REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DE GUELMA
FACULTE DES SCIENCES ET DE L'INGÉNIEUR

Mémoire de Magister

Présenté au département d'informatique
Pour l'obtention du diplôme de

Magister en Informatique

Option : Intelligence Artificielle et Imagerie

Par Mr. Aouine Mohammed

CATEGORISATION AUTOMATIQUE DE TEXTE ARABE

JURY

Président	**Mr**	**Hamid SERIDI**	**Pr**	**Université de Guelma**
Rapporteur	**M^me^**	**Hassina SERIDI**	**MC**	**Université d'Annaba**
Examinateur	**M^me^**	**Habiba BELLEILI**	**MC**	**Université d'Annaba**
Examinateur	**M^me^**	**Halima BAHI**	**MC**	**Université d'Annaba**

2009

Dédicace

A mes très chers parents,

A mes chers frères et sœurs,

A mes collèges de travail,

A mes meilleurs amis,

Je dédie ce mémoire,

Aouine Med

Remerciement

Mes remerciements vont à l'endroit de tous ceux qui ont contribué à la réussite de ce travail,

Notre reconnaissance va tout naturellement à **Dr Hacina SERIDI**

Leur conseils nous ont été d'un apport précieux.

Nous remercions **Pr Hamid SERIDI**, d'avoir si complaisamment accepté de participer à notre jury et d'en être le Président.

Nous sommes reconnaissant envers

Dr Halima BAHI et **Dr Habiba BELLEILI**

d'avoir bien voulu étudier ce travail et participer à la commission d'examen afin de le juger.

Qu'ils en soient louablement gratifiés.

Aouine Med

الملخص:

نظرا للكم الهائل للنصوص العربية في مواقع الانترنت، البريد الالكتروني، قواعـد بيانـات الـشركات، والمكتبات الالكترونية، أصبح من الصعب الحصول على المعلومات بصورة سريعة ودقيقة ولان تصنيف النصوص يدويا في هذه الحالة أصبح صعب ومحدود وان كان ممكنا فذلك سيؤثر حتمـا علـى جـودة وسرعة التصنيف إضافة إلى التكلفة الباهظة لذلك . من هنا صار من الضروري تصميم برامج تـصنيف آلية.

نوضح من خلال هذا البحث طريقة لتصنيف النصوص العربية باستخدام أنظمة SVM، التي تعتمد على التعليم الذاتي في مجال الذكاء الاصطناعي.

لتحقيق ذلك قمنا **أولا** بمعالجة لغوية وإحصائية للغة العربية تهدف في الأساس إلى تحـضيرها للمرحلـة الثانية وهي إنشاء نظام التصنيف عن طريق التعليم الذاتي باستعمال SVM

الكلمات المفاتيح: المعالجة الآلية للغة العربية، التصنيف الآلي، التعليم الذاتي.

Abstract:

Given the enormous increase in the volume of Arabic documents on the Internet, e-mail, the intranet and libraries, access to data in a precise and fast becomes very difficult. Manual categorization of the texts in this case is very difficult even if it can infect the efficiency, timeliness and cost. To do this it is necessary to develop automatic categorization.

This article proposes a method for categorizing Arabic texts using the Support Vector Machine (SVM) which is based on machine learning. A linguistic and statistical preprocessing is done to adapt the SVM Arabic text, then a model of categorization is generated from a learning process with SVM.

Keywords: Automatic processing of Arabic, Automatic Categorization, Machine learning.

Résumé:

Devant l'augmentation énorme du volume de documents arabe sur Internet, e-mail, l'intranet des entreprises et les bibliothèques numériques, l'accès aux données d'une façon précise et rapide devient très difficile. La catégorisation manuelle des textes dans ce cas est très difficile même s'il est possible elle infect l'efficacité, la rapidité et le coût. Pour cela il est donc nécessaire de développer des programmes de catégorisation automatique.

Cet article propose une méthode de catégorisation des textes arabes en utilisant les Support Vector Machine (SVM) qui est basée sur l'apprentissage automatique. Un prétraitement linguistique et statistique est effectué afin d'adapter les techniques SVM aux texte arabe, ensuite un model de catégorisation est générer à partir d'un processus d'apprentissage automatique avec SVM .

Mots clé: Traitement automatique de la langue arabe, Catégorisation automatique, Apprentissage automatique.

SOMMAIRE

SOMMAIRE

SOMMAIRE

LISTE DES FIGURES

LISTE DES TABLEAUX

Chapitre I

Introduction

CHAPITRE 01- Introduction

1. Introduction:

Devant l'augmentation énorme du volume d'informations numériques sur Internet ou dans l'intranet des entreprises, l'accès à ces données devient très difficile ce qui créer de nouveaux besoins pour organiser et traiter ces immenses volumes de données. Le coût élevé d'un expert de travailler continûment pour classifier les documents ainsi que leur capacité qui est limité : il ne peut pas travailler 24 heures et suivre le changement de flux de données.

