'indépendance ci-dessus, la probabilité conditionnelle de la variable de classe C peut être exprimée par où

$$\begin{aligned}
 p(F_i|C, F_j) &= p(F_i|C) \\
 p(C, F_1, \dots, F_n) &= p(C) p(F_1|C) p(F_2|C) p(F_3|C) \dots \\
 &= p(C) \prod_{i=1}^n p(F_i|C).
 \end{aligned}$$

Où Z (appelé « évidence ») est un facteur d'échelle qui dépend uniquement de F_1, \dots, \dots , à savoir une constante dans la mesure où les valeurs des variables caractéristiques sont connues.

Les modèles probabilistes ainsi décrits sont plus faciles à manipuler, puisqu'ils peuvent être factorisés par l'antérieure $P(C)$ (probabilité a priori de C) et les lois de probabilité indépendantes $P(F_i|C)$. S'il existe K classes pour C et si le modèle pour chaque fonction peut être exprimé selon paramètres, alors le modèle bayésien naïf correspondant dépend de $(k - 1) + n r k$ paramètres.

Dans la pratique, on observe souvent des modèles où $K=2$ (classification binaire) et $r=1$ (les caractéristiques sont alors des variables de Bernoulli). Dans ce cas, le nombre total de paramètres du modèle bayésien naïf ainsi décrit est de $2n+1$, avec n le nombre de caractéristiques binaires utilisées pour la classification.

2.5.2. Estimation de la valeur des paramètres

Tous les paramètres du modèle (la probabilité a priori de la catégorie et la loi de probabilité liée aux différentes caractéristiques) peuvent être approximés par rapport à la fréquence relative des catégories et des caractéristiques dans l'ensemble de données d'apprentissage. Il s'agit d'une estimation de la probabilité maximale de probabilité. Par exemple, la probabilité antérieure de la catégorie peut être calculée sur la base de l'hypothèse que la catégorie est à probabilité égale (i.e., chaque antérieure = $1 /$ (le nombre de classes)), ou elle peut être calculée en estimant la probabilité de chaque catégorie sur la base de l. Ensemble de données d'apprentissage (i.e. antérieure de $C =$ (nombre d'échantillons de C) / (nombre d'échantillons totaux)).

Pour estimer les paramètres d'une loi de probabilité relative à une caractéristique précise, il est nécessaire de présupposer le type de la loi en question ; sinon, il faut générer des modèles non-paramétriques pour les caractéristiques appartenant à

l'ensemble de données d'entraînement. Lorsque l'on travaille avec des caractéristiques qui sont des variables aléatoires continues, on suppose généralement que les lois de probabilités correspondantes sont des lois normales, dont on estimera l'espérance et la variance.

L'espérance, μ , se calcule avec

$$\mu = \frac{1}{N} \sum_{i=1}^N X_i$$

Où N est le nombre d'échantillons et X_i est la valeur d'un échantillon donné. La variance, σ^2 , se calcule avec

$$\sigma^2 = \frac{1}{(N-1)} \sum_{i=1}^N (X_i - \mu)^2$$

2.5.3. Construire un classificateur à partir du modèle de probabilités

Jusqu'à présent nous avons établi le modèle à caractéristiques indépendantes, à savoir le modèle de probabilités bayésien naïf. Le classificateur bayésien naïf couple ce modèle avec une règle de décision.

Le terme multinomial naïf Bayes (Naïve Bayes) nous indique simplement que chaque $p(f_i | c)$ est une distribution multinomiale, plutôt qu'une autre. Cela fonctionne bien pour les données qui peuvent facilement être converties en comptes, tels que le nombre de mots dans le texte.

En résumé, le classificateur Naïve Bayes est un terme général qui désigne l'indépendance conditionnelle de chacune des caractéristiques du modèle, tandis que le classificateur Naïve Bayes multinomial est une instance spécifique d'un classificateur Naïve Bayes qui utilise une distribution multinomiale pour chacune des caractéristiques. Le classificateur correspondant à cette règle est la fonction "classificateur" suivante :

$$\text{classifieur}(f_1, \dots, f_n) = \underset{c}{\operatorname{argmax}} p(C = c) \prod_{i=1}^n p(F_i = f_i | C = c).$$

2.5.4. Analyse

Fait étonnant, malgré les hypothèses d'indépendance relativement simplistes, le classificateur bayésienne naïf a plusieurs propriétés qui le rendent très pratique dans les cas réels. En particulier, la dissociation des lois de probabilités conditionnelles de classe entre les différentes caractéristiques aboutit au fait que chaque loi de probabilité peut être estimée indépendamment en tant que loi de probabilité à une dimension. Cela permet d'éviter nombre de problèmes venant du fléau de la dimension, par exemple le besoin de disposer d'ensembles de données d'entraînement dont la quantité augmente exponentiellement avec le nombre de caractéristiques.

Comme tous les classificateurs probabilistes utilisant la règle de décision du maximum a posteriori, il classifie correctement du moment que la classe adéquate est plus probable que toutes les autres. Par conséquent les probabilités de classe n'ont pas à être estimées de façon très précise.

Le classificateur dans l'ensemble est suffisamment robuste pour ne pas tenir compte de sérieux défauts dans son modèle de base de probabilités naïves. La documentation citée en fin d'article détaille d'autres raisons pour le succès empirique des classificateurs bayésiens naïfs.

