UNIVERSITE SAAD DAHLEB DE BLIDA

Faculté des sciences

Département de l'informatique

MEMOIRE DE FIN D'ETUDES

Master en informatique

Traitement automatique de la langue

CATEGORISATION AUTOMATIQUE DES TEXTES ARABES

Par

Ahmed ZEGGADA

Rabah MOULAI

Devant le jury composé de :

- Mme. OUKID	Enseignant, U. de Blida	Président
- Mr. CHERIF ZAHAR	Enseignant, U. de Blida	Examinateur
- Mr. ABBAS Mourad		Encadreur

.

Blida, Octobre 2019


بسم الله الرحمن الرحيم

DEDICACE

À nos très chers parents,

Nul mot, Nulle dédicace ne pourra suffisamment exprimer tout notre respect, notre considération et notre amour pour les sacrifices auxquels vous avez tant consentis pour notre instruction.

Pour votre générosité et votre bonté.

Et dont le présent ce travail en est une modeste récompense.

Nous vous exprimons toute notre affection

Nous dédicaçons tout aussi sincèrement notre mémoire

À toute notre chère famille,

À nos chers professeurs,

À nos chers amis,

À nos chers collèges,

À tous ceux qui nous ont aidés de près ou de loin,

Nous dédions ce modeste travail

REMERCIEMENTS

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]

Aussi,

Nous remercions Dr. Abbas Mourad ainsi que Dr. Lichouri Mohamed de nous avoir donné l'opportunité d'accomplir cette œuvre par leurs conseils judicieux.

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

Enfin, merci à nos familles, nos chers mères et pères pour le soutien et leurs encouragements qu'ils n'ont eu de cesse de nous apporter.

RESUME

Notre travail décrit un système de classification des poèmes arabes en fonction des époques dans lesquelles ils ont été écrits. 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 MNB (Multinomial Naïve Bayes), avec une exactitude de l'ordre de 70,21%, un score F1 de 68,8% et un Kappa égal à 0,398, cela sans extraire les mots vides. Nous avons observé que les mots vides peuvent avoir un impact positif sur la précision et inversement un impact négatif s'ils sont utilisés avec la technique de "Word Tokenizer" dans la phase de prétraitement.

Mot clés : Classification des textes, Langue arabe, Poèmes, Identification des ères, Mots vides, Word Tokenizer. Ngram Tokenizer

ABSTRACT

This paper describes a system for classification of Arabic poems according to the eras in which they were written. We used machine learning techniques where we applied a bunch of filters and classifiers. The best results were achieved by using the Multinomial Naive Bayes (MNB) algorithm, with an accuracy equal to 70.21%, and F1-Score of 68.8% and a Kappa equal to 0.398, without filtering stop words. We observed that the stop words can have a positive impact on the accuracy but also a negative impact if it is used with word tokenizer pre-processing.

Keywords: Text Classification, Arabic Language, Poems, Eras Identification, Stop words, Word Tokenizer Ngram Tokenizer.

ملخص

يصف عملنا نظام تصنيف القصائد العربية وفقًا العصور التاريخية التي كتبت فيها. استخدمنا تقنيات التعلم الآلي التي طبقناها على العديد من المرشحات والمصنفات. تم الحصول على أفضل النتائج باستخدام خوارزمية (Naive Bayes MNB Multinomial) ، مع دقة تساوي 70.21 ٪ ، ودرجة F1 68.8 ٪ KAPPA تساوي 0.398 ، وكل ذلك دون استخراج الكلمات فارغة. لقد لاحظنا أن كلمات التوقف يمكن أن يكون لها تأثير إيجابي على الدقة ولكن أيضًا لها تأثير سلبي إذا تم استخدامها مع تقنية word tokenizerفي مرحلة ما قبل المعالجة

الكلمات المفتاحية: تصنيف النصوص ، اللغة العربية ، القصائد ، تحديد العصور ، الكلمات المتوقفة ،

Word Tokenizer Ngram Tokenizer.

TABLE DES MATIÈRES

Table des matières

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ABREVIATIONS

CT : *Catégorisation de texte.*

TAL : *Traitement automatique des langues.*

TALN : *Traitement automatique des langues naturelles.*

TALA : *Traitement automatique de la langue arabe.*

RI : *Recherche d'information.*

SVM : *Machines à support de vecteurs. (Support Vector Machine)*

RNA : *Réseaux de neurone artificiel.*

NB : *Naïve Bayes.*

NBM : *Naïve Bayes Multinomial.*

KNN : *K-nearest neighbors.*

BAG *: Bagging Classificateur.*

MCC *: Multi Class Classificateur.*

RF : *Random Forest Classificateur.*

WEKA : *Waikato Environment for Knowledge Analysis.*

ARFF : *Attribute-Relation File Format.*

PI : *Ere Préislamique.*

OM : *Ere Omeyyade.*

AB: *Ere Abbasside.*

AN : *Ere Andalouse.*

OJDBC : *Oracle Java Database connectivity.*

UTF-8 : *Universal Character Set Transformation Format - 8 bits).*

Liste des abréviations

GUI : *Graphical User Interface.*

La révolution de l'information bousculée par le développement à grande échelle de l'Internet/Intranet a fait exploser la quantité d'informations textuelles disponibles. De même que la vulgarisation de l'informatique dans le monde, a permis de créer d'importants volumes de documents électroniques rédigés en différentes langues dont une grande partie est rédigée en langue arabe. Ce qui a mené, les travaux de recherche à aborder des problématiques plus variées telles que la traduction automatique ; la recherche de l'information et la catégorisation automatique des textes

Le projet "DAKHIRA AL Arabiya", est une initiative de la ligue arabe fondée sur le principe de la participation d'institutions scientifiques et culturelles de chaque pays membre, dont le centre de recherche Scientifique et Technique pour le Développement de la Langue Arabe (CRSTDLA) qui apporte sa pierre à cette audacieuse idée qui est une banque de données textuelles anciennes et modernes couvrant le patrimoine culturel arabe.

Notre travail, qui se veut une contribution à ce projet, consiste en la catégorisation automatique des textes arabes, particulièrement des poèmes arabes. Pour ce faire, nous nous appuyons sur un corpus de poèmes arabes mis à notre disposition par l'établissement CRSTDLA, dans un but de classification selon différentes ères (*Préislamique, Omeyade, Abbasside, Andalouse*). Ce corpus consiste en un recueil comportant 30866 poèmes dont 382000 vers. A cette échelle, il s'agit de la toute première étude couvrant un volume anthologique de pareille importance.

L'utilisation de WEKA, prescrit dans le cadre de nos travaux, a abouti à des résultats satisfaisants qui ont fait l'objet d'intérêt lors d'ateliers internationaux PATRECH auxquels nous fument invités par l'université de Trento (Italie) pour présenter ces dits-travaux.

Il est également prévu une publication dans la revue scientifique internationale « The journal of ACM » (Association for Computing Machinery), revue éminemment estimée par la communauté informatique universitaire internationale.

Notre mémoire est ainsi organisé :

- Un premier chapitre intitulé "*Catégorisation et classification automatique*

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 d'une interface de prédiction de catégorisation.
-

Chapitre-I

Catégorisation

et

Classification automatique des textes

a.

I.1 -Introduction

La révolution de l'information bousculée par le développement à grande échelle des accès réseaux Internet/Intranet a fait exploser la quantité d'informations textuelles disponibles en ligne ou hors ligne ; de même que la vulgarisation de l'informatique dans le monde des entreprises, des administrations et chez les particuliers, a permis de créer d'importants volumes de documents électroniques rédigés en langue naturelle, générant ainsi de nouvelles problématiques pour lesquelles l'apprentissage statistique ne possède pas de réponses.

D'autre part, les limites d'une approche manuelle, coûteuse en temps de travail, peu générique, et relativement peu efficace, ont motivé la recherche dans le domaine de la classification automatique de textes, objet de notre travail,

Mais alors, comment partitionner cette masse d'information en groupes ou classes afin d'en dégager des ressemblances par thème, par auteurs, par langue, ou par d'autres critères de classification.

C'est en ces termes que peut être formulée la problématique de classification de textes.

I.2 – Définition

La catégorisation de textes (C.T) (- également connue sous le nom de classification de textes) consiste à trouver/établir une liaison fonctionnelle entre un ensemble de textes et un ensemble de catégories (étiquettes, classes). Cette liaison fonctionnelle, que l'on appelle également modèle de prédiction, est estimée par un apprentissage automatique (machine Learning method). Pour ce faire, il est nécessaire de disposer d'un ensemble de textes préalablement étiquetés, dit ensemble d'apprentissage, à partir duquel nous estimons les paramètres du modèle de prédiction le plus performant, c'est-à-dire le modèle qui produit le moins d'erreurs de prédiction.

Formellement, la catégorisation de textes consiste à associer une valeur booléenne à chaque paire (dj , ci) ∈D×C, où "D" est l'ensemble des textes et "C" est l'ensemble des catégories. La valeur V (Vrai) est alors associée au couple (dj , ci) si le texte dj appartient à la classe ci 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) Φ : D×C → {V,F} qui associe une ou plusieurs catégories à un document dj tel que la décision donnée par cette procédure « coïncide le plus possible » avec la fonction Φ : ˇ D×C → {V,F}, la vraie fonction qui retourne pour chaque vecteur dj une valeur ci (Sebastiani, 2002)[1].

I.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 à 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.

I.3.1–La classification supervisée

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 a vecteur de support, K voisins Proches, Arbre de Décision…

Notre travail repose principalement sur la catégorisation de texte en utilisant la méthode de Classification supervisée.

I.3.2–La classification non supervisée

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]

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

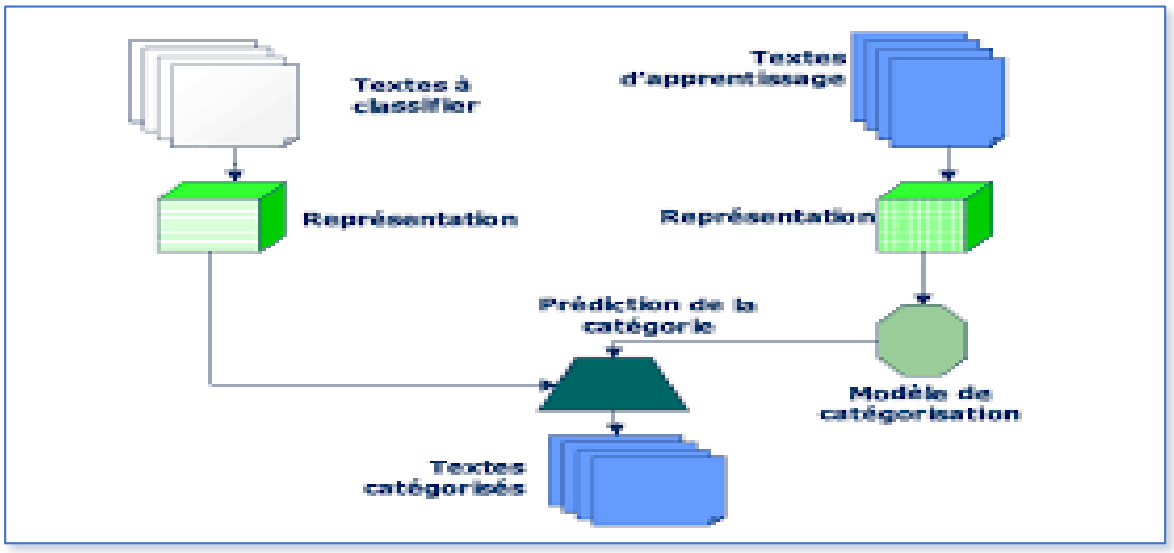


Figure1 1. Processus de la catégorisation de textes

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 [4].

I.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 :

I.4.1.1–Représentation en sac de mots (Bag of words en Anglais) :

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.

I.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 certaine 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.

I.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.

I.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.

I.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 [5].

I.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

I.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 Ci, 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.