On a besoin donc d'outils nous permettant de chercher, classer, conserver, mettre a jour et analyser les données accessibles. Il nous faut des outils qui nous aident à trouver dans un temps raisonnable l'information désirée, effectuant certaines tâches à notre place, ou, du moins, nous facilitant le travail.

Un de ces domaines prometteurs est La fouille de texte "text mining" qui consiste à gérer le contenu des sources volumineuses de textes stockées ", plus simplement : extraction d'information depuis de textes" dans des bases (Internet, intranet,…). La fouille de texte fait appel à divers méthodes d'analyse, comme la linguistique, la catégorisation du texte qui est l'objet d'étude de ce mémoire.

2. Apprentissage automatique:

La catégorisation automatique de texte (CAT) consiste à affecter des textes à des catégories en se basant sur leur contenu, estimé par un apprentissage automatique (machine learning méthode). Les approches actuelles de catégorisation de textes sont directement liées à l'apprentissage automatique, qui se divise principalement en deux manières d'apprendre : l'apprentissage

supervisé et l'apprentissage non supervisé. C'est dans l'approche dite supervisée que s'inscrit la façon dont on aborde aujourd'hui le problème de la catégorisation automatique de textes. Dans ce paradigme, l'apprentissage s'effectue à partir d'un ensemble d'exemples où chacun d'eux est constitué d'un objet d'entrée et d'une valeur de sortie désirée pour cet objet. En connaissant les sorties prévues, l'algorithme peut généraliser les exemples afin d'identifier les différents attributs des objets qui justifient une sortie particulière, pour être en mesure de traiter de nouvelles données. Ce processus correspond exactement à ce qui est souhaité en catégorisation de textes : un apprentissage sur une banque de documents dont la classe d'appartenance est connue du classificateur (ayant été déterminée au préalable par un humain.

3. Approche proposé:

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, etc.

Plusieurs méthodes sont utilisées pour le catégorisation de textes comme: Algorithmes naïve Bayes [1], les K plus proches voisins [2], l'entropie maximum [3], les SVM (Support Vector Machine) [4], et les modèles Markovien [5].

Mais pour la langue arabe peu de travaux sont effectuées dans la catégorisation automatique. La catégorisation de langue arabe est très différente de la catégorisation de l'anglais car l'arabe est très inflexionnelle et dérivable ce qui rend la tache d'analyse morphologique très complexe [6]

Afin de surmonter les problèmes de catégorisation de texte arabe que nous venons de présenter, une approche de catégorisation de textes arabes basée sur la classification par Machine à Vecteurs Support SVM "Support Vector Machine" est proposée.

4. Plan de mémoire:

Le mémoire est structuré comme suite:

- **Chapitre 02: Processus de catégorisation de Texte** qui détermine les différentes étapes de la catégorisation automatique de textes aussi nous présentant les principaux algorithmes d'apprentissage dans le domaine de la catégorisation de texte, en détaillant notre algorithme SVM (Support Vector Machine).
- **Chapitre 03: Langue Arabe:** une étude sur les caractéristiques et la spécificité de la langue arabe.
- **Chapitre 04: Conception du système proposé** : présente de façon détaillée l'approche que nous proposons.
- **Chapitre 05: Tests et Résultats**: Test et évaluation de note système avec d'autres travaux déjà réalisés.

Chapitre II

Catégorisation de Texte

CHAPITRE 02 – Catégorisation de Texte

1. Catégorisation de texte:

La catégorisation de texte consiste à chercher 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. 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 possible, c'est-à-dire le modèle qui produit le moins d'*erreur* en prédiction.

2. Comment catégoriser un texte:

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. Pour identifier la catégorie ou la classe à laquelle un texte est associé, un ensemble d'étapes est habituellement suivie. 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.

Il comporte deux phases que l'on peut distinguer comme suit **[7]:**

- ✓ **L'apprentissage**, qui comprend plusieurs étapes et aboutit à un modèle de prédiction:
    a. Nous disposons d'un ensemble de textes étiquetés (pour chaque texte nous connaissons sa catégorie) ;
    b. A partir de ce corpus, nous extrayons les k descripteurs (ou mots, ou termes) (t_1; ..; t_k) les plus pertinents au sens du problème à résoudre ;
    c. Nous disposons alors d'un tableau « descripteurs × individus », et pour chaque texte nous connaissons la valeur de ses descripteurs et son étiquette;
    d. Nous appliquons un algorithme d'apprentissage sur ce tableau afin d'obtenir un modèle de prédiction.

✓ **Le classement** d'un nouveau texte dx, qui comprend deux étapes :

a. Recherche puis pondération des occurrences (t_1; ...; t_k) des termes dans le texte dx à classer ;

b. Application du modèle sur ces occurrences afin de prédire l'étiquette de ce texte dx.