2.6. Réseau de neurones récurrents et « Long Short-Term Memory »

Les convNets et réseaux de neurones connectés sont conçus pour traiter des problèmes dont les variables en entrée ont une instance indépendante dans le temps. Par contre, il y a certains problèmes où l'ordre des événements compte. Comme dans le cas des traitements vidéo qui sont des séquences d'images dans le temps, ou des traitements de textes qui sont une suite de mots successifs et interdépendants. Pour cela, le réseau de neurones récurrent (RNN, Recurrent Neural Network), une autre extension du réseau de neurones connecté, a été introduit. Le RNN est un réseau de neurones avec mémoire où les informations du passé importent dans l'algorithme. Donc il est fait pour traiter le problème de données séquentielles.

Un simple RNN se construit juste en prenant la couche sortie de la précédente étape et le concaténé avec l'entrée de l'étape courante. Tout cela dans le but d'avoir la prédiction de l'étape courante (équation 1), d'où le nom de récurrent pour la récurrence.

$$y_t = f_w(x_t - y_{t-1}) \quad (1)$$

y_t : Entrée à l'instant t

x_t : Sortie prédite pour l'instant t-1

y_{t-1} : Sortie prédite pour l'instant t

w : Poids

f : La fonction d'activation dans le réseau de neurones connectés (tanh, sigmoïde, relu).

Dans le cas général on n'utilise pas cette forme simple de RNN car cette forme ne suffit pas pour avoir les meilleurs résultats. De plus, l'information apportée par la valeur de sortie à $t - 1$ n'est pas très riche [26]. Ainsi, au lieu d'utiliser la valeur de sortie de l'instant t-1, on met en paramètres les valeurs de la couche cachée à t-1 qui sont plus significatives en termes neuronaux. La formulation de RNN est dans l'équation 3.

$$h_t = f_w(x_t, h_{t-1}) \quad 2$$

$$y_t = W \cdot h_t \quad 3$$

x_t : Entrée à l'instant t

h_{t-1} : Couche cachée pour l'instant t-1

h_t : Couche cachée prédite pour l'instant t

y_t : Sortie prédite pour l'instant t

f : La fonction d'activation dans le réseau de neurones connecté.

La Figure 13 présente l'illustration de l'équation 3 sous la forme d'un graphe.

D'après cette figure et l'équation 3, le problème avec le RNN est le fait de garder en mémoire toutes les informations de chaque instant si l'on veut qu'il soit assez flexible.

Par exemple (pris de [27]) dans le texte "J'ai grandi en France ... (2000mots)... Je parle couramment le xxxx". On voudrait prédire le xxxx. Le plus logique est xxxx soit égal à "français". Mais pour qu'un RNN trouve que xxxx est égal à français, il doit mémoriser 2000 instants, ce qui est conséquent.

Tout cela va créer un très profond réseau de neurones au niveau de la récurrence (voir figure 13) et cela pénalise l'apprentissage en matière de mémoire (de l'ordinateur) et de temps d'apprentissage.

Comme solution, Hochreiter et Schmidhuber [28] ont inventé le LSTM (Long-Short Term Memory Cell) en 1997. Le LSTM est juste une autre forme de RNN. Le LSTM est conçu dans le but de supporter les problèmes aux longs termes de dépendances parce que sa plus grande particularité est de mémoriser beaucoup d'informations. Dans une couche LSTM, on a 4 neurones que l'on appelle porte alors que dans un RNN on a un seul neurone. Ces neurones de LSTM ont chacun leurs rôles et interagissent entre eux de manière spécifique. Le modèle mathématique de LSTM est dans l'équation

$$f_t = \sigma(W_f[x_t, h_{t-1}] + b_f) \quad 4$$

$$i_t = \sigma(W_i[x_t, h_{t-1}] + b_i) \quad 5$$

$$\tilde{C}_t = \tanh(W_{\tilde{c}}[x_t, h_{t-1}] + b_{\tilde{c}}) \quad 6$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad 7$$

$$\sigma_t = \sigma(W_o[x_t, h_{t-1}] + b_o) \quad 8$$

$$h_t = \sigma_t * \tanh(C_t) \quad 9$$

f_t : Forget gate (porte d'oubli), c'est ce noeud qui décide quelle information sera supprimée/oubliée et quelle information sera gardée.

i_t : Input gate (porte d'entrée), c'est ce noeud qui décide quelle valeur sera mise à jour,

\tilde{C}_t : New memory cell sert à stocker les informations. Combinée avec i_t , on y stocke les informations qui viennent d'être mises à jour.

C_t : Final memory cell supprime les anciennes informations oubliées de l'état précédent

$f_t * C_{t-1}$ et ajoute les nouvelles informations mises à jour ($i_t * \tilde{C}_t$).

σ_t : Output gate : décide quelle information va à la sortie,

h_t : Hidenstate : est la sortie, on filtre le final memory cell C_t avec l'output gate,

Les $W_f, W_i, W_{\tilde{c}}, W_o$ et $b_f, b_i, b_{\tilde{c}}, b_o$ sont les poids et biais respectifs des neurones f, i, \tilde{C}, o .

Ils sont indépendants du temps mais ce sont toujours eux que l'on va apprendre dans la phase d'apprentissage.