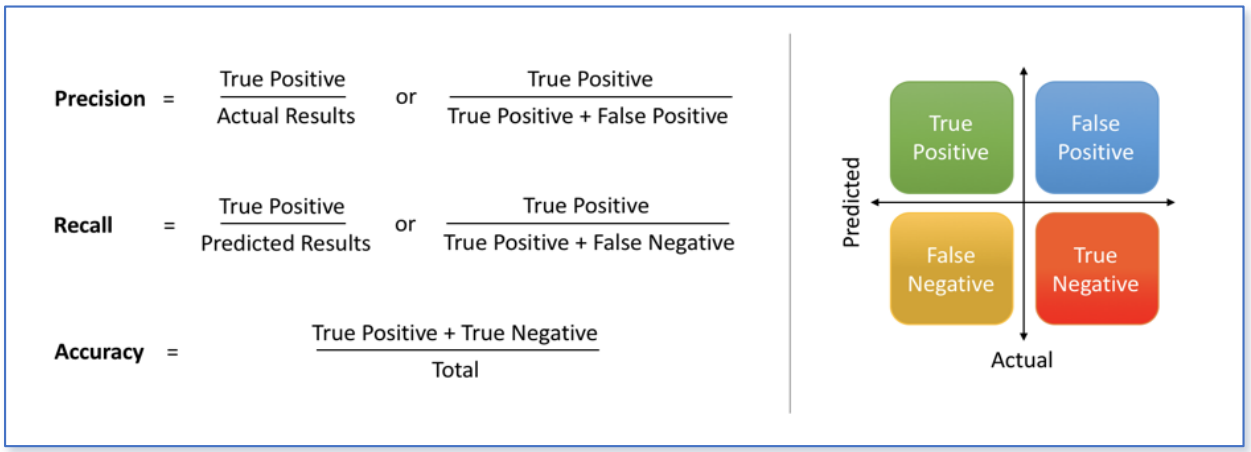


Figure-1.2. Mesures d'évaluation de Rappel, Précision et Exactitude

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) :

Exactitude = (VP + TN) / (VP + VN + FP + FN)

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

F= 2*(P*R) / (P +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 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 C1[6].

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

I.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 poèmes arabes selon leurs époques d'apparition entre le 5eme et le 15eme siècle.

I.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'ambigüité, 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 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 [7].

I.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-II

Traitement Automatique de la Langue Arabe (TALA)

II.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 [8].

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.

II.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.

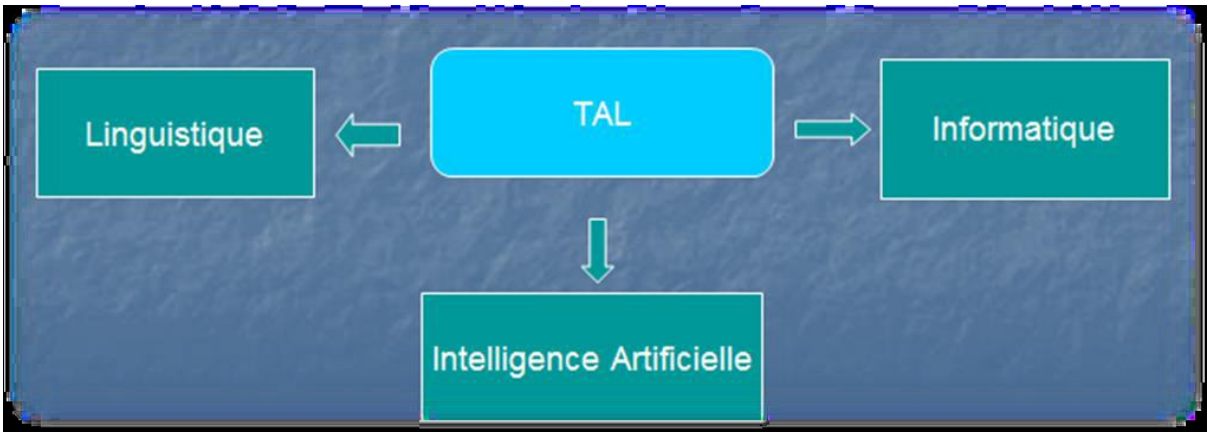


Figure 2.1 : Présente La Pluridisciplinaire De TAL

La réalisation de n'importe quelle application parmi celles citées précédemment passe principalement par différent 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[9].

II.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 02 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.

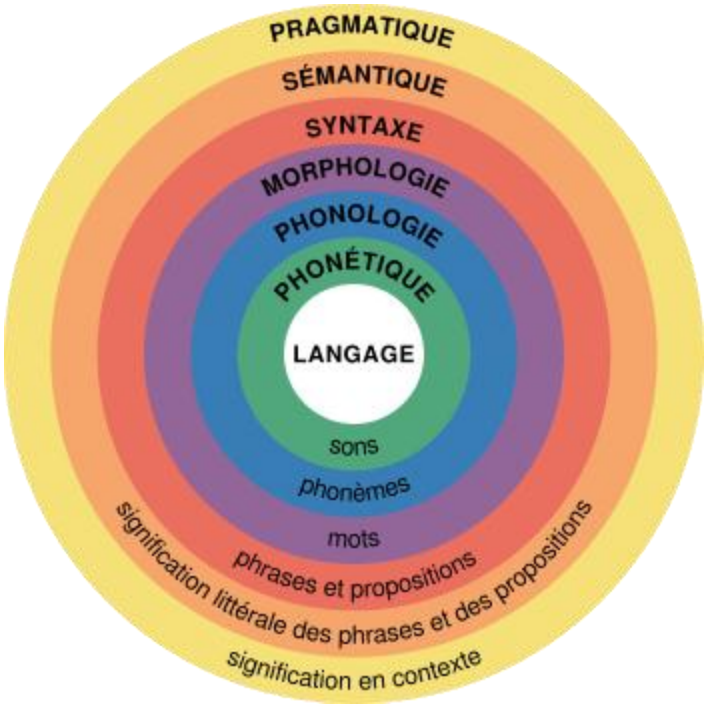


Figure 2.2: Les niveaux de traitement

II.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 [10].

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.

II.5–La langue arabe :

L'arabe (en arabe : العربية, al-ʿarabīya') est une langue afro-asiatique de la famille des langues sémitiques. Avec un nombre de locuteurs estimé entre 315 et 375 millions de personnes au sein du monde arabe et de la diaspora arabe. L'arabe est de loin la langue sémitique la plus parlée.

La langue arabe est originaire de la péninsule arabique, où elle devint au VIIe siècle la langue du Coran et la langue liturgique de l'islam. L'expansion territoriale de l'empire arabe au moyen âge a conduit à l'arabisation au moins partielle sur des périodes plus ou moins longues du Moyen-Orient, de l'Afrique du Nord et de certaines régions en europe (péninsule Ibérique, Sicile, Crète, Chypre, territoires où elle a disparu, et Malte, où le maltais en constitue un prolongement particulier).

Parlée d'abord par les arabes, cette langue qui se déploie géographiquement sur plusieurs continents s'étend sociologiquement à des peuples non arabes, et est devenue aujourd'hui l'une des langues les plus parlées dans le monde. C'est la langue officielle de plus de vingt pays et de plusieurs organismes internationaux, dont l'une des six langues officielles de l'Organisation des Nations Unies [11].

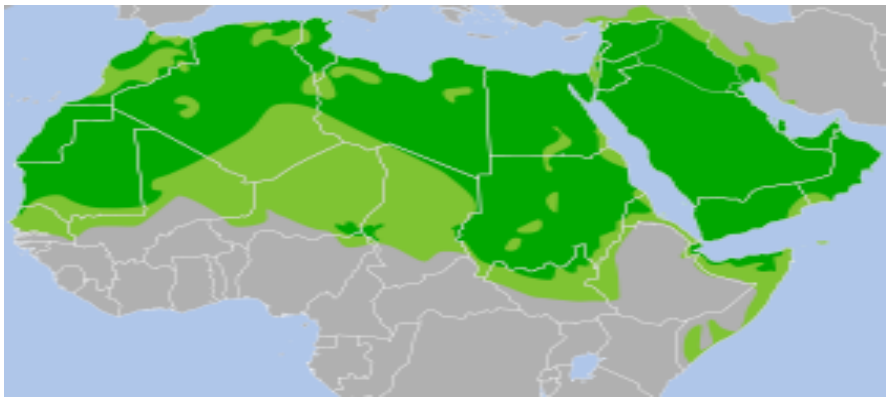


Figure 2.3: Répartition géographique de la langue arabe

II.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) [12]. 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.

II.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)[13]. 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 2.1: 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 (ب، ت، ك) 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 «لَعَّف » 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) [14].

- 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 « مفعل ».

- 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 ?»

II.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 :تذكر dérivé de la racine (ذ ك ر) selon le schème تَفعل
- Suffixe : " ون" suffixe verbal exprimant le pluriel.
- Enclitique : "نا" pronom suffixe.

Cet exemple montre la richesse morphologique de la langue arabe [15]. 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.

II.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 04 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 [16].

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 .grandes كبيرات

- Le masculin pluriel : Pour le pluriel masculin on rajoute les deux lettres ين ou ونdépendamment de la position du mot dans la phrase.

Exemple : الراجع revenant devient .revenants الراجعون ou الراجعين

- 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 (في، على، حتى، من) et les conjonctions (بل، أم، أو، ثم) qui sont utilisés pour lier des noms, des verbes ou des phrases (Taani et al. 2009) [17]. 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.

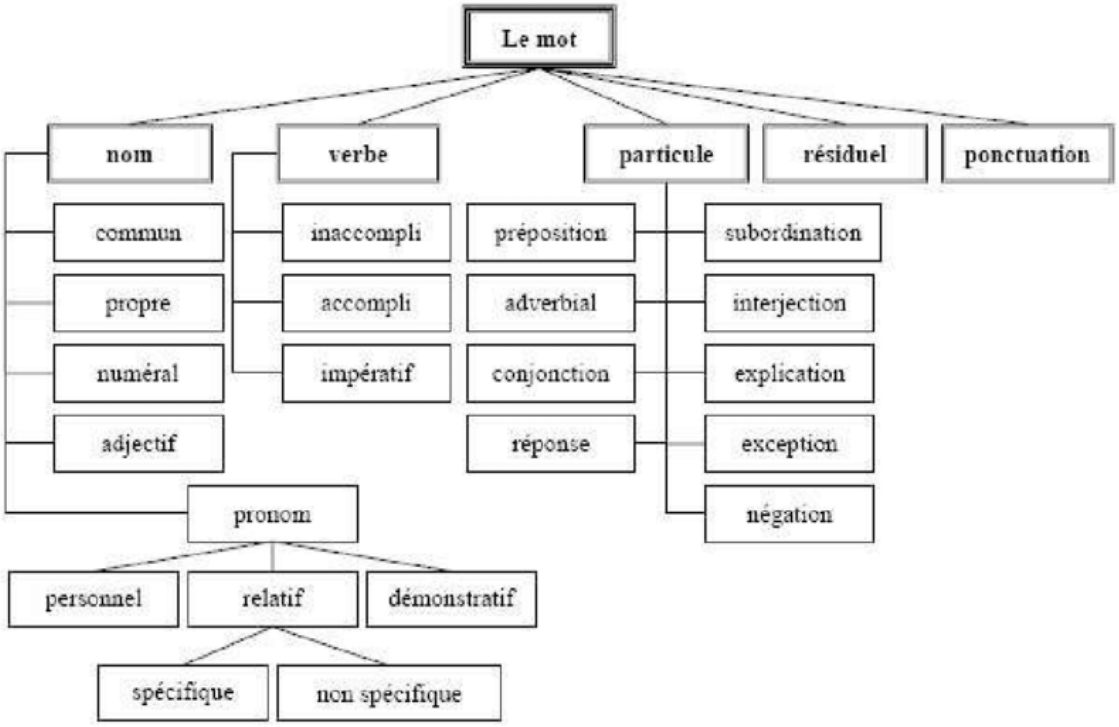


Figure 2.4 : Classification des unités lexicales

II.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.

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 [18].

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 =

koutoube "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.

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'ambigüité et des problèmes lors de l'analyse automatique.