Le processus de catégorisation, est résumé dans la figure 2.1:

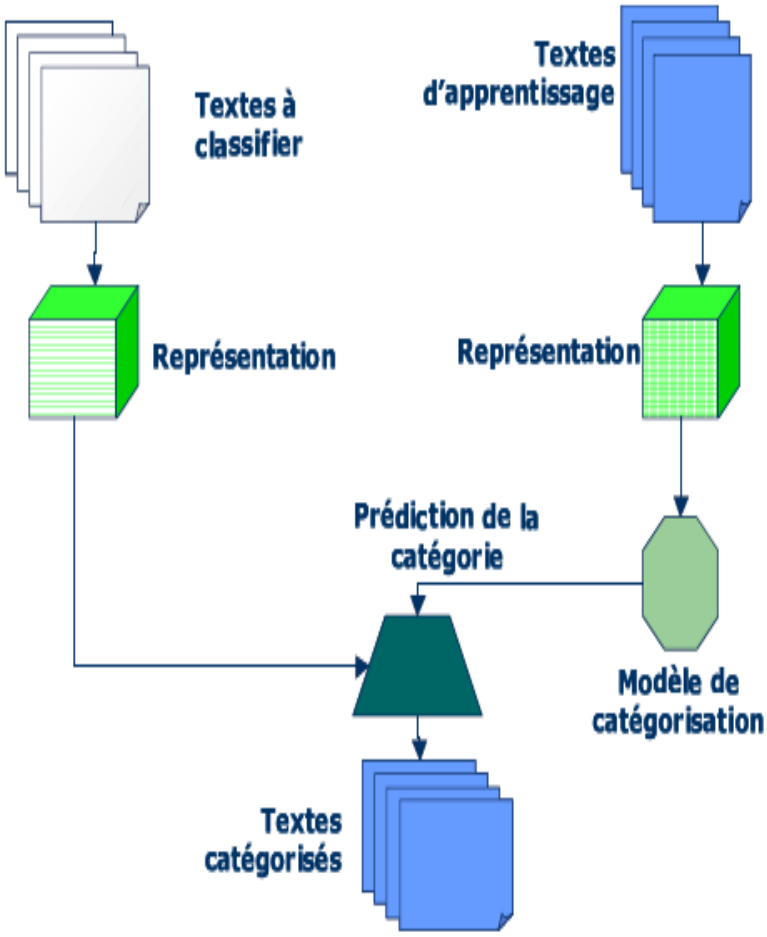


Figure 2.1 La tâche de catégorisation

Notons que les k descripteurs les plus pertinents (t_1; ...; t_k) sont extraits lors de la première phase par analyse des textes du corpus d'apprentissage. Dans la seconde phase, celle du classement d'un nouveau texte, nous cherchons simplement la fréquence de ces k descripteurs (t_1; ...; t_k) dans ce texte à classer.

2.1 Représentation des textes:

Pour pouvoir entraîner une machine sur de telles données il faut avant toute chose spécifier un format d'entrée qui soit compris par l'algorithme d'apprentissage. Pour la catégorisation automatique de textes, le format le plus utilisé est la représentation vectorielle "bag of words". On représente un texte d sous la forme suivante : $d = \langle w_1, w_2, \ldots, w_{|T|} \rangle$

Où T désigne l'ensemble des termes du lexique et w_i désigne la pondération du terme $_i$ dans le texte.

Pour calculer le poids w_i la méthode la plus utilisées est *TFIDF* (acronyme pour «term frequency inverse document frequency») [8]. Celle-ci donne plus d'importance aux mots qui apparaissent souvent à l'intérieur d'un même texte, et donne également moins de poids aux mots qui appartiennent à plusieurs textes,

Le poids d'un terme t_k dans un texte d_j est calculé ainsi :

$$tfidf\left(t_k, d_j\right) = \#\left(t_k, d_j\right).\log\frac{|Tr|}{\#\left(t_k\right)}$$

Où

- $\#\left(t_k, d_j\right)$ est le fréquence de t_k dans d_j
- $|Tr|$ est le nombre de textes d'entraînements
- $\#\left(t_k\right)$ est le nombre de textes d'entraînements dans lesquels t_k apparaît au moins une fois.

2.2 Le Prétraitement:

Avant l'encodage vectoriel, Il est nécessaire d'effectuer au préalable du codage d'un document dans un espace de mots une transformation permettant le passage de l'espace du caractère à un espace de mots. Cette transformation est appelée prétraitement.

2.2.1 La Segmentation:

La phase de segmentation consiste à découper la séquence des caractères afin de regrouper les caractères formant un même mot. Une segmentation classique consiste à découper les séquences de caractères en fonction de la présence ou l'absence de caractères de séparation (de type « espace », « tabulation » ou « retour à la ligne »).

2.2.2 Le traitement morphologique:

Le traitement morphologique consiste à effectuer un traitement au niveau de chacun des mots afin de regrouper un ensemble de mots significativement identiques.

1. Suppression de mots vides : Certains mots comme les prépositions, conjonctions ou articles, qui ne contiennent aucune information sémantique, qui ne modifie pas le sens des mots qui les accompagnent, ont pu être retirés des documents. Ce prétraitement, permettant de réduire la taille du lexique,

2. Le stemming: Consiste à regrouper sous un même identifiant des mots dont la racine est commune.

3. Lemmatisation : La lemmatisation consiste à regrouper des mots dont la signification est la même alors même que leurs racines sont différentes (par exemple, « maison » et « baraque »). C'est une tâche plus complexe que le stemming et qui repose habituellement sur l'utilisation de grandes bases de connaissances.
Exemple: le lemme d'un mot au pluriel sera sa forme au singulier, celui d'un verbe conjugué sera sa forme infinitive.