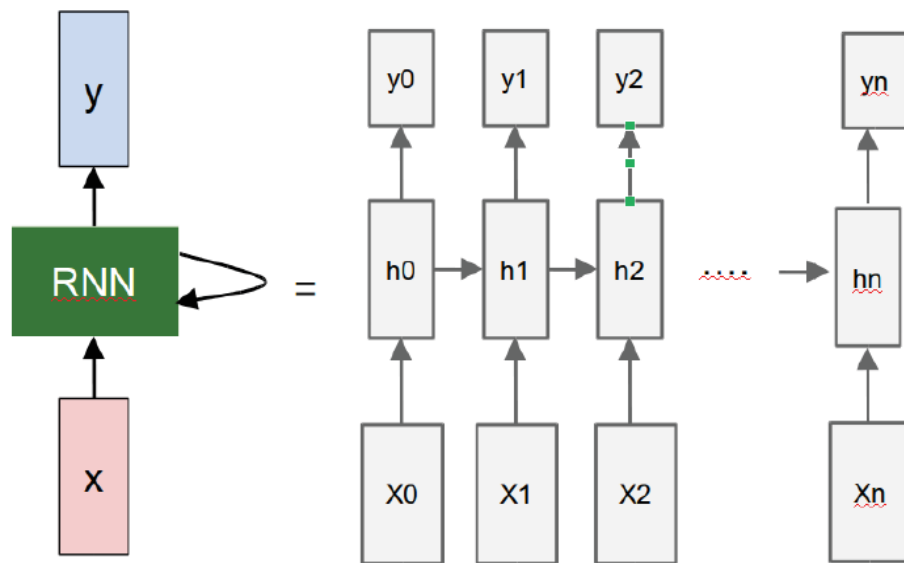


Figure 13 : Réseau de neurones récurrent [29]

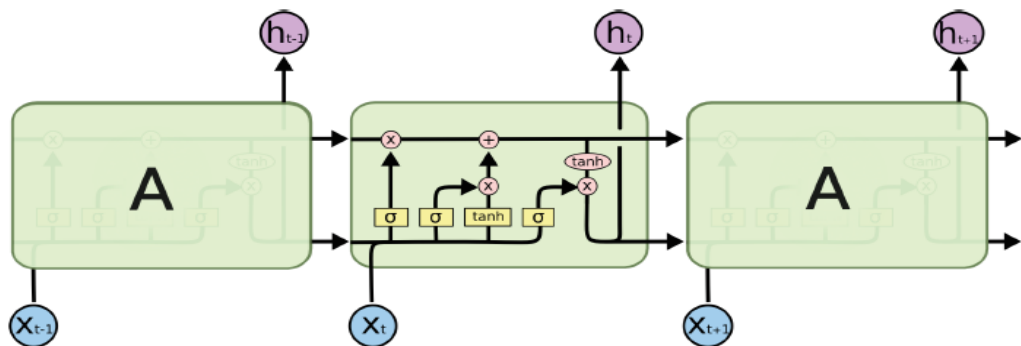


Figure 14 : Module répété d'un LSTM, avec les 4 neurones (portes) en jaune [27]

3. Conclusion

La classification supervisée de documents a fait beaucoup de progrès ces dernières années.

Nous avons présenté les principales techniques de classification automatique supervisées, utilisées pour classer des unités textuelles en groupes homogènes.

La discrimination (ou les méthodes supervisées) peut être basée sur des hypothèses probabilistes (Classificateur naïf de Bayes, méthodes paramétriques) ou sur des notions de proximité (plus proches voisins) ou bien encore sur des recherches dans des espaces d'hypothèses (arbres de décision, réseaux de neurones). Certes l'approche supervisée est très utilisée pour les raisons et les avantages qu'on a mentionné pour chaque méthode.

Chapitre 4

Expérimentation

&

Implémentation

1. Introduction

Comme déjà vu dans les précédents chapitres la classification de texte consiste à attribuer des catégories prédéfinies à des documents en texte libre selon leur contenu. La classification du texte arabe et surtout la classification du saint coran possède ses propres difficultés et limites résultant de la nature de la langue arabe qui est une langue riche en variétés avec une morphologie très complexe laquelle morphologie peut faire d'une analyse ordinaire une tâche très compliquée.

Le but de ce chapitre est de mettre en évidence les algorithmes les plus efficaces que nous avons appliqués afin de classifier notre corpus.

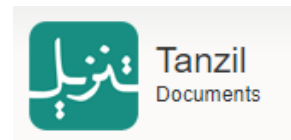
2. Présentation du corpus

La collection des manuscrits, des livres et des articles de presses pour les analyser est une tâche très ardue dans sa nature [Abdelali et al., 2004b]. Grâce à l'avancée technologique dans le stockage informatique et l'accès à une large quantité d'information, la construction des corpus de textes continue à se développer.

2.1. Qu'est-ce qu'un corpus ?

Un corpus est un ensemble de documents respectant deux critères, à savoir l'homogénéité; un corpus homogène couvre un domaine spécifique dans toute sa diversité et sa taille, représentée par le nombre de mots.

Pour notre travail, toutes les expérimentations ont été portées sur le corpus TANZIL avec une modification



2.2. Le projet TANZIL

Tanzil est un projet coranique lancé au début de 2007 pour produire un texte coranique Unicode hautement vérifié à utiliser dans les sites Web et les applications coraniques. Le but du projet Tanzil est de produire un texte coranique Unicode standard et de servir de source fiable pour ce texte standard sur le Web.

The screenshot shows the website interface for Tanzil Documents. At the top left is the logo with the Arabic word 'تنزيل' and 'Tanzil Documents'. A search bar is located at the top right. On the left side, there is a 'Navigation' menu with links to Home, Project Info, FAQ, News, Resources, Credits, Contribution, Donation, and All Pages. Below that is a 'Links' section with icons for Quran Navigator, Quran Text, and Translations. The main content area has a 'Welcome!' heading followed by a paragraph about the project's goal. Below this is a section titled 'Tanzil Quran Text Features' with four bullet points: Accuracy, Searchability, Pause Marks, and Compatibility. To the right of this section is a decorative Islamic calligraphy graphic.