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) [19], 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.

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 ambigüités 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 "وجع "est un caractère qui fait partie de ce mot alors que dans le mot " وحصل,"il s'agit d'un proclitique.

II.7. 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'ambigüité, etc.

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

Chapitre-III

Les Classificateurs

III.1 Introduction

La recherche accorde ces dernières années, beaucoup d'importance au traitement des données textuelles. Ceci pour plusieurs raisons : un nombre croissant de collections mises en réseau et distribuées au plan international, le développement de l'infrastructure de communication et de l'Internet. Les traitements manuels de ces données s'avèrent très coûteux en temps et en personnel, ils sont peu flexibles et leur généralisation à d'autres domaines est presque impossible ; c'est pour cela que l'on cherche à mettre au point des méthodes automatiques.

A l'heure actuelle, de nombreux logiciels de classification de textes sont disponibles, ils ont fait l'objet de publications et leurs champs d'application s'élargissent de jour en jour.

En général, ces systèmes sont basés sur des algorithmes d'apprentissage automatique, Nous présentons donc des méthodes d'apprentissage qui, à partir de documents déjà classés, permettent de classer de nouveaux documents.

III.2 Algorithmes d'apprentissage

En apprentissage automatique, différents types de classificateurs ont été mis au point, et cela dont le but d'atteindre un degré maximal de précision et d'efficacité, chacun ayant ses avantages et ses inconvénients. Mais, ils partagent toutefois des caractéristiques communes.

Parmi la panoplie de classificateurs existants, on peut faire des regroupements et distinguer des grandes familles. Notre travail de classification s'étant effectué à l'aide du Logiciel WEKA, nous avons opté pour les nomenclatures des familles et des algorithmes de WEKA.

Dans les pages qui suivent, nous allons exposer en détail quelques algorithmes que nous avons utilisés, le classificateur bayésien naïf et particulièrement le multi-nomial (NBM) algorithme que nous avons utilisé dans notre étude, surpassé par d'autres mais souvent utilisé comme point de référence en raison de sa simplicité et des bons résultats que nous avons obtenus pour notre classification des poèmes arabes.

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) ou SMO dans WEKA.
- Les réseaux de neurones(RNA).
- L'algorithme de Naïve Bayes.
- Multinomial Naïve Bayes.

III.2.1 Algorithme des k-voisins les plus proches KNN

III.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 [20].

III.2.1.2 principes 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 :

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

La figure suivante illustre le fonctionnement de l'algorithme KNN :

Paramètre : le nombre K de voisins
Contexte : un échantillon de L textes classés en C = c_1, c_2, ..., c_n classes
Début
  Pour chaque texte T faire
  Transformer le texte T en vecteur T = (x_1, x_2, ..., x_m),
    Déterminer les K plus proches textes du texte T selon une métrique de distance,
  Combiner les classes de ces K exemples en une classe C.
  Fin pour

Figure 3.1. 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 :

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

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

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

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

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

III.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.

III.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.

III.2.2. Les arbres de décision :

III.2.2.1. Définition :

Les arbres de décision sont les plus populaires des méthodes d'apprentissage. Les Algorithmes connus sont ID3 (Quinlan 1986)[21] et C4.5 (Quinlan 1993)[22] appelé dans WEKA sous le nom J48. 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.

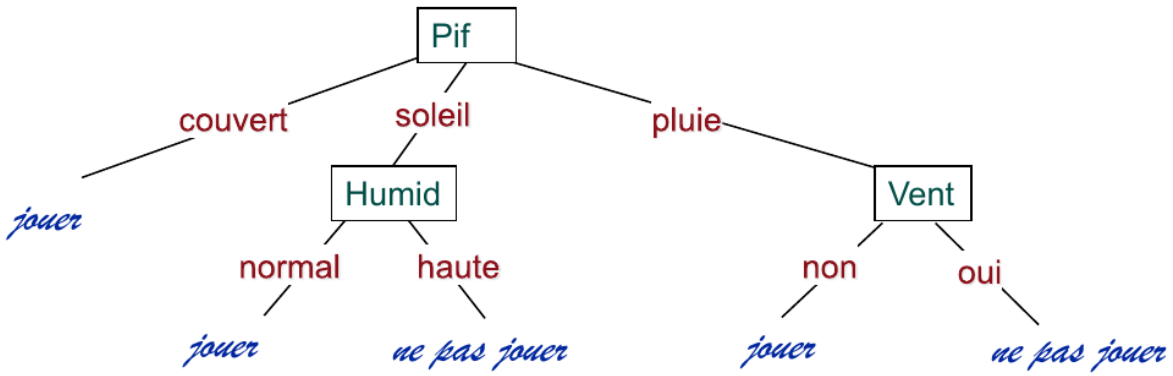


Figure3. 2. Exemple d'arbre de décision

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..

III.2.2.2. Algorithme :

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

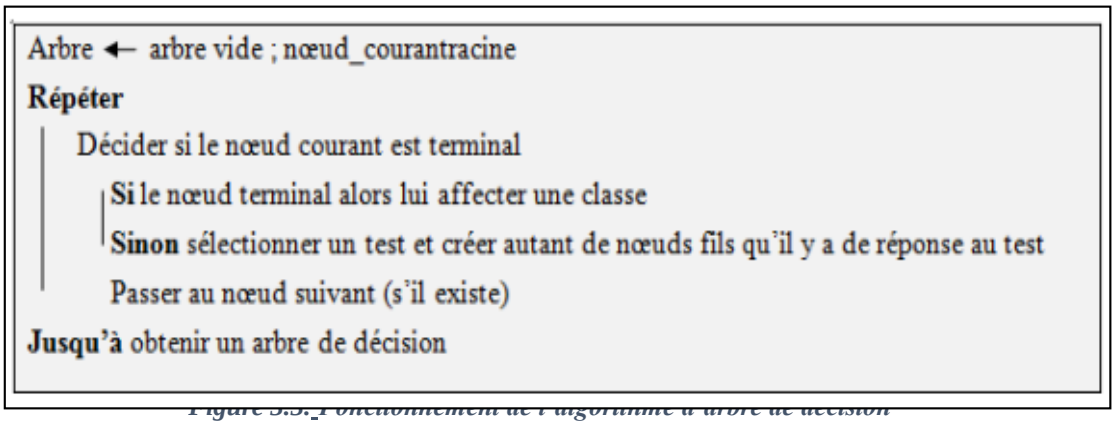


Figure 3.3. Fonctionnement de l'algorithme à arbre de décision

III.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.

III.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).

III.2.3. Machines à support de vecteurs (ou SVM)

Les machines à support de vecteurs (SVM, SMO dans WEKA) sont à l'origine de nouvelles méthodes de catégorisation, bien que les premières publications sur le sujet datent des années 60.[23]

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 mainte 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 :

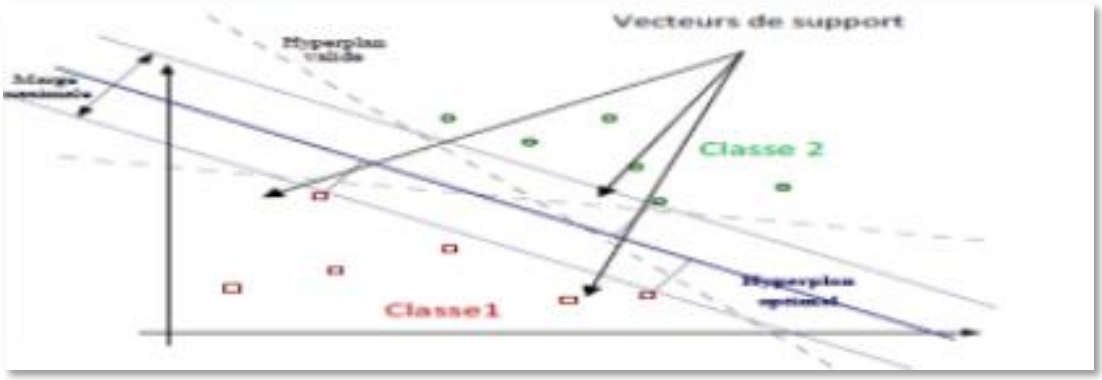


Figure 3.4. Vecteurs de support machines

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 classifie 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 [10].

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.

III.2.4 Réseaux de neurones

Un réseau de neurones (NeuralNetwork dans WEKA) 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 .Ils 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.[24]

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 Ni neurones, prenant leurs entrées sur les Ni-1 neurones de la couche précédente. À chaque synapse est associé un poids synaptique, de sorte que les Ni-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.

III.2 .5 Classification naïve bayésienne

La classification naïve bayésienne (NaiveBayes dans WEKA) 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 ». [25]

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.

III.2.5.1 Description du modèle Bayésienne

Le modèle probabiliste pour un classificateur est le modèle conditionne ($\boldsymbol{C}|\boldsymbol{F1}$, … … . . , $\boldsymbol{F\mathfrak{y}}$) où "C" est une variable de classe dépendante dont les instances ou classes sont peu nombreuses, conditionnée par plusieurs variables caractéristiques$\boldsymbol{F1}$, … … . . , $\boldsymbol{F\mathfrak{y}}$.

Lorsque le nombre de caractéristiques $\mathfrak{y}$ est grand, ou lorsque ces caractéristiques peuvent prendre un grand nombre de valeurs, baser ce modèle sur des tableaux de probabilités devient impossible. [26]

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

$$\mathrm{p(C|F1, \dots \dots \dots, F\mathfrak{y}\ )} = \frac{p(C)\ \ p(\mathrm{F1, \dots \dots \dots, F\mathfrak{y}\ |\ C})}{\mathrm{p(F1, \dots \dots \dots, F\mathfrak{y}\ )}}$$

En langage courant, cela signifie :

$$\text{postérieure} = \frac{\text{antérieure} \times \text{vraisemblance}}{\text{évidence}}$$

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

$$\boldsymbol{p}(\boldsymbol{C},\boldsymbol{F1}, \dots \dots \dots, \boldsymbol{F\mathfrak{y}}\ )$$

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

$$p(C, F_1, \ldots, F_n)$$

$$= p(C)\, p(F_1, \ldots, F_n|C)$$

$$= p(C)\, p(F_1|C)\, p(F_2, \ldots, F_n|C, F_1)$$

$$= p(C)\, p(F_1|C)\, p(F_2|C, F_1)\, p(F_3, \ldots, F_n|C, F_1, F_2)$$

$$= p(C)\, p(F_1|C)\, p(F_2|C, F_1)\, p(F_3|C, F_1, F_2)\, p(F_4, \ldots, F_n|C, F_1, F_2, I$$

$$= p(C)\, p(F_1|C)\, p(F_2|C, F_1)\, p(F_3|C, F_1, F_2)\, \ldots p(F_n|C, F_1, F_2, F_3, \ldots$$

C'est là que nous faisons intervenir l'hypothèse naïve : si chaque $\boldsymbol{Fi}$ est indépendant des autres caractéristiques$\boldsymbol{yi} \neq \boldsymbol{j}$ R alors

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

$$p(F_i|C, F_j) = p(F_i|C)$$

$$p(C, F_1, \ldots, F_n) = p(C)\, p(F_1|C)\, p(F_2|C)\, p(F_3|C)\, \cdots$$

$$= p(C) \prod_{i=1}^{n} p(F_i|C).$$

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

$$p(F_i|C, F_j) = p(F_i|C)$$

$$p(C, F_1, \ldots, F_n) = p(C)\, p(F_1|C)\, p(F_2|C)\, p(F_3|C)\, \cdots$$

$$= p(C) \prod_{i=1}^{n} p(F_i|C).$$

où Z (appelé « évidence ») est un facteur d'échelle qui dépend uniquement de $\boldsymbol{F1}$, … … . . , , 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(Fi|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.