2.2.3 Le traitement syntaxique:

a. Sélection des attributs : Les critères les plus utilisés sont [9] :

- **La fréquence** *«document frequency»* : Il s'agit tout simplement d'éliminer les mots dont le nombre de documents dans lesquels ils apparaissent est en dessous d'un certain seuil. L'idée sous-jacente est que ces mots n'apportent pas d'information utile à la prédiction de la catégorie d'un texte ou qu'ils n'influencent pas la performance globale du classificateur. Il y a aussi la possibilité que ces termes soient le résultat d'erreurs, comme un mot mal orthographié. Dans ce cas, leur élimination est donc bénéfique. Opérer de cette façon pour réduire le nombre d'attributs est plutôt simple et rapide. Un bémol à l'utilisation de cette technique est que généralement, on ne l'utilise pas pour une sélection vraiment agressive, parce qu'il est admis que les termes ayant une fréquence faible/moyenne sont relativement informatifs et devraient être conservés.

- **Le gain d'information** (*«information gain»*) : On mesure en quelque sorte le pouvoir de discrimination d'un mot, le nombre de bits d'information obtenu pour la prédiction de la catégorie en sachant la présence ou l'absence d'un mot. Cette méthode est souvent mise en pratique dans les arbres de décisions, pour choisir l'attribut qui va le mieux diviser l'ensemble des instances en deux groupes homogènes.

- **L'information mutuelle** («*mutual information*») : Cette façon d'évaluer la qualité d'un mot dans la prédiction de la classe d'un document est basée sur le nombre de fois qu'un mot apparaît dans une certaine catégorie. Plus un mot va apparaître dans une catégorie, plus l'information mutuelle du mot et de la catégorie va être jugée élevée. Plus un mot va apparaître en dehors de la catégorie (et plus une catégorie va apparaître sans le mot), moins l'information mutuelle va être jugée élevée. Il faut ensuite faire une moyenne des scores du mot jumelé à chacune des catégories. La faiblesse de cette mesure est qu'elle est beaucoup trop influencée par la fréquence des mots. Pour

une même probabilité conditionnelle sachant la catégorie, un terme rare va être avantagé, car il risque moins d'apparaître en dehors de la catégorie.

- **La statistique du** χ^2 : Mesure statistique bien connue, elle s'adapte bien à la sélection d'attributs, car elle évalue le manque d'indépendance entre un mot et une classe. Elle utilise les mêmes notions de cooccurrence mot/catégorie que l'information mutuelle, mais une différence importante est qu'elle est soumise à une normalisation, qui rend plus comparable les termes entre eux. Elle perd quand même de la pertinence pour les termes peu fréquents.

- **La force du terme** *«term strength»* : Il s'agit d'une méthode plutôt différente des autres. Elle se propose d'estimer l'importance d'un terme en fonction de sa propension à apparaître dans des documents semblables. Une première étape consiste à former des paires de documents dont la similarité cosinusoïdale est supérieure à un certain seuil. La force d'un terme est ensuite calculée à l'aide de la probabilité conditionnelle qu'il apparaisse dans le deuxième document d'une paire, sachant qu'il apparaît dans le premier.

b. Tableau " individus - variables": Consiste à transformer l'ensemble des textes en un tableau croisé "individus - variables" :

- L'**individu** est un texte d_j, étiqueté lors de la phase d'apprentissage, et à classer dans la phase de prédiction.

- Les **variables** sont les descripteurs (les termes) t_k qui sont extraits des données textuelles.

- Le **contenu du tableau** (les éléments w_{kj}), au croisement du texte j et du terme k, représente le poids de ce terme k dans le texte j.

2.3 Algorithmes d'apprentissage:

L'objectif de la phase d'apprentissage est d'induire une fonction de catégorisation à partir d'un jeu d'entraînement. On appelle jeu d'entraînement un ensemble de documents dont la catégorie est connue a priori. Pour apprendre, un système doit nécessairement intégrer de la connaissance à priori sur la nature de la solution. La question est alors de savoir comment introduire cette connaissance à priori dans l'algorithme d'apprentissage.

2.3.1 Modélisation Markovienne

Un MMC est la combinaison de deux processus stochastiques, une chaîne de Markov et une densité de probabilité associée à chaque état de la chaîne. Un MMC (figure 2.2) modélise la production d'une séquence de la façon suivante [10]:

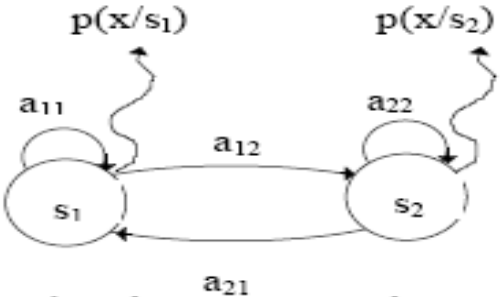


Figure 2.2 : Un MMC simple à deux états, ergodique- voir texte pour les notations.