Figure 15 : Le site web « TANZIL »

2.3. Pourquoi TANZIL ?

Depuis l'apparition de la première copie numérique du Saint Coran, il y avait eu des efforts substantiels pour produire un texte coranique précis, mais en raison de certaines difficultés, ces efforts ont échoué dans de nombreux cas, et Malheureusement, les textes du Coran sont apparus dans la majorité des sites Web coraniques et les applications souffraient de nombreuses erreurs et fautes de frappe.

2.4. Fautes de frappe dans certains textes existants

De nombreux textes numériques du Coran accessibles au public sur le Web ont souffert de nombreuses fautes de frappe, principalement en raison des faits suivants:

- **Diacritiques manquants :** Un texte coranique populaire qui était largement utilisé dans les sites Web et les applications était basé sur une ancienne copie sans diacritique du Coran dans laquelle des signes diacritiques avaient été ajoutés manuellement au fil du temps par des volontaires. Malheureusement, les signes diacritiques ajoutés à ce texte étaient incomplets et contenaient des tonnes de fautes de frappe.
 - **Conversion de texte :** Alors que les copies imprimées du Saint Coran sont généralement publiées en écriture Uthmani, les copies électroniques du Coran étaient généralement préparées dans le script simple (Imlaei). La conversion entre ces deux types de textes avait été à l'origine de nombreuses incohérences et fautes de frappe dans les textes numériques.
 - **Difficultés techniques :** Les anciens jeux de caractères arabes utilisés dans les pages Web et les applications coraniques manquaient de certains signes diacritiques et symboles essentiels. Par exemple, l'encodage arabe largement utilisé appelé «Windows-1256» n'avait pas de point de code pour small-alef, qui était nécessaire pour représenter un texte complet du Coran.
- ❖ Le data set qui nous avons utilisé c'est un extrait de projet TANZIL, cet data set est illustré dans la figure suivante :

text
الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ
كل شيء نسي في الحياة
الرَّحْمَنُ الرَّحِيمُ
مهما ساءت الأمور فليست شرا كلها
مَالِكِ يَوْمِ الدِّينِ
ولن تجد الناس جميعا يجمعون على أمر واحد
إِنَّكَ تَخْبُؤُا وَإِنَّكَ تَسْتَجِيبُ
ودورها في تثقيف الإنسان
اهْبِئَا الصِّرَاطَ الْمُسْتَقِيمَ
التحبير عن قضاياها
صِرَاطَ الَّذِينَ أَنْعَمْتَ عَلَيْهِمْ غَيْرِ الْمَغْضُوبِ عَلَيْهِمْ وَلَا الضَّالِّينَ
من هذه الفنون الأدب

Figure 16 : Un extrait du fichier du jeu de données utilisé (data sets)

3. La mise en œuvre de l'approche proposée

3.1. Le langage utilisé

Notre projet est fondé sur l'utilisation de l'apprentissage en profondeur dans la classification des versets du saint coran.

Comme nous l'avons déjà mentionnée ci-dessus la popularité croissante de python qui reflète à des caractéristiques spécifiques (capacité puissante, les plus demandés par les développeurs...etc.)

D'après l'IEEE¹³, (Guilloux, 2018) cela s'explique d'abord par le fait que Python est maintenant répertorié en tant que langage pour l'embarquer. L'association des professionnels techniques explique qu'auparavant, l'écriture d'applications embarquées était la chasse gardée des langages compilés, parce que cela impliquait des machines avec une puissance de traitement et une mémoire limitée. Mais aujourd'hui, beaucoup de microcontrôleurs modernes ont assez de puissance pour héberger un interpréteur Python. Et un aspect intéressant de l'utilisation de Python dans ce domaine serait qu'il est très pratique dans certaines applications de communiquer avec du matériel via une invite interactive ou de recharger dynamiquement des scripts à la volée. « La croissance de Python dans un nouveau

¹³ IEEE publie les principales revues, transactions, lettres et magazines de premier plan en génie électrique, informatique, biotechnologie, télécommunications, énergie et énergies, ainsi que des dizaines d'autres technologies.

domaine ne peut que renforcer la popularité du langage », affirme l'IEEE. Mais ce n'est pas tout.

En générale l'implémentation des projets deep Learning se fait par le langage python nous choisissons ce langage *a fortiori* de grandes bibliothèques disponible pour l'apprentissage en profondeur tel que : Keras, Theano, Tensorflow, Tensorbraod, Spacy, Numpy...etc.). Ainsi que la simplicité, abondance de soutien...etc.

3.2. Environnement d'exécution

Nous utilisons **google colab** comme un environnement d'exécution de notre projet

Qu'est-ce que **Google colab** :

Google Colab est un service en nuage (cloud) gratuit qui prend désormais en charge les GPU gratuits

Nous pouvons :

- Améliorons nos compétences de codage en langage de programmation Python.
- Développons des applications d'apprentissage approfondi à l'aide de bibliothèques populaires telles que **Keras**, **TensorFlow**, **PyTorch** et **OpenCV**.

La fonctionnalité la plus importante qui distingue Colab des autres services de cloud computing gratuits est la suivante : Colab fournit un GPU est totalement gratuit.