III.2.5.2 Estimation de la valeur des paramètres

Tous les paramètres du modèle (probabilités a priori des classes et lois de probabilités associées aux différentes caractéristiques) peuvent faire l'objet d'une approximation par rapport aux fréquences relatives des classes et caractéristiques dans l'ensemble des données d'entraînement. Il s'agit d'une estimation du maximum de vraisemblance des probabilités. Les probabilités a priori des classes peuvent par exemple être calculées en se basant sur l'hypothèse que les classes sont équiprobables (i.e. chaque antérieure = 1 / (nombre de classes)), ou bien en estimant chaque probabilité de classe sur la base de l'ensemble des données d'entraînement (i.e. antérieure de C= (nombre d'échantillons de C) / (nombre d'échantillons total)).

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 xi est la valeur d'un échantillon donné. La variance, σ^2, se calcule avec

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

Si, pour une certaine classe, une certaine caractéristique ne prend jamais une valeur donnée dans l'ensemble de données d'entraînement, alors l'estimation de probabilité basée sur la fréquence aura pour valeur zéro. Cela pose un problème puisque l'on aboutit à l'apparition d'un facteur nul lorsque les probabilités sont multipliées. Par conséquent, on corrige les estimations de probabilités avec des probabilités fixées à l'avance.

III.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 (Naive Bayes Multinomial dans WEKA) nous indique simplement que chaque p (fi | 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 Naive Bayes est un terme général qui désigne l'indépendance conditionnelle de chacune des caractéristiques du modèle, tandis que le classificateur Naive Bayes multinomial est une instance spécifique d'un classificateur Naive Bayes qui utilise une distribution multinomiale pour chacune des caractéristiques. Le classificateur correspondant à cette règle est la fonction "classificateur" suivante :

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

III.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'autre raisons pour le succès empirique des classificateurs bayésiens naïfs.

III.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 raison et les avantages qu’on a mentionné pour chaque méthode.

Chapitre-IV

Expérimentations

et

Implémentation

IV.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 de la poésie arabe 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 de poésies selon leurs différentes époques d'apparition notamment :

- ✓ ***L'époque préislamique (PI)*** : de 478 à 624 après JC
- ✓ ***L'époque omeyyade (OM)*** : de 625 à 750 après JC
- ✓ ***L'époque abbasside. (AB)*** : de 750 après JC jusqu'à 1258
- ✓ ***L'époque andalouse (AN) :*** la plus longue. Elle s'étend sur presque 8 siècles : de 750 à 1492

Par ailleurs, nous avons essayé de créer une interface dans le but de faire des prédictions sur ces mêmes poèmes. De plus, nous avons tenu à démontrer l'impact des mots vides et des techniques de tokenisations soit le "WORD_TOKENIZER et NGRAM" sur la performance de la catégorisation. Plusieurs expériences utilisant 08 différents classificateurs ont été appliquées tels que :

- ✓ K-voisins les plus proches connu dans WEKA sous la dénomination d'Instance Basic Learner (IBK).
- ✓ Machine à support de vecteurs (SVM).
- ✓ Algorithmes d'arbres de décision comme J48 et forêt aléatoire (RF).
- ✓ Bagging (BAG) et Multi Class Classifier (MCC) de la famille Meta Classificateurs.
- ✓ Naïve Bayes (NB) et Multinomial Naïve Bayes (NBM).

IV.2–Corpus

La poésie arabe est la forme la plus ancienne et la plus importante de la littérature arabe d'aujourd'hui. La poésie arabe antique constitue probablement la principale

source de la description de la vie sociale, politique et intellectuelle dans le monde arabe.

La poésie a traversé des changements majeurs à la fois dans sa forme et dans les sujets.

Notre corpus de la poésie arabe, utilisé tout au long de notre travail, comprend plus de 30K poèmes allant du 6ème au 21ème siècle. Il comprend également des métadonnées de poèmes telles que "nom de poète", "période du poème" et "sa catégorie". Nous avons téléchargé cet ensemble de données depuis "Kaggle3". Ces données-là ont été extraites du site "www.adab.com". Etant donné que notre travail principal est centré sur le problème de la classification, nous avons choisi de ne conserver qu'une fraction de l'ensemble de données correspondant à quatre époques :

L'ère Préislamique (PI), l'ère Omeyyade (OM), l'ère Abbasside (AB) et l'ère Andalouse (AN). Les statistiques du Data set obtenu sont présentées dans le tableau suivant :

	Ère Préislamique	Ère Omeyyade	Ère Abbasside	Ère Andalouse	TOTAL
# Poèmes	1 461	3 700	19 410	6 295	**30 866**
# Sentences (Baytes)	14 977	41 137	236 722	89 514	**382 350**
# Sentences Unique	14 906	41 062	235 503	89 304	**380 600**
# Mots	143 878	395 059	2 282 718	868 145	**3 689 800**
# Mots Unique	46 065	86 434	249 124	128 993	**323 995**

Tableau 4.1.: Statistiques du Corpus utilisés

La grande particularité du corpus ayant servi dans nos expérimentations est qu'il comporte un nombre exceptionnel de poèmes et qu'il s'agit de la toute première étude de classification d'un nombre aussi important de poèmes arabes.

IV.3–Présentation de l'outil WEKA

WEKA (WAIKATO Environment for Knowledge Analysis) est un outil de fouille de données (licence GNU) développé en Java. [27] Il a été créé à l'université de Waikato, en Nouvelle-Zélande, par un groupe de chercheurs issus de l'apprentissage automatique, de la reconnaissance de formes et de la fouille de données.**Voir Figure 1**

WEKA permet de prétraiter des données (onglet Preprocess dans l'interface graphique), faire de la classification supervisée (Classify) et non-supervisée (Cluster), des régressions (Select Attributes), rechercher des règles d'association (Associate), et de visualiser différentes représentations graphiques des données (Visualize). [28]

Il s'agit d'un logiciel « open source » gratuit dédié à la classification et à la fouille de données. Il s'adresse à deux types de publics. D'un côté, il présente une interface graphique, le rendant ainsi accessible à une utilisation de type « chargé d'études » sur des données réelles. De l'autre, du fait que le code source est librement disponible et l'architecture interne très simplifiée, il se prête à une utilisation de chercheurs qui veulent avant tout expérimenter de nouvelles techniques en améliorant celles déjà implémentées ou en introduisant de nouvelles.

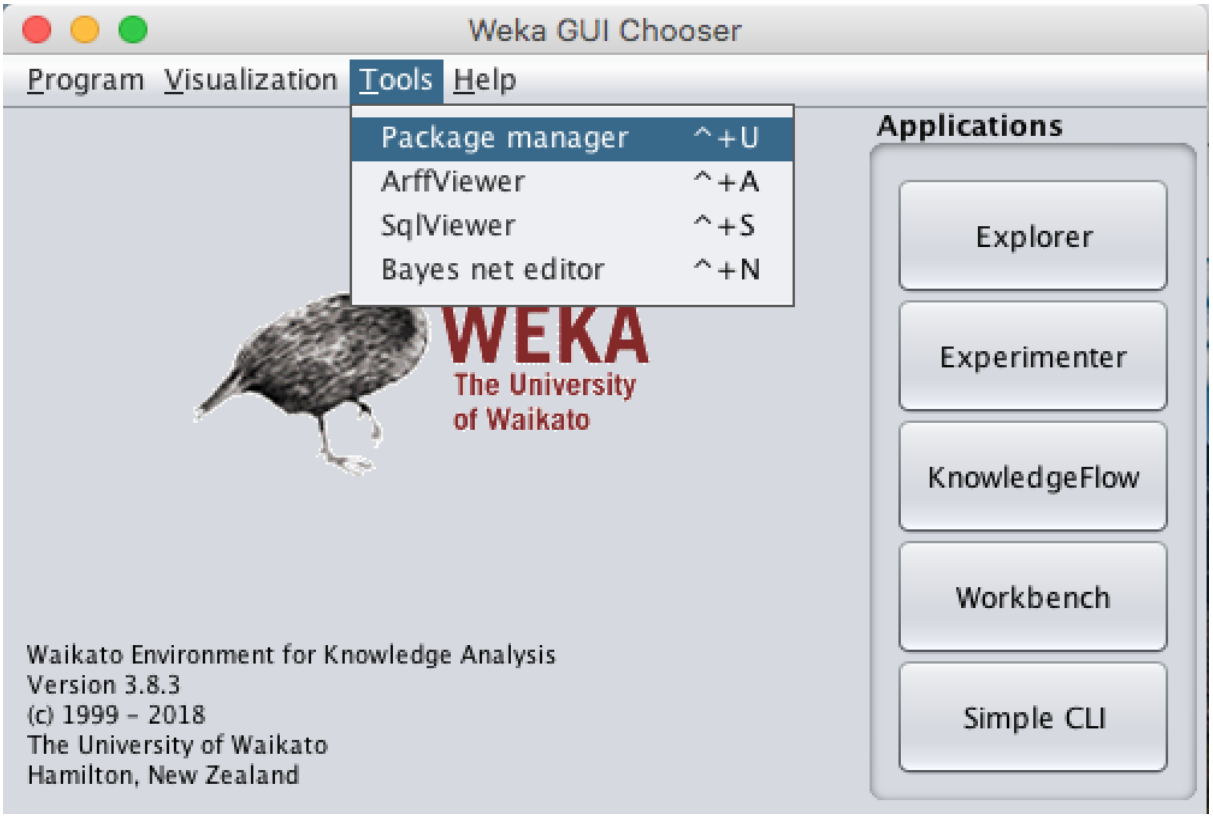


Figure 4.2 Interface graphique de WEKA

IV.3.1 **Structure de données :**

WEKA traite des données contenues dans des fichiers respectant le format ARFF Attribute-Relation File Format. Il s'agit de fichiers de type texte, décrivant des ensembles de "tuples" caractérisés par un certain nombre d'attributs communs.

IV.3.2 **Caractéristiques principales:**

· Plus de 49 outils de prétraitement de données.

· Plus de 76 algorithmes de classification/régression regroupés en 07 familles.

· Plus de 8 algorithmes de "clustering".

· Plus de 15 évaluateurs d'attributs et plus de 10 algorithmes de recherche pour la sélection d'attribut.

· 3 algorithmes de recherche de règles d'association.

· 3 interfaces graphiques GUI.

- « Explorer» (explorateur d'analyse de données).

- « Expérimenter » (environnement expérimental) .

- «KnowledgeFlow » (le nouveau modèle de processus avec interface).

IV.4–Le Prétraitement

Cette phase consiste à convertir le corpus en un format approprié en supprimant toutes les informations sans valeurs significatives tout en passant notre corpus de poésie à travers un processus de nettoyage des données et suppression de tous les signes de ponctuation, les signes diacritiques, les chiffres et les lettres non arabes. En plus de normaliser certaines formes d'écriture telles que le remplacement de أ إ آ par ا et aussi ي ىء par ى, le ة par ه, etc. Ce sont les formes d'écriture qui sont majoritairement adoptées.

En fin, il est nécessaire d'encoder notre corpus de texte au format UTF-8.

Nous avons dû faire face à quelques difficultés pour charger tous les poèmes dans WEKA. De plus, et afin de conserver notre corpus pour de futures classifications, nous l'avons chargé dans une base de données Oracle en important tout le corpus dans plusieurs tables relationnelles, et créant une connexion OJDBC avec WEKA.(**Voir figure 2)**

Dans le but de tirer pleinement profit des différents paramètres et fonctionnalités de WEKA nous avons appliqué les deux (02) filtres suivants :

- **NominalToString** filtre: ce filtre assure la conversion d'un attribut nominal à une chaine de caractère.
- **StringToWordVector** filtre : permet la conversion des attributs de forme chaine de caractères en un ensemble d'attributs numériques représentant les informations d'occurrence de mot à partir du texte contenu dans ces chaînes de caractères.

Après cela, il devenait possible d'appliquer les différentes techniques de prétraitement de WEKA comme les mots vides "STOWORDSHANDLER" et les méthodes de segmentations comme le "Word tokenizer" ainsi que le "Ngram Tokenizer".