En commençant en un état de départ, on passe d'état en état suivant la matrice de transition de la chaîne Markovienne en émettant chaque fois qu'on arrive dans un état une valeur jusqu'à ce que l'état final soit atteint. Si on note $S = \{s_1, s_2, ..., s_p\}$ l'ensemble des états et X l'espace des observations, $b_k(x) = p(x/s_k)$ sera la probabilité d'émettre x dans l'état s_k, $a_{ij} = p(s_i, s_j)$. sera la probabilité de transition de l'état s_j à l'état s_i .Pour rendre les calculs et l'estimation possible, on fait des hypothèses, les plus courantes sont : l'hypothèse Markovienne, qui dit que a_{ij} dépend uniquement des états s_i et s_j, l'hypothèse d'indépendance des sorties qui dit que $b_k(x)$ dépend uniquement

de s_k et x. Les probabilités de transition et d'émission sont apprises par un algorithme itératif du type EM. En reconnaissance, un MMC prendra en entrée une séquence et fournira en sortie la séquence d'états la plus probable ainsi que son score.

L'application de ces modèles a principalement été développée en parole. Depuis une dizaine d'année, ils sont également très utilisés pour modéliser des séquences biologiques, en TALN, ce sont les premiers modèles à avoir été développés avec succès pour l'étiquetage morphosyntaxique. Ils sont également de plus en plus utilisés pour la reconnaissance de l'écriture.

Les méthodes fondées sur les données ont ainsi besoin de quantités importantes de données d'apprentissage sous la forme de grands corpus. Par rapport aux méthodes par règles, l'intérêt des méthodes par modèles de Markov est de nécessiter moins de travail humain : la compilation des données est effectuée automatiquement. En revanche, selon la taille des données utilisées, cette compilation, accompagnée des inévitables optimisations, peut s'avérer tout aussi coûteuse en terme de temps.

Récemment plusieurs études ont proposé des implémentations de ces modèles pour l'accès à l'information textuelle. Elles considèrent en général des MMC discrets - X est un ensemble fini, on parlera alors de symboles pour désigner ses éléments. En texte, ces symboles seront tout simplement les mots du texte ou les termes issus d'un prétraitement. [11] décrivent les MMC dans le contexte de traitement du langage naturel.

2.3.2 Réseaux Bayesiens:

L'introduction de connaissance a priori sur la nature de la solution peut se faire dans le cadre des réseaux bayesiens. Dans ce cadre, il s'agit, à partir de données et/ou d'une connaissance experte de formaliser le problème en faisant apparaître des quantités reliées entre elles de façon probabiliste. Il faut trouver comment formuler la distribution à priori pour qu'à la fois, elle soit fidèle au problème abordé et qu'elle facilite les calculs. Les applications de ce type d'approche sont multiples comme la sélection des variables pertinentes ou le remplacement des données manquantes et surtout l'inférence de n'importe quelle variable à partir de certaines variables observées.

Pour prédire la classe à laquelle un document appartient [12]:

$$P\{Y = y_i \mid X = x\} = \frac{P\{X = x \mid Y = y_i\}.P\{Y = y_i\}}{\sum_i P\{X = x \mid Y = y_i\}.P\{Y = y_i\}}$$

L'idée consiste à classer un texte suivant la catégorie pour laquelle cette probabilité est la plus élevée.

$$y = \arg\max_{y_i} (P\{X = x \mid Y = y_i\}.P\{Y = y_i\})$$

Remarquons que le dénominateur du théorème de Bayes est identique pour toutes les catégories. Par conséquent, nous devons uniquement considérer le numérateur.

Pour calculer *y*, nous avons besoin des deux quantités suivantes :

1. P {Y = y_i}
2. P {X = x | Y = y_i}

On parle souvent de probabilité a priori pour la première quantité. On peut l'estimer en calculant la proportion de documents du training set appartenant à la classe y_i (méthode du maximum de vraisemblance).

$$P\{Y = y_i\} = \frac{|\{(x,y) \in Tr \mid y = y_i\}|}{|Tr|}$$

Observons à présent la seconde quantité. Lorsque l'on parle de P{X = x}, cela se réfère à la probabilité d'avoir un document tel que sa représentation vectorielle soit $hw_1,w_2, \ldots ,w_{|T|}i$. Nous supposons que les textes nous sont fournis dans le format bag of words et que les poids sont binaires (0 : aucune occurrence, 1 : au moins une occurrence).