Les étapes d'utilisation :

1. Nous utilisons google colab dans notre google drive
2. Nous créons un dossier appel ' Projet '
3. Créons un nouveau carnet via **NOUVEAU NOTEBOOK**

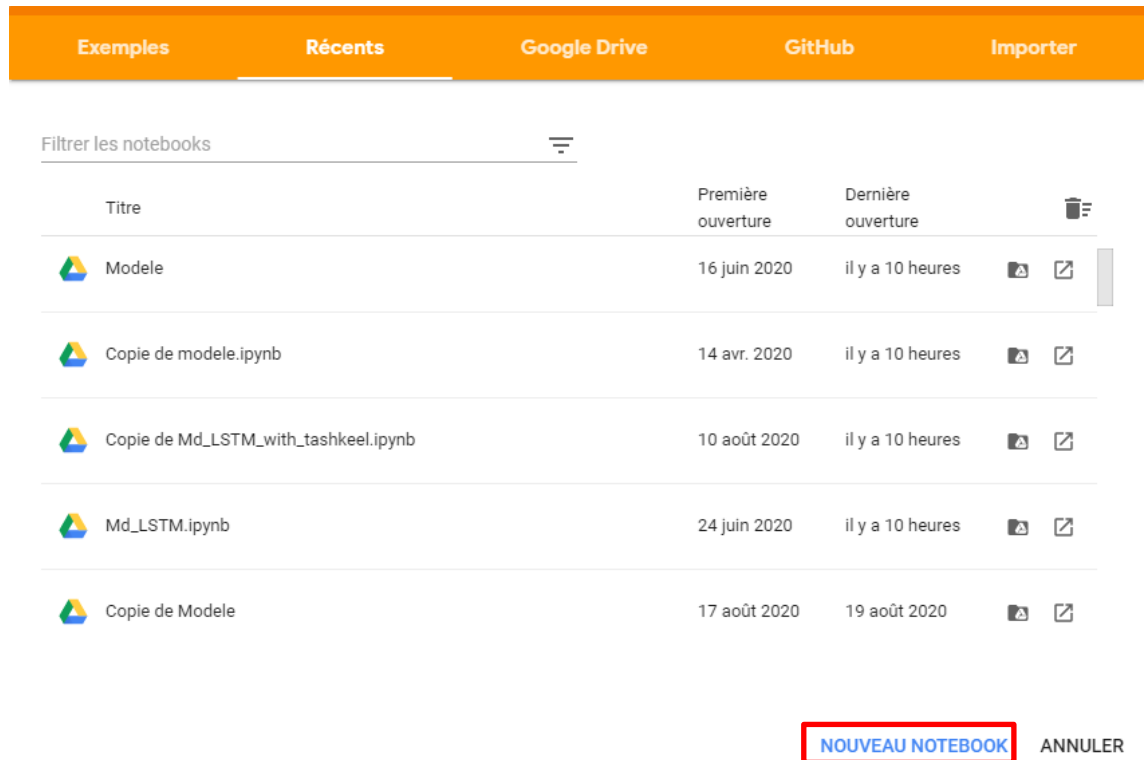


Figure 17 : L’environnement « Google Colab » - Création NOUVEAU NOTEBOOK -

4. Nous renommons le projet ‘Essai’

5. Pour changer le mode d’exécution, dans la barre des options choisir “exécution” puis “modifier le type d’exécution” et mettre l’option accélérateur matériel en mode GPU ou TPU.