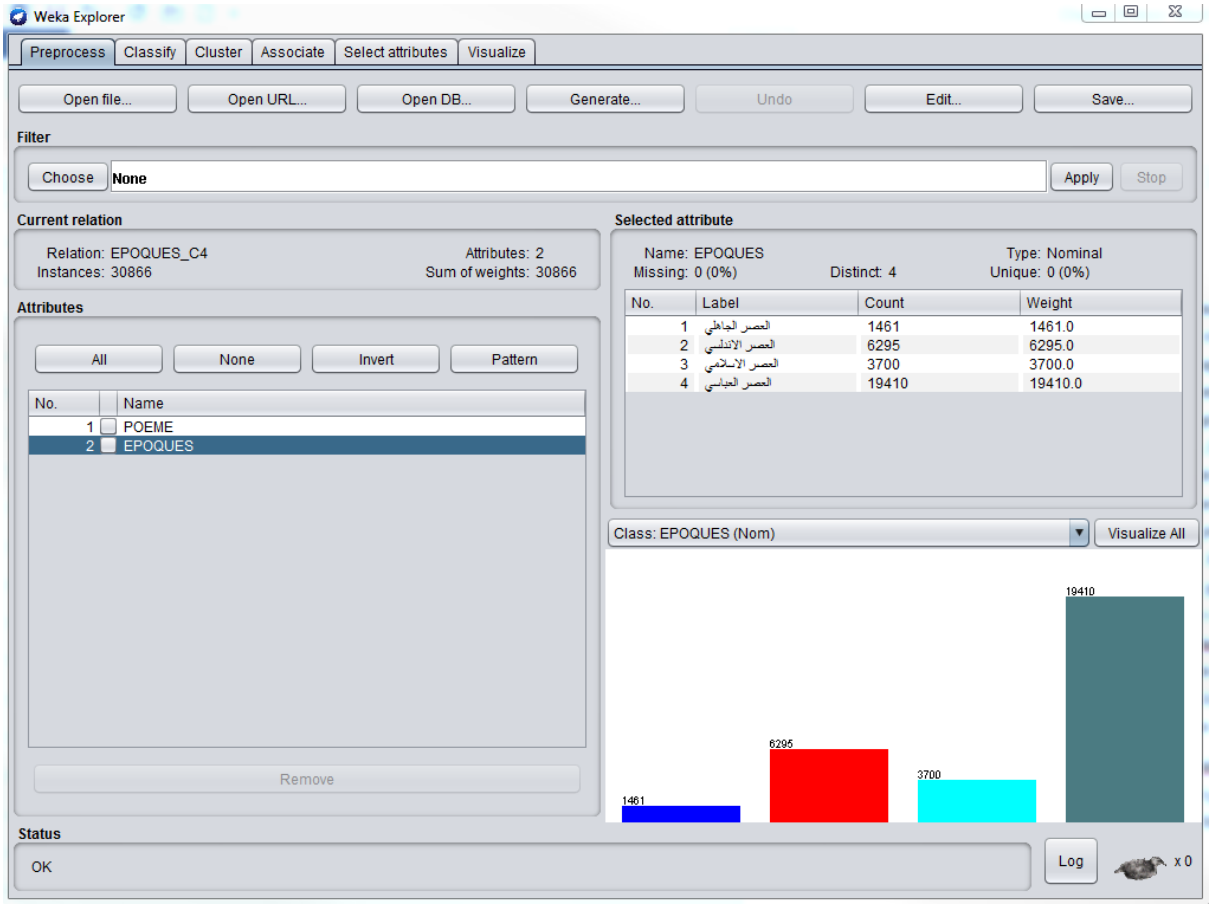


Figure 4.3 Interface de prétraitement de WEKA

IV.5–Expérimentations :

Comme cité dans la section de la présentation de l'outil WEKA, les classificateurs sont regroupés par famille nous avons essayé d'appliquer au moins un classificateur de chaque famille, voire deux classificateurs, sur notre corpus sans faire aucun prétraitement et cela de deux manières :

- ✓ ***Soit, en utilisant tous les poèmes du corpus comme données d'apprentissage*** : le tableau 2 montre la comparaison entre ces différents classificateurs selon différentes mesures.

Nous avons utilisé ce mode de classification afin de démontrer la grande marge de poèmes non-correctement classifiés sur lesquels on peut refaire des prédictions en réutilisant le même modèle de classification. Ce qui fera l'objet de l'interface développée.

	J48	IBK	Naïve Bayes	SVM	Naïve Bayes MultiNomial	Multi Class Classifier	RANDOM FOREST	Bagging
Précision Moyenne	85.50%	99.70%	57.70%	78.90%	70.20%	76.40%	99.90%	80.20%
Rappel Moyen	85.40%	99.70%	51.10%	77.30%	71.60%	76.10%	99.90%	74.70%
F-Mesure Moyen	84.60%	99.70%	53.20%	74.60%	69.90%	74.10%	99.90%	70.50%
Exactitude	85.36%	**99.67%**	51.09%	**77.25%**	71.56%	76.13%	**99.91%**	64.93%
KAPPA	0.71	**0.99**	0.19	**0.51**	0.43	0.50	**1.00**	0.42

Tableau 4.2. Résultats de classification -Toutes données utilisées pour apprentissage.

- ✓ ***Soit en répartissant le corpus en deux partis de données : 66% pour l'apprentissage et 34% pour le test***. Le tableau 3 décrit la comparaison entre ces classificateurs.

	J48	IBK	NB	SVM	NBM	MCC	RF	Bag
Précision Moyenne	55.60%	47.60%	57.10%	66.40%	68.40%	66.00%	64.80%	63.20%
Rappel Moyen	59.30%	48.40%	51.20%	68.50%	70.00%	67.50%	63.80%	64.90%
F-Mesure Moyenne	56.70%	47.60%	53.20%	65.70%	68.40%	65.70%	51.30%	55.60%
Exactitude	59.34%	48.44%	51.20%	**68.70%**	**70.21%**	**68.07%**	63.84%	64.93%
KAPPA	0.17	0.04	0.19	**0.34**	**0.40**	**0.35**	0.05	0.12

Tableau 4.3. Résultats de classification avec répartition (66 -34%).
Les 03 meilleurs résultats sont gras selon Exactitude et KAPP

Nous avons choisi les trois (03) meilleurs classificateurs (ceux dont les résultats en rouge) selon les critères d'exactitude et de KAPPA respectivement NBM, SVM et MCC et qui feront l'objet des autres tests (configurations expérimentales) suivants :

1. WordTokenizer with Stop words (WrdTok StpWord)
2. WordTokenizer without Stop words (WrdTok Wt StpWord)
3. NGRAMTokenizer with Stop words (NgramTok StpWord)
4. NGRAMTokenizer without Stop words (NgramTok Wt StpWord)

Au préalable, nous appliquerons à notre corpus les quatre configurations citées ci-dessus. Puis nous expérimenterons chacun des 03 meilleurs classificateurs choisis (SVM, NBM et MCC) sur le corpus ainsi prétraité.

Chaque expérimentation commence par créer un modèle en premier lieu. Par la suite ce modèle est testé et enfin évalué en utilisant différents paramètres et mesures de performance telle que l'exactitude, la précision, le Rappel, F-mesure, Kappa ainsi que la Matrice de Confusion.
Pour chaque classificateur, les résultats comparatifs de ces expérimentations sont illustrés par un graphique ci-après :

- ***SVM WITH STOP WORDS WITH Word Tokenizer*** **:**

Instances: 30866

Test mode: split 66.0% train, remainder test

Time taken to build model: 6081.73 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.59 seconds

=== Summary ===

Correctly Classified Instances	7191	68.5249 %
Incorrectly Classified Instances	3303	31.4751 %
Kappa statistic	0.339	
Total Number of Instances	10494	

=== Detailed Accuracy by Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Area Class
0.282	0.026	0.341	0.282	0.309	العصر الجاهلي

	0.305	0.054	0.586	0.305	0.401	العصر الاندلسي
	0.389	0.034	0.616	0.389	0.477	العصر الأموي
	0.894	0.586	0.722	0.894	0.799	العصر العباسي
Avg.	0.685	0.385	0.664	0.685	0.657	

=== Confusion Matrix ===

a	b	c	d	<-- classified as
137	15	101	232	a = العصر الجاهلي
17	645	27	1428	b = العصر الاندلسي
143	26	497	613	c = العصر الأموي
105	414	182	5912	d = العصر العباسي

- ***SVM WITHOUT STOP WORDS WITH Word Tokenizer:***

Instances: 30866

Test mode: split 66.0% train, remainder test

Time taken to build model: 9578.7 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 1.76 seconds

=== Summary ===

Correctly Classified Instances	7209	68.6964 %
Incorrectly Classified Instances	3285	31.3036 %
Kappa statistic	0.3444	
Total Number of Instances	10494	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
	0.297	0.027	0.350	0.297	0.321	العصر الجاهلي
	0.311	0.050	0.611	0.311	0.412	العصر الاندلسي
	0.389	0.037	0.592	0.389	0.470	العصر الأموي
	0.894	0.581	0.724	0.894	0.800	العصر العباسي
Avg.	0.687	0.382	0.668	0.687	0.659	

=== Confusion Matrix ===

a	b	c	d	<-- classified as
144	10	105	226	a = العصر الجاهلي
13	658	31	1415	b = العصر الاندلسي
143	24	498	614	c = العصر الأموي
112	385	207	5909	d = العصر العباسي

- ***SVM WITHOUT STOP WORDS WITH NGRAM:***

Instances: 30866

Test mode: split 66.0% train, remainder test

Time taken to build model: 0.08 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.68 seconds

=== Summary ===

Correctly Classified Instances	7238	68.3533 %
Incorrectly Classified Instances	3256	31.6467%
Kappa statistic	0.3372	

Total Number of Instances 10494

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.313	0.028	0.349	0.313	0.330	العصر الجاهلي
0.299	0.051	0.595	0.299	0.398	العصر الاندلسي
0.387	0.036	0.599	0.387	0.470	العصر الأموي
0.891	0.586	0.721	0.891	0.797	العصر العباسي
Avg. 0.684	0.386	0.664	0.684	0.655	

=== Confusion Matrix ===

a	b	c	d	<-- classified as
152	7	102	224	a = العصر الجاهلي

10	633	29	1445	|	b = العصر الاندلسي
147	30	495	607	|	c = العصر الأموي
126	394	200	5893	|	d = العصر العباسي

- ***SVM WITH STOP WORDS WITH NGRAM:***

Instances: 30866

Test mode: split 66.0% train, remainder test

Time taken to build model: 6798.8 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.56 seconds