Réexprimons la seconde quantité par la ***Chain Rule*** :

$$\begin{aligned} P\{X = x \mid Y = y_i\} &= P\{w_1, w_2, \ldots, w_{|\mathcal{T}|} \mid Y = y_i\} \\ &= P\{w_1 \mid w_2, \ldots, w_{|\mathcal{T}|}, Y = y_i\} \\ &\times P\{w_2 \mid w_3, \ldots, w_{|\mathcal{T}|}, Y = y_i\} \times \ldots \\ &\times P\{w_{|\mathcal{T}|} \mid Y = y_i\} \end{aligned} \quad \textbf{……(1)}$$

L'aspect naïf de ce genre de classificateur provient du fait que l'on considère que les évènements w_{jx} sont indépendants.

Par conséquent, la seconde quantité devient:

$$P\{X = x \mid Y = y_i\} = P\{w_1 \mid Y = y_i\} \times P\{w_2 \mid Y = y_i\} \times \ldots \times P\{w_{|T|} \mid Y = y_i\} \quad \text{.....(2)}$$

En pratique, lorsque l'on classe un nouveau document, on ne tient compte que des w_l non nuls dans l'expression de P {X = x | Y = y_i}. La raison est qu'il y a généralement nettement moins de termes dans un document que dans notre lexique.

Pour calculer P {w_l |Y=y_i}, il suffit de compter le nombre d'occurrences du terme l dans l'ensemble des documents classés sous y_i et de diviser cette quantité par la somme des occurrences de tous les termes dans cette même catégorie. On effectue aussi souvent un lissage de Laplace [13] pour éviter d'avoir des éléments nuls (qui se propageraient d'office par la Chain Rule).

Pour récapituler ceci, un entraînement avec Naïve Bayes consiste à calculer les quantités suivantes à partir du training set :

- P {Y = y_i} : Un tableau contenant |Y| entrées
- P {w_l | Y = y_i} : Une matrice contenant $|T_r| \times |Y|$ entrées

A partir de ces structures de données, nous pouvons classer de nouveaux documents en utilisant les formules (1) et (2).

2.3.3 Arbres de décision:

Les arbres de décision sont une des techniques les plus populaires du ML supervisé. Cette méthode est très facile à mettre en oeuvre, de plus, on peut facilement interpréter les règles de décision issues de l'apprentissage. Il existe plusieurs versions des AD dont les plus connues sont ID3 [14] et C4.5 [15]. Nous présentons ici la version la plus simple : ID3.

Un AD est constitué, comme tout arbre, de noeuds et de feuilles. Chaque noeud contient un test et possède autant de descendants qu'il y a de valeurs possibles pour ce test. Pour la classification, aux extrémités des branches de l'arbre, on trouve des feuilles qui indiquent le résultat de la classification, c.à.d la classe que l'on assigne. Pour classer un nouveau document, il suffit de le soumettre à la racine de l'arbre et de le laisser parcourir les branches au fil des résultats des tests jusqu'à atteindre une feuille.

Dans le cadre des AD, l'entraînement consiste à créer l'arbre à partir du training set. Il s'agit d'un processus récursif dans lequel les données d'entraînement seront utilisées pour déterminer l'attribut à tester lors de la création des noeuds.

Au départ, on dispose d'un ensemble de documents correspondant au training set complet. Sur base de cet ensemble, on va déterminer un attribut sur lequel le test de la racine de l'arbre va porter.

Il existe plusieurs manières de déterminer les tests à réaliser dans les noeuds.

L'idée commune est de créer des tests qui discriminent le plus possible les exemples d'apprentissage. En clair, cela veut dire que l'attribut sur lequel on fait le test doit séparer les exemples en deux ensembles de taille voisine. Dans ID3, on se sert d'un critère basé sur l'entropie, notion issue de la théorie de l'information. L'entropie du training set se définit comme suit :

$$Entropie(Tr) = -\sum_{y \in \mathcal{Y}} P(Y = y) \log_2 P(Y = y)$$

L'entropie permet de mesurer l'homogénéité des exemples. Si l'entropie vaut 0, tous les exemples appartiennent à la même catégorie, alors que si elle vaut 1, il y a autant d'exemples positifs que négatifs. Notons par ailleurs que dans ce contexte on définit $0 \log 0 = 0$. L'entropie est utilisée pour formuler une mesure appelée Information Gain (IG) :

$$IG(Tr, j) = Entropie(Tr) - \left(\frac{|Tr_{j,1}|}{|Tr|} * Entropie(Tr_{j,1})\right) - \left(\frac{|Tr_{j,0}|}{|Tr|} * Entropie(Tr_{j,0})\right)$$

Où $Tr_{j,k} = \{(x,y)\ 2\ Tr \mid w_j = k\}$ et $_j$ est l'index de l'attribut (le terme) pour lequel le gain d'information est calculé.

Cette mesure est d'autant plus grande que l'attribut $_j$ est discriminant. En conséquence de quoi, lors de la création des noeuds, l'index de l'attribut sur lequel portera le test sera déterminé de la façon suivante :

$$best = \arg\max_j(IG(Tr', j))$$

Où Tr' est le sous-ensemble des exemples du training set ayant satisfait aux tests menant au noeud courant.