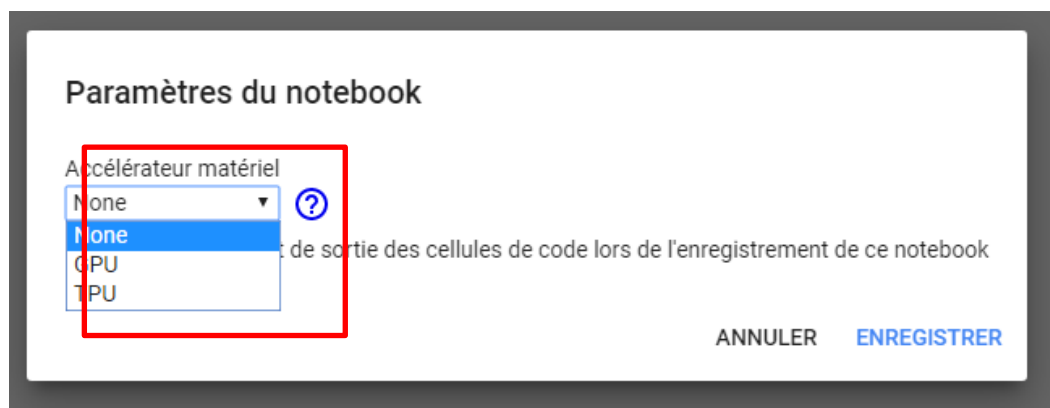


Figure 18 : Mode d’exécution

6. En fin nous exécutons le code en choisissant les cellules de type ‘code’

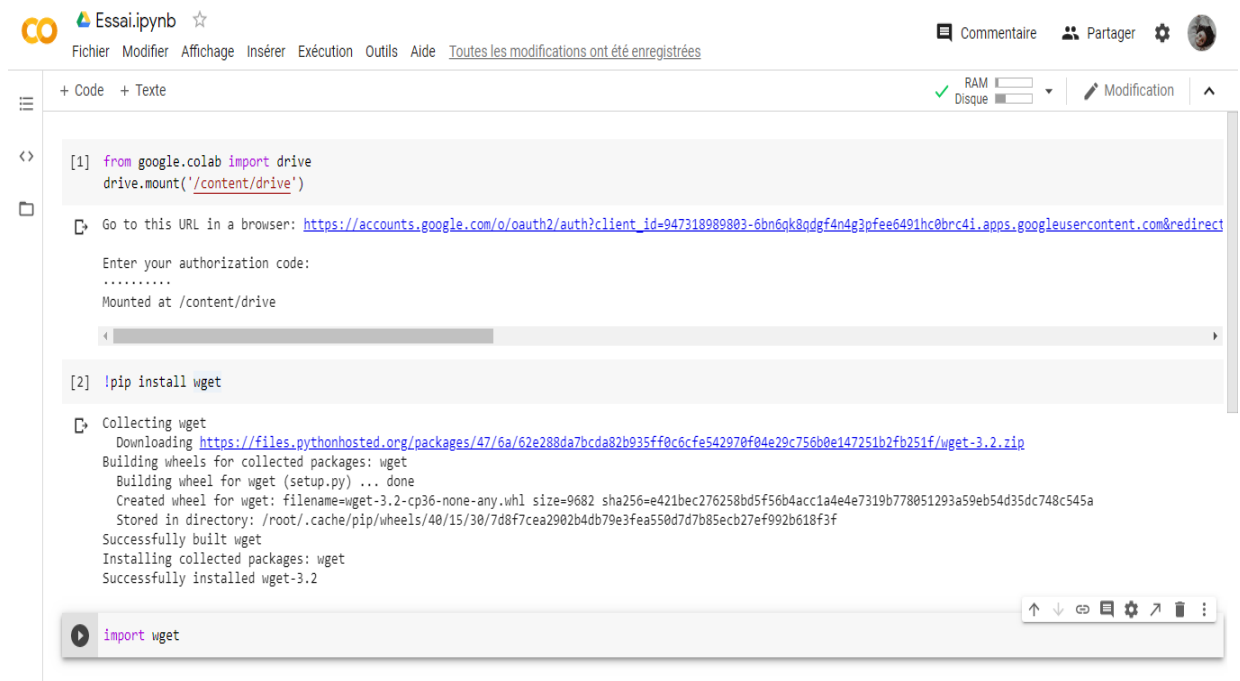


Figure 19 : Environnement d'exécution « Google Colab »

3.3. Les bibliothèques utilisées

Nous allons besoin de plusieurs bibliothèques dans notre application tell que :

```
import os

import re
import pickle
import numpy as np
import pandas as pd
from pandas import DataFrame
import keras
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras import layers
from keras.backend import clear_session
pd.set_option('display.max_colwidth', -1)
seed = 42
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(color_codes=True)
```

Figure 20 : les différentes bibliothèques utilisées dans notre implémentation

3.4. Phase 1 prétraitement

Nous appliquons cette phase avant de transmettre nos données au modèle.

Le prétraitement des données peut comporter plusieurs étapes en fonction des données et de la situation, notre jeu de données est de format csv, L'extraction et le prétraitement des caractéristiques de texte pour les algorithmes de classification sont très importants.

- ✓ Le jeu de données est contenu 50 phrases

Ensuite nous divisons le data set en deux parties, la première partie est pour l'entraînement et la deuxième partie pour le test.

- Pour l'entraînement nous effectuons 85 % de données
- Pour le Test nous effectuons 15% de données

```
[ ] df.groupby('label').cleaned_text.count().plot.bar(ylim=0)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa4091a95f8>
```

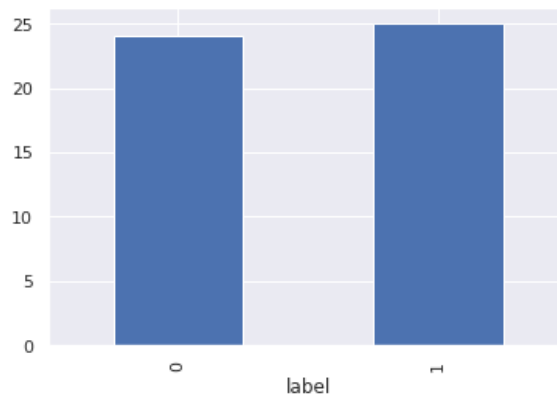


Figure 21 : Statistique du Data sets

- Le 1 signifie des versets du saint coran
- Le 0 signifient des phrases arabes

3.4.1. Nettoyage et prétraitement du texte

Dans la PNL, les textes et les documents contiennent de nombreux bruit (mot vide, orthographe, mot redondante...etc.).

Dans notre travail on a essayé de faire deux configurations :

- **WITH TASHKEEL**
- **WITHOUT TASHKEEL**

3.5. Phase 2 Création de modèle

L'architecture de modèle est basée sur les réseaux de neurones de type **LSTM** (Long Short Time Memory).

3.6. Phase 3 Prédiction (Résultat)

Nous allons utiliser la Fonction **predict** pour la prédiction d'une phrase

```
[ ] Testing_context = ["بخادعون الله والذين امنوا وما يخدعون الا انفسهم وما يشعرون"]  
  
txts = tok.texts_to_sequences(Testing_context)  
txts = sequence.pad_sequences(txts,maxlen=max_len)
```

```
[ ] preds = model.predict(txts)  
print(preds)
```

```
↳ [[0.9983718]]
```

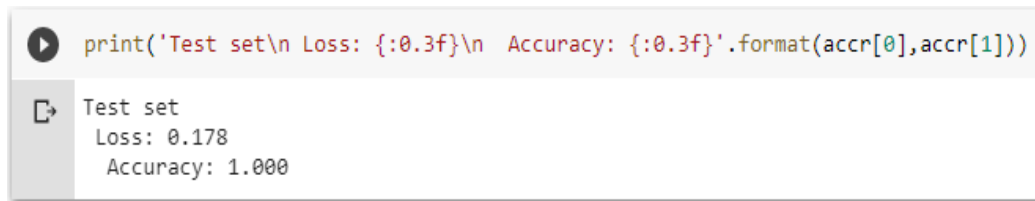
```
[ ] Testing_context = ["ودورها في تكويف الانسان"]  
  
txts = tok.texts_to_sequences(Testing_context)  
txts = sequence.pad_sequences(txts,maxlen=max_len)
```

```
[ ] preds = model.predict(txts)  
print(preds)
```

```
↳ [[0.36358014]]
```