=== Summary ===

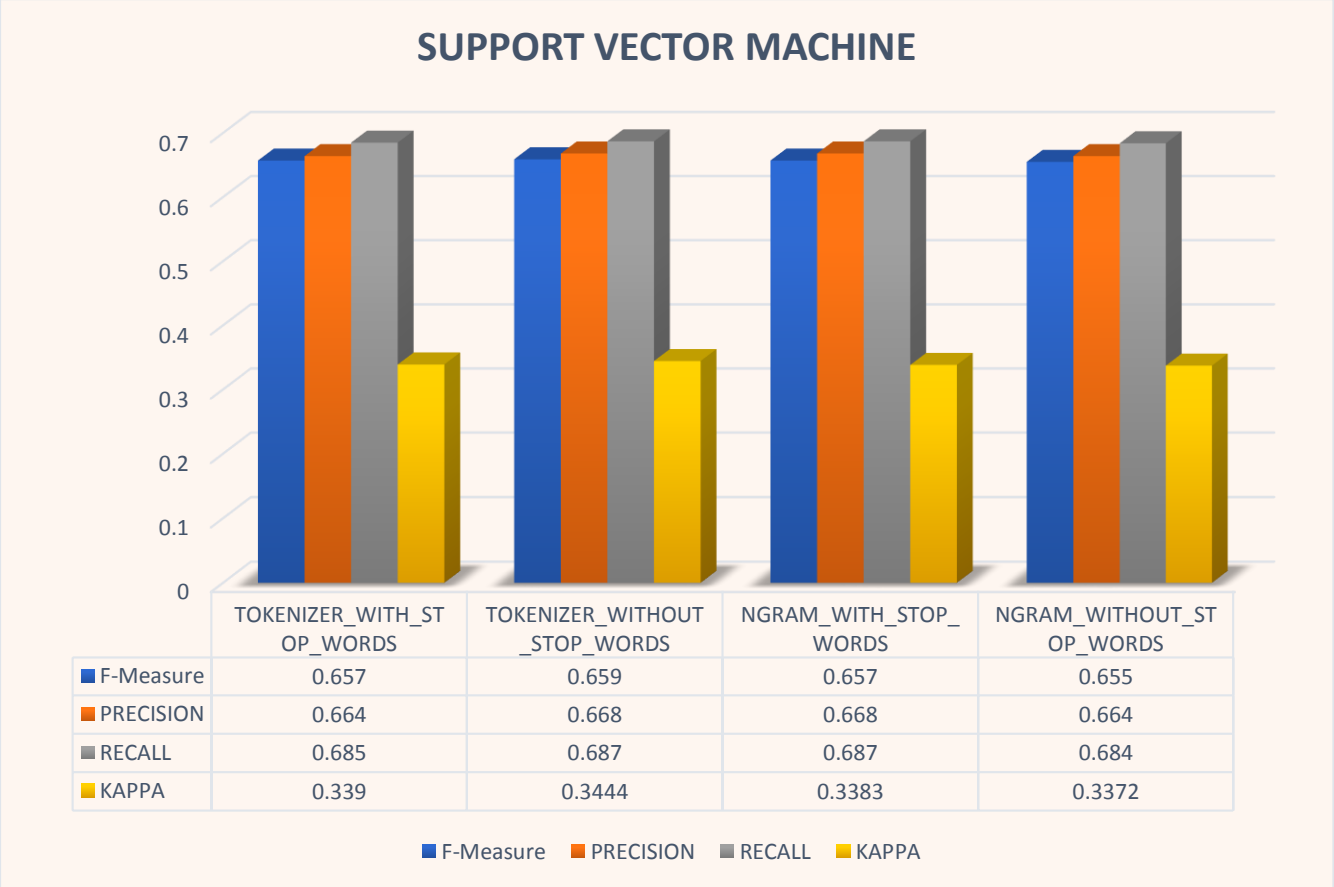
Correctly Classified Instances	7206	68.6678 %
Incorrectly Classified Instances	3288	31.3322 %
Kappa statistic	0.3383	
Total Number of Instances	10494	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.295	0.029	0.332	0.295	0.312	العصر الجاهلي
0.299	0.048	0.612	0.299	0.401	الاندلسي العصر
0.372	0.033	0.613	0.372	0.463	العصر الأموي
0.900	0.593	0.721	0.900	0.801	العباسي العصر
Avg. 0.687	0.388	0.668	0.687	0.657	

=== Confusion Matrix ===

a	b	c	d	<-- classified as
143	13	99	230	a = الجاهلي العصر
22	632	28	1435	b = الاندلسي العصر
146	22	476	635	c = الأموي العصر
120	365	173	5955	d = العباسي العصر



	TOKENIZER_WITH_STOP_WORDS	TOKENIZER_WITHOUT_STOP_WORDS	NGRAM_WITH_STOP_WORDS	NGRAM_WITHOUT_STOP_WORDS
F-Measure	0.657	0.659	0.657	0.655
PRECISION	0.664	0.668	0.668	0.664
RECALL	0.685	0.687	0.687	0.684
KAPPA	0.339	0.3444	0.3383	0.3372

Figure 4.3. Schéma de comparaison du classificateur SVM

- ***NAIVEBAYESMULTINOMIAL WITH STOP WORDS WITH Word Tokenizer:***

Instances: 30866

Test mode: split 66.0% train, remainder test

Time taken to build model: 0.05 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.59 seconds

=== Summary ===

Correctly Classified Instances	7350	70.04 %
Incorrectly Classified Instances	3144	29.96 %
Kappa statistic	0.3972	
Total Number of Instances	10494	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
	0.342	0.017	0.494	0.342	0.404	العصر الجاهلي
	0.392	0.067	0.595	0.392	0.472	العصر الاندلسي
	0.494	0.047	0.593	0.494	0.539	العصر الأموي
	0.865	0.509	0.743	0.865	0.800	العصر العباسي
Avg.	0.700	0.341	0.684	0.700	0.684	

=== Confusion Matrix ===

a	b	c	d	<-- classified as
166	5	121	193	a = العصر الجاهلي
10	829	29	1249	b = العصر الاندلسي

```
   98     14     632     535   |   c = العصر الأموي
   62     545    283     5723  |   d = العصر العباسي
```

- ***NAIVEBAYESMULTINOMIAL WITHOUT STOP WORDS WITH Word Tokenizer:***

Instances: 30866

Test mode: split 66.0% train, remainder test

Time taken to build model: 0.04 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.41 seconds

=== Summary ===

Correctly Classified Instances	7368	70.2115 %
Incorrectly Classified Instances	3126	29.7885 %
Kappa statistic	0.3985	
Total Number of Instances	10494	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.348	0.018	0.484	0.348	0.405	العصر الجاهلي
0.378	0.062	0.605	0.378	0.465	العصر الاندلسي
0.499	0.047	0.595	0.499	0.543	العصر الأموي
0.871	0.512	0.743	0.871	0.802	العصر العباسي

Avg.	0.702	0.342	0.685	0.702	0.684

=== Confusion Matrix ===

a	b	c	d	<-- classified as
169	4	117	195	| a = العصر الجاهلي
11	800	36	1270	| b = العصر الاندلسي
105	12	638	524	| c = العصر الأموي
64	506	282	5761	| d = العصر العباسي

- ***NAIVEBAYESMULTINOMIAL WITH STOP WORDS WITH Word NGRAM:***

Instances: 30866

Test mode: split 66.0% train, remainder test

Time taken to build model: 0.05 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.28 seconds

=== Summary ===

Correctly Classified Instances	7340	69.9447 %
Incorrectly Classified Instances	3154	30.0553 %
Kappa statistic	0.3996	
Total Number of Instances	10494	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.371	0.018	0.504	0.371	0.428	العصر الجاهلي
0.396	0.068	0.595	0.396	0.475	الاندلسي العصر
0.497	0.051	0.577	0.497	0.534	العصر الأموي
0.860	0.500	0.746	0.860	0.799	العصر العباسي

Avg.	0.699	0.336	0.683	0.699	0.684

=== Confusion Matrix ===

a	b	c	d	<-- classified as
180	6	114	185	| a = العصر الجاهلي
11	838	41	1227	| b = العصر الاندلسي
96	19	636	528	| c = العصر الأموي
70	546	311	5686	| d = العصر العباسي

- ***NAIVEBAYESMULTINOMIAL WITHOUT STOP WORDS WITH Word NGRAM:***

Instances: 30866

Test mode: split 66.0% train, remainder test

Time taken to build model: 0.08 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.68 seconds

=== Summary ===

Correctly Classified Instances	7311	69.6684 %
Incorrectly Classified Instances	3183	30.3316 %
Kappa statistic	0.3916	
Total Number of Instances	10494	

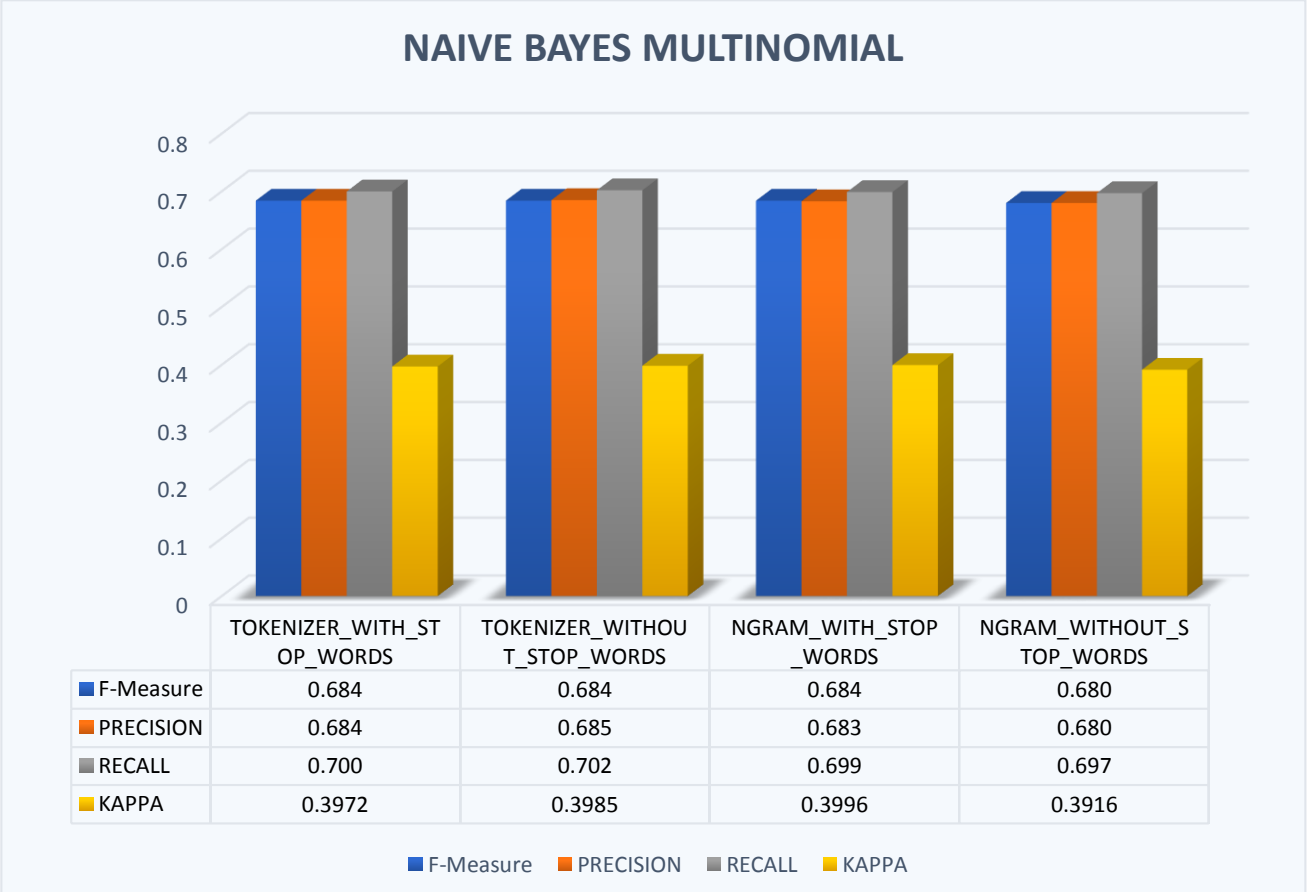
=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.386	0.019	0.493	0.386	0.433	العصر الجاهلي
0.373	0.068	0.582	0.373	0.455	العصر الاندلسي

العصر الأموي	0.496	0.048	0.590	0.496	0.539	
العصر العباسي	0.862	0.510	0.742	0.862	0.797	
Avg.	0.697	0.342	0.680	0.697	0.680	

=== Confusion Matrix ===

a	b	c	d	<-- classified as
187	5	108	185	a = العصر الجاهلي
12	790	37	1278	b = العصر الاندلسي
111	15	635	518	c = العصر الأموي
69	548	297	5699	d = العصر العباسي



	TOKENIZER_WITH_STOP_WORDS	TOKENIZER_WITHOUT_STOP_WORDS	NGRAM_WITH_STOP_WORDS	NGRAM_WITHOUT_STOP_WORDS
F-Measure	0.684	0.684	0.684	0.680
PRECISION	0.684	0.685	0.683	0.680
RECALL	0.700	0.702	0.699	0.697
KAPPA	0.3972	0.3985	0.3996	0.3916

Figure 4.4. Schéma de comparaison du classificateur NBM

- ***MULTICLASSCLASSIF WITH STOP WORDS WITH Word Tokenizer:***

Instances: 30866

Test mode: split 66.0% train, remainder test

Time taken to build model: 2593.58 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 1.33 seconds