2.3.4 K- plus proches voisins :

k-NN (k-Nearest Neighbour) est une méthode très connue dans le domaine de la catégorisation automatique. L'idée de k-NN est de représenter chaque texte dans un espace vectoriel, dont chacun des axes représente un élément textuel.

k-NN est un algorithme de catégorisation dans lequel les classes ne sont pas représentées sous forme de texte "prototype" (profil de catégorie). Chaque nouveau texte à traiter sera comparé à l'ensemble des textes du jeu d'apprentissage afin de trouver la catégorie qui lui est plus proche, soit en moyenne celle qui contient le plus de textes voisins.

L'algorithme suivant montre comment classer un nouvel exemple :

Algorithme : classification par k-PPV

Paramètre : le nombre k de voisins

Contexte : un échantillon de l textes classés en C = c1, c2, ..., cn classes

Pour chaque texte t **faire**

2: transformer le texte t en vecteur t = (x1, x2, ..., xm)

3: déterminer les k plus proches textes du texte t selon une métrique de distance

4: combiner les classes de ces k exemples en une classe c

Fin pour

Sortie : le texte t associé à la classe c.

3. Support Vecteur Machines

SVM est une méthode de classification binaire par apprentissage supervisé, elle fut introduite par Vapnik en 1995[9]. Cette méthode est donc une alternative récente pour la classification. Le succès de cette méthode est justifié par les solides bases théoriques qui la soutiennent. Elle repose sur l'existence d'un classificateur linéaire dans un espace approprié. Puisque c'est un problème de classification à deux classes, cette méthode fait appel à un jeu de données d'apprentissage pour apprendre les paramètres du modèle. Elle est basée sur l'utilisation de fonctions dites noyau (kernel) qui permettent une séparation optimale des données.

Dans la présentation des principes de fonctionnements, nous schématiserons les données par des « points » dans un plan.

3.1. Principes:

- Les données sont plongées dans un espace de grande dimension par une transformation souvent non linéaire.

- Dans cet espace transformé, les classes seront séparées par des classifieurs linéaires qui maximisent la marge (une distance entre les classes).

- Les hyperplans peuvent être déterminés au moyen d'un nombre de points limité qui seront appelés *les vecteurs supports* (ce sont ces points qui définissent la frontière).

Dans la figure 2.3, on détermine un hyperplan optimal qui sépare les deux ensembles de points. Les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan, sont appelés *vecteurs de support*.

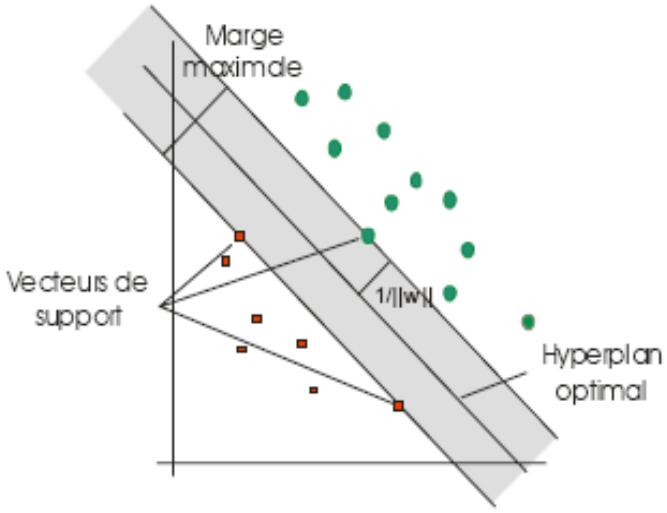


Figure 2.3 *représentation de L'hyperplan séparateur optimal qui maximise la marge dans l'espace de redescription.*

3.2 Hyperplan optimal:

Il est évident qu'il existe une multitude d'hyperplan valide mais la propriété remarquable des SVM est que cet hyperplan doit être optimal. Nous allons donc en plus chercher parmi les hyperplans valides, celui qui passe « au milieu » des points des deux classes d'exemples. Intuitivement, cela revient à chercher l'hyperplan le « plus sûr ». En effet, supposons qu'un exemple n'ait pas été décrit parfaitement, une petite variation ne modifiera pas sa classification si sa distance à l'hyperplan est grande. Formellement, cela revient à chercher un hyperplan dont la distance "marge" minimale aux exemples d'apprentissage est maximale.