Figure 22 : Prédiction des phrases

Nous utilisons une mesure d'évaluation pour Tester la performance de notre Modèle (**Accuracy**)



```
print('Test set\n Loss: {:.3f}\n Accuracy: {:.3f}'.format(accur[0], accur[1]))
```

```
Test set
Loss: 0.178
Accuracy: 1.000
```

Figure 23 : les mesures d'évaluation

4. Expérience

4.1. Cadre expérimental

Nous effectuons des expériences sur deux jeux de données pour les tâches (data_with_tashkeel, data_without_tashkeel), voir le tableau ce dessous, il contient des récapitulatifs :

Jeux de données	Train	Test
Data_with_tashkeel	43	8
Data_without_tashkeel	35	15

Tableau 5 : les statistiques de notre expérience sur les deux jeux de données (data_with_tashkeel, data_without_tashkeel)

4.2. Mesure d'évaluation

Il peut être difficile de déterminer si le modèle LSTM fonctionne correctement avec les problèmes de prédiction de séquence.

Pour obtenir un bon score de compétences de modèle, il est important de savoir si notre modèle convient à nos données ou s'il est sous-ajusté ou sur-ajusté et pourrait mieux fonctionner avec une configuration différente.

4.3. Bon ajustement

Un bon ajustement est un cas où la performance du modèle est bonne à la fois sur l'entraînement et sur les ensembles de validation.

```
[ ] from keras.callbacks import EarlyStopping
    model.fit(sequences_matrix,y_train,batch_size=128,epochs=10,
              validation_split=0.2,callbacks=[EarlyStopping(monitor='val_loss',min_delta=0.0001)])
```

↳ Train on 32 samples, validate on 9 samples

Epoch	Time	Loss	Accuracy	Val Loss	Val Accuracy
Epoch 1/10	0s 4ms/step	0.2181	0.8750	0.5160	0.6667
Epoch 2/10	0s 3ms/step	0.1801	1.0000	0.3711	0.8889
Epoch 3/10	0s 4ms/step	0.1113	1.0000	0.3462	0.8889
Epoch 4/10	0s 3ms/step	0.0840	1.0000	0.3245	0.8889
Epoch 5/10	0s 4ms/step	0.0614	1.0000	0.3069	0.8889
Epoch 6/10	0s 3ms/step	0.0421	1.0000	0.2937	0.8889
Epoch 7/10	0s 3ms/step	0.0328	1.0000	0.2860	0.8889
Epoch 8/10	0s 4ms/step	0.0281	1.0000	0.2802	0.8889
Epoch 9/10	0s 4ms/step	0.0220	1.0000	0.2796	0.8889
Epoch 10/10	0s 4ms/step	0.0226	1.0000	0.2734	0.8889

<keras.callbacks.callbacks.History at 0x7f15ff41c9b0>

Figure 24 : Les statistiques de notre modèle

5. Résultats principaux

Dans le tableau ci-dessous nous montrons les principaux résultats pour l'étude comparative qui ont déjà fait dans la partie précédant avec notre modèle.

Nous utilisons les deux mesures d'évaluation (Accuracy & Loss) pour comparer Les performances des modèles

Modèle	Data_With_Tashkeel		Data_Without_Tashkeel	
	Accuracy	Loss	Accuracy	Loss
Sequential Model	0.7215	0.4826	0.8611	0.4324
Naive bayes	0.6128	0.291	0.6725	0.215
Support Vector Machine (SVM)	0.6922	0.4725	0.7821	0.3916
Notre Modèle (LSTM)	0.8667	0.2127	1.000	0.178

Tableau 6 : Les principaux résultats pour l'étude comparative

6. Conclusion

Dans ce chapitre nous présentons les différentes expérimentations de notre modèle, nous utilisons les jeux de données, nous comparons aussi notre modèle par d'autres modèles.

L'objectif principal de cette expérimentation est l'amélioration de performances, aussi pour vérifier l'efficacité de modèle LSTM.

Nous avons démontré l'efficacité de notre modèle sur deux jeux de données, nous présentons aussi les différentes mesures utilisées pour tester l'ajustement de notre modèle. Pour on peut dépasser au d'autres résultats dans des versions ultérieurs.

Conclusion Générale et Perspective

La classification de textes arabes s'est avérée au cours des dernières années comme un domaine majeur de recherche pour les entreprises comme pour les particuliers. Ce dynamisme est en partie dû à la demande importante des utilisateurs pour cette technologie. Elle devient de plus en plus indispensable dans de nombreuses situations où la quantité de documents textuels électroniques rend impossible tout traitement manuel.

La catégorisation de textes a essentiellement progressé ces dix dernières années grâce à l'introduction des techniques héritées de l'apprentissage automatique qui ont amélioré très significativement les taux de bonne classification. Il reste néanmoins difficile de fournir des valeurs chiffrées sur les performances qu'un système de classification peut actuellement atteindre.

Dans ce mémoire nous avons présenté les étapes de classification automatique de textes arabes et coraniques utilisant les réseaux de neurones récurrents de type LSTM.

En guise de conclusion, les résultats que nous avons obtenus sont encourageants.