=== Summary ===

Correctly Classified Instances	7087	67.5338 %
Incorrectly Classified Instances	3407	32.4662 %
Kappa statistic	0.3428	
Total Number of Instances	10494	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
	0.315	0.038	0.285	0.315	0.299	الجاهلي العصر
	0.357	0.070	0.563	0.357	0.437	الاندلسي العصر
	0.387	0.035	0.607	0.387	0.473	الأموي العصر
	0.859	0.545	0.729	0.859	0.789	العباسي العصر
Avg.	0.675	0.364	0.660	0.675	0.657	

```
=== Confusion Matrix ===

   a    b    c    d   <-- classified as
 153   14   83  235 |  a = العصر الجاهلي
  45  756   42 1274 |  b = العصر الاندلسي
 142   34  495  608 |  c = العصر الأموي
 197  538  195 5683 |  d = العصر العباسي
```

- ***MULTICLASSCLASSIF WITHOUT STOP WORDS WITH Word Tokenizer:***

```
Instances:    30866
Test mode:    split 66.0% train, remainder test
Time taken to build model: 4935.94 seconds
=== Evaluation on test split ===
Time taken to test model on test split: 2.36 seconds
=== Summary ===
Correctly Classified Instances        7143               68.0675 %
Incorrectly Classified Instances      3351               31.9325 %
Kappa statistic                          0.3496
Total Number of Instances            10494
=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   Class
                 0.311    0.036      0.296     0.311     0.304    العصر الجاهلي
                 0.356    0.066      0.575     0.356     0.440    العصر الاندلسي
```

الأموي	0.473	0.386	0.608	0.035	0.386
العصر					
العباسي	0.794	0.869	0.731	0.546	0.869
العصر					
Avg.	0.660	0.681	0.664	0.363	0.681

=== Confusion Matrix ===

a	b	c	d	<-- classified as
151	12	89	233	a = العصر الجاهلي
37	753	48	1279	b = العصر الاندلسي
136	43	494	606	c = العصر الأموي
186	501	181	5745	d = العصر العباسي

- ***MULTICLASSCLASSIF WITH STOP WORDS WITH NGRAM:***

Instances: 30866

Test mode: split 66.0% train, remainder test

Time taken to build model: 6949.4 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 1.15 seconds

=== Summary ===

Correctly Classified Instances	7141	68.0484 %
Incorrectly Classified Instances	3353	31.9516 %
Kappa statistic	0.3463	
Total Number of Instances	10494	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.334	0.036	0.312	0.334	0.323	العصر الجاهلي
0.352	0.065	0.579	0.352	0.438	العصر الاندلسي
0.376	0.032	0.621	0.376	0.468	العصر الأموي
0.870	0.557	0.727	0.870	0.792	العصر العباسي
Avg. 0.680	0.369	0.665	0.680	0.659	

=== Confusion Matrix ===

a	b	c	d	<-- classified as
162	12	68	243	a = العصر الجاهلي
43	745	40	1289	b = العصر الاندلسي
138	31	481	629	c = العصر الأموي
176	498	186	5753	d = العصر العباسي

- ***MULTICLASSCLASSIF WITHOUT STOP WORDS WITH NGRAM:***

Instances: 30866

Test mode: split 66.0% train, remainder test

Time taken to build model: 5202.36 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 2 seconds

=== Summary ===

Correctly Classified Instances	7105	67.7054 %
Incorrectly Classified Instances	3389	32.2946 %
Kappa statistic	0.3423	
Total Number of Instances	10494	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.322	0.037	0.295	0.322	0.308	العصر الجاهلي
0.344	0.065	0.571	0.344	0.430	العصر الاندلسي
0.384	0.035	0.600	0.384	0.468	العصر الأموي
0.866	0.552	0.728	0.866	0.791	العصر العباسي
Avg. 0.677	0.367	0.661	0.677	0.657	

=== Confusion Matrix ===

a	b	c	d	<-- classified as

156	8	85	236	|	a = العصر الجاهلي
49	729	46	1293	|	b = العصر الاندلسي
138	38	491	612	|	c = الأموي العصر
186	502	196	5729	|	d = العصر العباسي

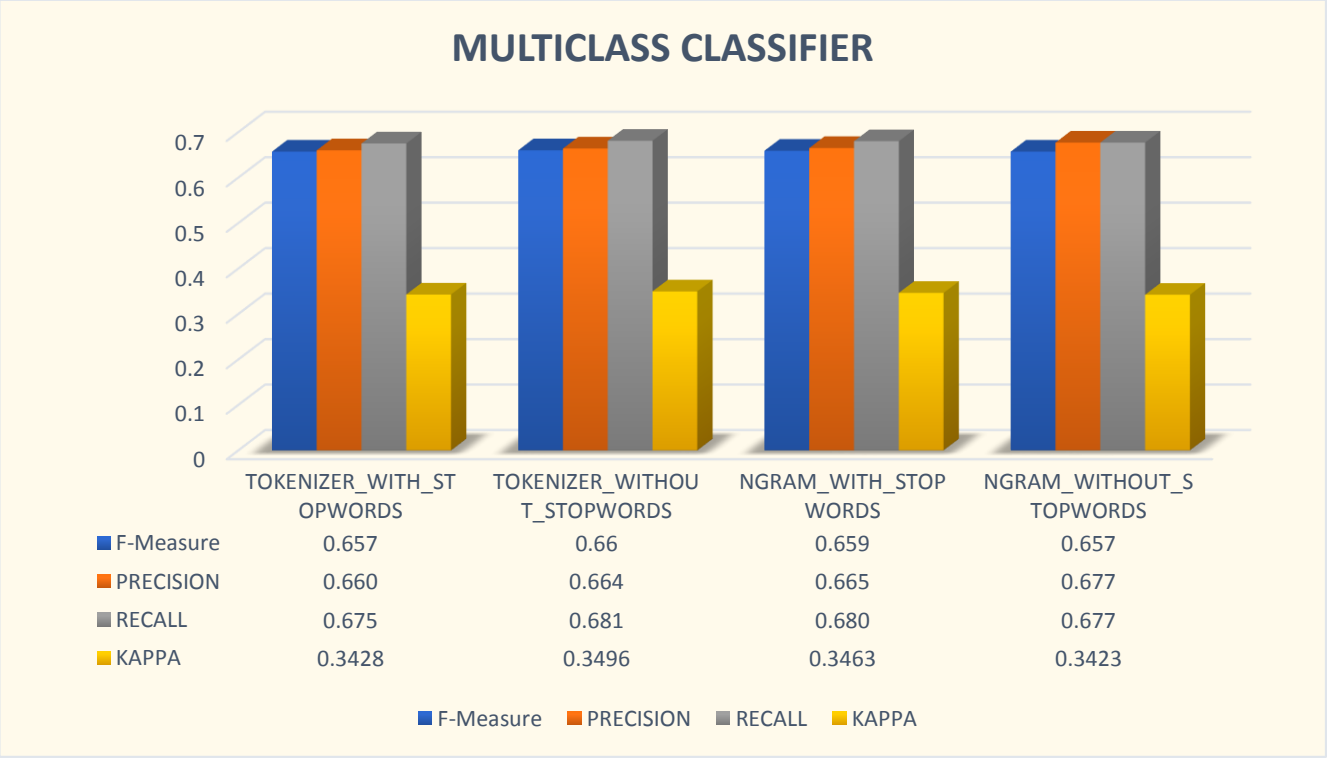


Figure 4.5 Schéma de comparaison du classificateur MCC

Déduction : Pour tous les Algorithmes et a partir des schémas de comparaison, on peut clairement constater que le fait de retirer les mots vides en utilisant la segmentation "Word_tokenizer" dégrade les résultats, par contre avec la segmentation de type NGRAM, extraire ces mots vides peut nettement augmenter les performances en termes de précision, de rappel et de score F1

. Ceci étant, il demeure que "Word tokenizer" en maintenant les mots vides retourne de meilleurs résultats que les Ngram sans ces mots vides.

IV.6–Résultats :

Comme mentionné ci-dessus, les expériences ont été menées à l'aide de WEKA (Waikato Environnent pour l'acquisition de connaissances). Nous avons mené un processus de classification utilisant un ensemble de classificateurs, répartissant 66% de l'ensemble de données du corpus pour la l'apprentissage et 34% pour le test.

Notre démarche a consisté à faire dans un premier lieu des comparaisons entre les différentes configurations pour chaque classificateur en termes de précision, rappel et score F1.

En second lieu, nous comparons les résultats de ces mêmes classificateurs en termes d'exactitude (Figure-4.6). On constate alors que les résultats du classificateur Naive Bayes Multinomial sont nettement meilleurs avec un score de (70,21%), suivent alors dans l'ordre SVM puis MCC.

De ce fait, notre choix se portera sur le modèle généré par le classificateur NBM, et ce lors de la phase d'implémentation pour en établir les prédictions.

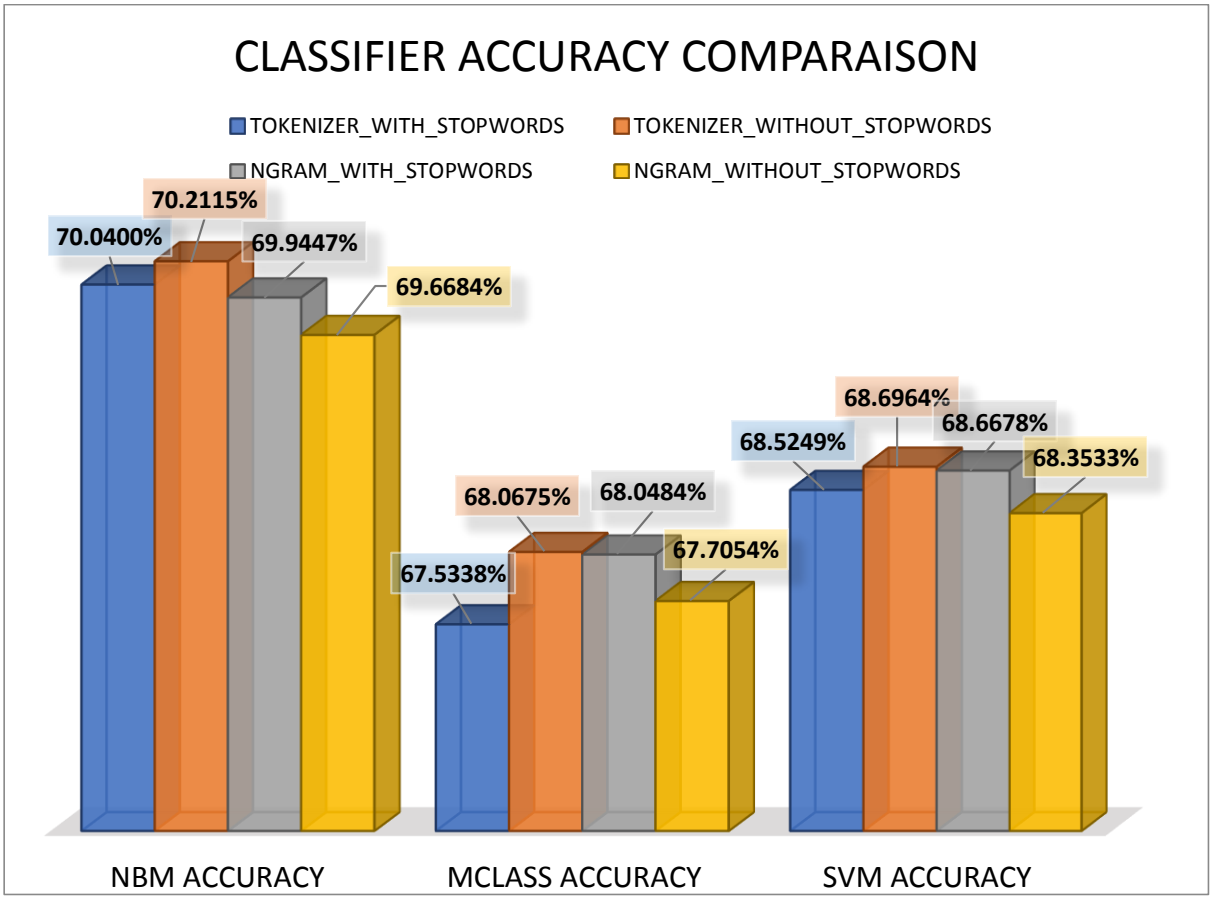


Figure 4.6 Schéma de comparaison des classificateurs NBM, MCC et SVM

IV.7–Interfaces Graphiques de l'application :

Nous avons mis au point une application (développée sous Python) de prédiction de classification dont nous présentons un aperçu de l'interface utilisateur graphique (GUI).