La comparaison avec d'autres modèles en employant les mesures d'évaluation (Accuracy, Loss) montre que les résultats de l'approche proposée évaluée sur le data set (with-tashkeel, without-tashkeel) sont comme suit : (Accuracy=0.8667, Loss=0.2127), (Accuracy=1.000, Loss= 0.178)

Perspectives

le sujet étant très vaste, il reste beaucoup à faire pour améliorer ce travail, on peut donc proposer comme perspectives, d'ajouter d'autres langues pour rendre le système multi-langues, d'étendre l'éventail des formes des mots arabes pris en compte par l'analyse morphologique, d'intégrer d'autres techniques et méthodes de classification, ainsi que toute autre idée jugée utile, réalisable et bénéfique.

Dans la suite, on a l'envie de réaliser les travaux suivants :

- ✓ Comparaison entre les méthodes supervisée et non supervisé.
- ✓ Appliquer d'autres approches de représentation des textes.
- ✓ Utiliser les données de très hautes dimensions.
- ✓ La catégorisation des récitations coraniques.

Références

- [1] T.DERDRA Amel, F.BENSFIA. 2011-2012. La Représentation Conceptuelle pour la Catégorisation des Textes Multilingue. Mémoire de Master. Université Abou Bakr Belkaid– Tlemcen.
- [2] Frédéric Hourdeau. 2019. IA et machine learning : Différence entre apprentissage supervisé et apprentissage non supervisé.
- [3] Radwan JALAM. 2003. Apprentissage automatique et catégorisation de textes multilingues UNIVERSITÉ LUMIÈRE LYON2.
- [4] BENTAALLAH Mohamed Amine. 2015. Utilisation des Ontologies dans la Catégorisation de Textes Multilingues Université Djillali Liabes de Sidi Bel Abbes,.
- [5] [https://fr.wikipedia.org/wiki/Matrice de confusion](https://fr.wikipedia.org/wiki/Matrice_de_confusion)
- [6] Amélioration du produit scalaire via les mesures de similarités sémantiques dans le cadre de la catégorisation des textes Université Abou Bakr Belkaid– Tlemcen.
- [7] Mohamed Zakaria Kurdi. 2018. Traitement automatique des langues et linguistique informatique 2.
- [8] Mr. Mouhamed Amine CHERAGUI. 2016. Mise en place d'un chunker (grammaire en tronçons) des textes arabes.
- [9] Bilel Bahloul, méthode pour l'analyse automatique d'opinion de la langue arabe
- [10] ABBAS Mourad. 2019. CATEGORISATION AUTOMATIQUE DES TEXTES ARABES Université SAAD DAHLEB DE BLIDA
- [11] Fouad Soufiane Douzidia. 2004. Résumé automatique de texte arabe Université de Montréal.
- [12] Mustafa et al, M. Mustafa, H. AbdAlla, and H. Suleman. 2008 .Current Approaches in Arabic IR: A Survey. In Proceedings The Annual International Conference on Asia-Pacific Digital Libraries (ICADL), Bali, Indonesia.

- [13] Khoja et al, S. Khoja, R. Garside, and G. Knowles. 2001. A Tagset for the Morphosyntactic Tagging of Arabic". Proceedings of the Corpus Linguistics. Lancaster University (UK).
- [14] Dr. Mounir ZRIGUI Cours Traitement automatique de la langue unité de recherche RIADI, faculté des Sciences de Monastir Tunisie.
- [15] BENHALIMA MAISSA. 2017. Implémentation d'une méthode hybride (Morphologique & statistique) pour l'analyse des mots arabes. UNIVERSITE MOHAMED BOUDIAF - M'SILA.
- [16] A. Taani. 2009. A rule-based approach for tagging non-vocalized Arabic words. The International Arab Journal of Information Technology. P : 320-328.
- [17] Dhifallah OTHMEN. Etiquetage morphosyntaxique de l'arabe avec Nooj
- [18] Belguith et al, L. Hadrich Belguith, L. Baccour et M. Ghassan. 2005. Segmentation de textes arabes basée sur l'analyse contextuelle des signes de ponctuations et de certaines particules. Actes de la 12ème conférence sur le Traitement Automatique des Langues Naturelles TALN'2005 - Dourdan France, vol. 1, pages 451–456.
- [19] Amirouche Radia. UNE COMBINAISON DE CLASSIFIEURS POUR LA RECONNAISSANCE DES VISAGES HUMAINS UNIVERSITE BADJI – MOKHTAR – ANNABA
- [20] J.R. Quinlan 1986. Induction of Decision Trees, Machine Learning. 81-106
- [21] Quinlan, J. R. 1993. Programs for Machine Learning, Morgan Kaufmann, San Mateo, California.
- [22] Touina Hanane. Classification automatique de textes UNIVERSITE MOHAMED BOUDIAF - M'SILA
- [23] SIMON REHEL. Janvier 2005. Catégorisation automatique de textes et cooccurrence de mots provenant de documents non étiquetés. Mémoire présenté à la Faculté des études supérieures de l'Université Laval. Québec.
- [24] Harry Zhang. 2004. The Optimality of Naive Bayes . Conférence FLAIRS

- [25] Caruana.R & Niculescu-Mizil.A. 2006. An empirical comparison of supervised learning algorithms. Proceedings of the 23rd international conference on Machine learning.
- [26] Jozefowicz.R , Zaremba.W & Sutskever.I. 2015. An empirical exploration of recurrent network architectures. In Proceedings of the 32nd International Conference on Machine Learning (ICML-15), pages 2342–2350.
- [27] Karpathy.A , Johnson.J & Fei-Fei.L. 2015. Visualizing and understanding recurrent networks. arXiv preprint arXiv :1506.02078.
- [28] Hochreiter.S & Schmidhuber.J. 1997. Long short-term memory. Neural computation.
- [29] Karpathy.A. 2015. The unreasonable effectiveness of recurrent neural networks. Andrej Karpathy blog.