Cette application utilise le modèle WEKA généré par le classificateur ayant fourni le meilleur score (NBM).

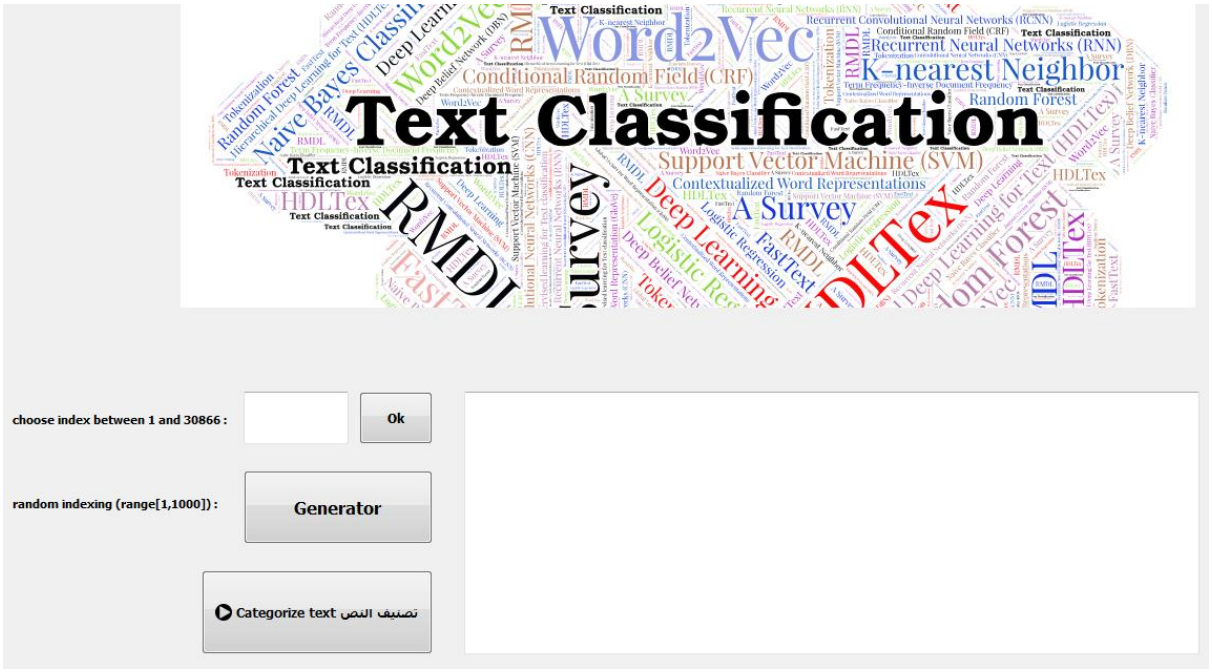


Figure 4.7 Interface Principale de l'application

On Choisit l'index du fichier ARFF terme ou texte à classifier.

Comme le montre l'image ci-dessus, on peut procéder de deux (02) manières :

- ✓ En introduisant l'index directement
- ✓ En le générant d'une manière aléatoire (clic sur le bouton Génération)

Une fois le fichier sélectionné, le poème indexé sera affiché ainsi que sa classe associée dans la fenêtre.

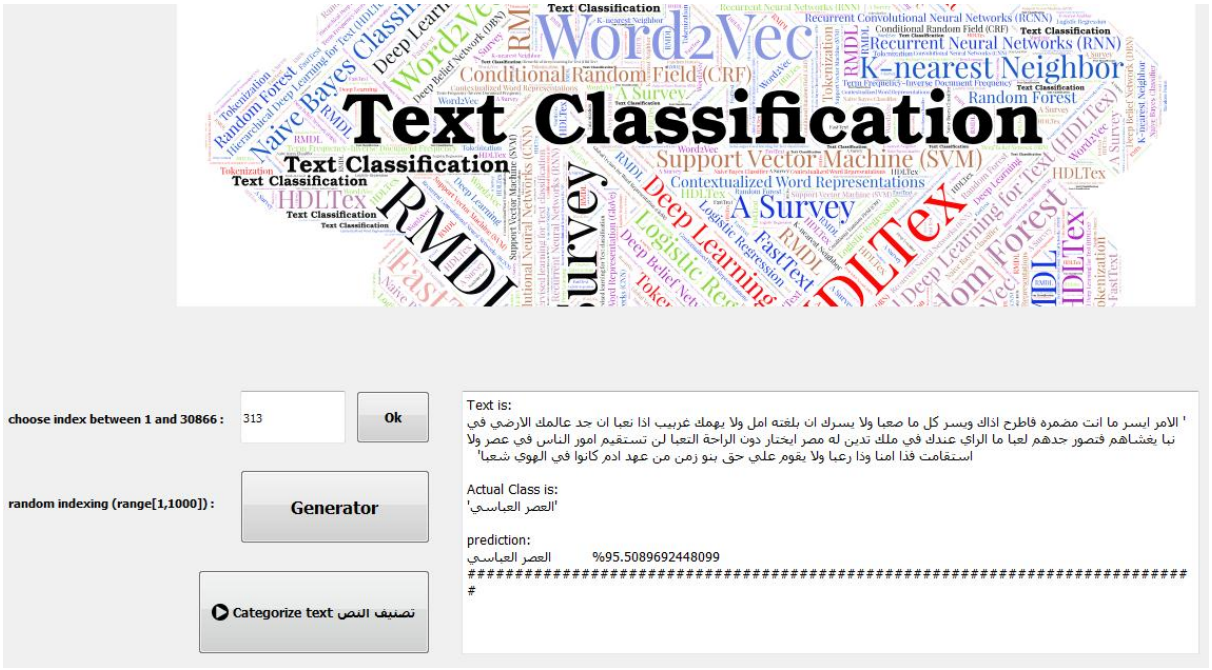


Figure4.8 Prédiction avec insertion manuelle de l'index

Figure 4.9 Prédiction avec insertion aléatoire de l'index

En cliquant sur : "Categorize text", l'application traite le poème et prédit sa classe avec un pourcentage de similarité.

Dans l'exemple illustré, on peut voir que la prédiction est exacte avec un pourcentage de près de 96%

IV.8-Conclusion :

Cet ouvrage aborde le problème de la catégorisation de la poésie dans la langue arabe.

Nous avons commencé par sélectionner quatre des ères (catégories) les plus connues de notre corpus de poésie arabe.

De plus, nous avons utilisé quatre différentes méthodes prétraitements, à savoir WORD TOKENISER avec et sans "STOP-WORDS" et "NGRAM TOKENISER". Avec et sans "STOP-WORDS".

Nous avons mené plusieurs expériences sur cet ensemble de données via WEKA puis nous avons choisi les trois (03) meilleurs classificateurs ayant restitué les résultats les plus probants que nous commentons comme suit :

Nous avons constaté qu'avec WORD TOKENIZER et avec STOP-WORDS, nous obtenons de meilleurs résultats qu'avec WORD TOKENIZER sans STOP-WORDS.

Au contraire, Ngram Tokenizer sans STOP WORDS a donné de meilleurs résultats que Ngram Tokenizer avec STOPWORDS.

La plus grande exactitude ("Accuracy") est obtenue avec le classificateur Bayésien multinomial avec WORD TOKENIZER et sans STOP WORDS dont le score est de 70,21%.

Ces expériences, attestent que la suppression des STOP WORDS, souvent d'usage, lors de l'étape de prétraitement n'est pas toujours recommandable. En revanche, maintenir ces STOP-WORDS peut mener dans certains cas de catégorisation à des résultats concluants.

Notre recommandation consiste à réaliser, des tests avec et sans STOP-WORD, ce afin de mieux en évaluer l'impact sur le modèle et sur la problématique posée. S'en devrait suivre un choix judicieux de son utilisation ou pas dans la phase de prétraitement.

Bibliographie

[1] Utilisation de WordNet dans la catégorisation de textes multilingues Published, 2007.

[2] Emilie DUMONT SIMILARITE DES SEQUENCES VIDEO : APPLICATION AUX RUSHES Université de Nice-Sophia Antipolis

[3] Radwan JALAM Apprentissage automatique et catégorisation de textes multilingues UNIVERSITÉ LUMIÈRE LYON2 Année 2003

[4] Radwan JALAM le 4 juin 2003 Apprentissage automatique et catégorisation de textes multilingues

[5] BENTAALLAH Mohamed Amine Utilisation des Ontologies dans la Catégorisation de Textes Multilingues Université Djillali Liabes de Sidi Bel Abbes

[6] https://fr.wikipedia.org/wiki/Kappa_de_Cohen

[7] Amélioration du produit scalaire via les mesures de similarités sémantiques dans le cadre de la catégorisation des textes Université Abou Bakr Belkaid– Tlemcen

[8] Mohamed Zakaria Kurdi , Traitement automatique des langues et linguistique informatique 2, Volume 2

[9] CHERAGUI Mohamed AmineConception et Réalisation d'un lemmatiseur hybride de texte arabe Université Ahmed Draya Adrar Algérie

[10] Bilel Bahloul, méthode pour l'analyse automatique d'opinion de la langue arabe

[11] https://fr.wikipedia.org/wiki/Arabe

[12] Fouad Soufiane Douzidia Résumé automatique de texte arabe Université de Montréal

[13] Mustafa et al., 2008 M. Mustafa, H. AbdAlla, and H. Suleman, .Current Approaches in Arabic IR: A Survey. In Proceedings The Annual International Conference on Asia-Pacific Digital Libraries (ICADL), Bali, Indonesia. 2008

[14] Khoja et al. 2001 S. Khoja, R. Garside, and G. Knowles. A Tagset for the Morphosyntactic Tagging of Arabic". Proceedings of the Corpus Linguistics. Lancaster University (UK), 2001.

[15] Cours du Dr. Mounir ZRIGUI Traitement automatique de la langue unité de recherche RIADI,faculté des Sciences de Monastir Tunisie

[16] BENHALIMA MAISSA Implémentation d'une méthode hybride (Morphologique & statistique) pour l'analyse des mots arabes UNIVERSITE MOHAMED BOUDIAF - M'SILA

[17] Taani, 2009 A. Taani. A rule-based approach for tagging non-vocalized Arabic words. The International Arab Journal of Information Technology, pp. 320-328, 2009

[18] Dhifallah OTHMEN Etiquetage morphosyntaxique de l'arabe avec Nooj

[19] Belguith et al. 2005 L. Hadrich Belguith, L. Baccour et M. Ghassan. 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. Vol. 1, pages 451–456, 2005

[20] Amirouche Radia UNE COMBINAISON DE CLASSIFIEURS POUR LA RECONNAISSANCE DES VISAGES HUMAINS UNIVERSITE BADJI – MOKHTAR – ANNABA

[21] J.R. Quinlan (1986). Induction of Decision Trees, Machine Learning, (1), 81-106

[22] Quinlan, J. R. (1993), C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, California.

[23] Touina Hanane Classification automatique de textes UNIVERSITE MOHAMED BOUDIAF - M'SILA

[24] SIMON REHEL : « 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, Janvier 2005

[25] [20] Harry Zhang "The Optimality of Naive Bayes". Conférence FLAIRS 2004

[26] Caruana, R. and Niculescu-Mizil, A.: "An empirical comparison of supervised learning algorithms".Proceedings of the 23rd international conference on Machine learning, 2006.

[27] Apprentissage à partir d'exemples ; Cours Master ; janvier 2009.

[28] Implémentation et Génération d'un Ensemble Bagging dans la Plateforme Weka UNIVERSITE ABDELHAMID IBN BADIS DE MOSTAGANEM