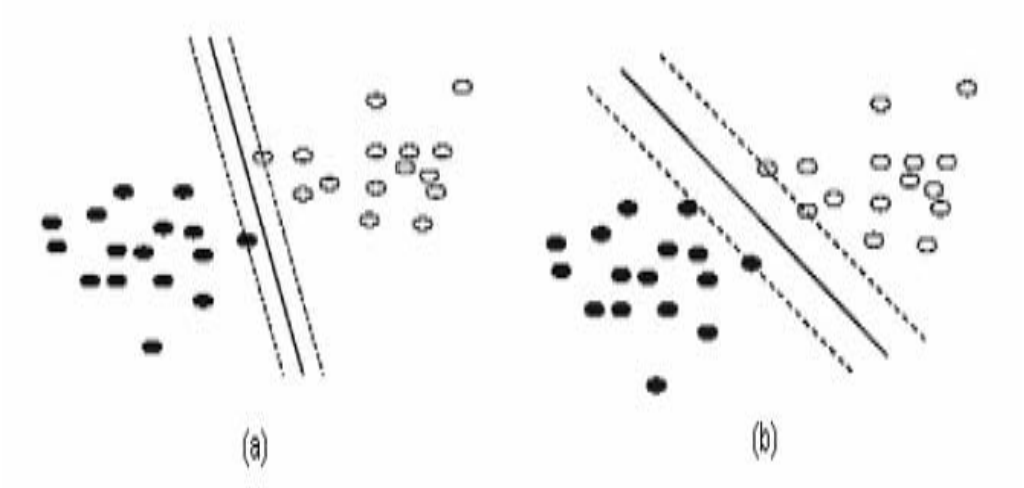


Figure 2.4 *Maximisation de la marge avec les SVM*
(a) *une frontière possible séparant les données*
(b) *la frontière entraînant une marge maximale*

Dans la figure 2.4, la partie de droite représente la frontière recherchée entre les données. Il s'agit de la ligne en trait continu. La marge, soit la distance entre cette ligne et les lignes en trait pointillé, y est maximale. La partie de gauche présente une situation où la marge serait plus petite.

Intuitivement, le fait d'avoir une marge plus large procure plus de sécurité lorsque l'on classe un nouvel exemple. De plus, si l'on trouve le classificateur qui se comporte le mieux vis-à-vis des données d'apprentissage, il est clair qu'il sera aussi celui qui permettra au mieux de classer les nouveaux exemples.

3.3 Cas non linéaire séparable:

Parmi les modèles des SVM, on constate les cas linéairement séparable et les cas non linéairement séparable. Les premiers sont les plus simples de SVM car ils permettent de trouver facilement le classificateur linéaire. Dans la plupart des problèmes réels il n'y a pas de séparation linéaire possible entre les données, le classificateur de marge maximale ne peut pas être utilisé car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables.

Pour surmonter les inconvénients des cas non linéairement séparable, l'idée des SVM *L'idée des SVM :* transformer un problème de séparation non linéaire dans l'espace de représentation en un problème de séparation linéaire dans un espace de re-description de grande dimension. Adapté de [18].

En effet, intuitivement, plus la dimension de l'espace de re-description est grande, plus la probabilité de pouvoir trouver un hyperplan séparateur entre les exemples est élevée. Ceci est illustré par le schéma suivant :

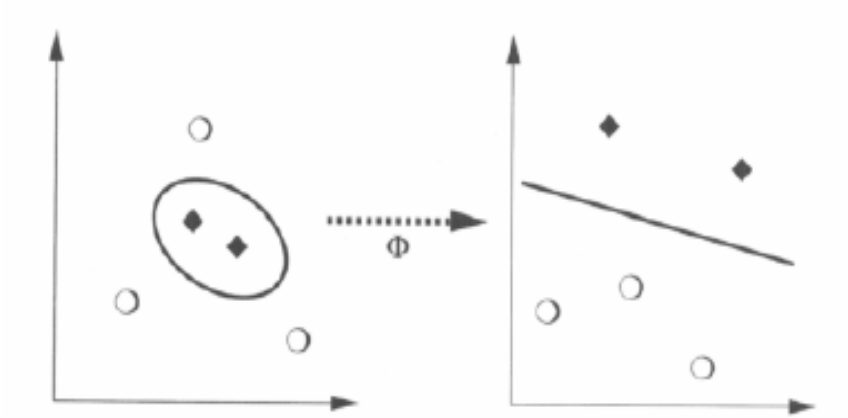


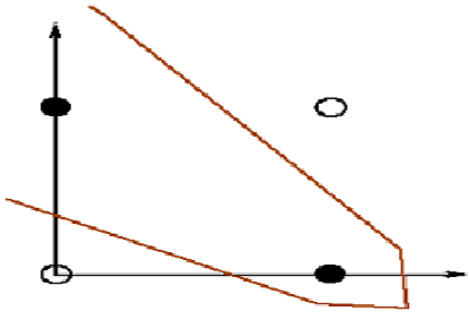
Figure 2.5 *Transformation d'un problème de séparation non linéaire en un problème de séparation linéaire*

Cette transformation non linéaire est réalisée via une fonction noyau. En pratique, quelques familles de fonctions noyau paramétrables sont connues et il revient à l'utilisateur de SVM d'effectuer des tests pour déterminer celle qui convient le mieux pour son application. On peut citer les exemples de noyaux suivants :

1. Linéaire : $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x} \cdot \boldsymbol{x}'$
2. Polynomial : $k(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x} \cdot \boldsymbol{x}')^d$ ou $(c + \boldsymbol{x} \cdot \boldsymbol{x}')^d$
3. Sigmoïde $k(x, y) = \tanh(x.y + 1)$ / *Tanh*: La fonction tangente hyperbolique
4. RBF (Radial Basis Function) $k(x, y) = e - \sigma(x\_y)^2$

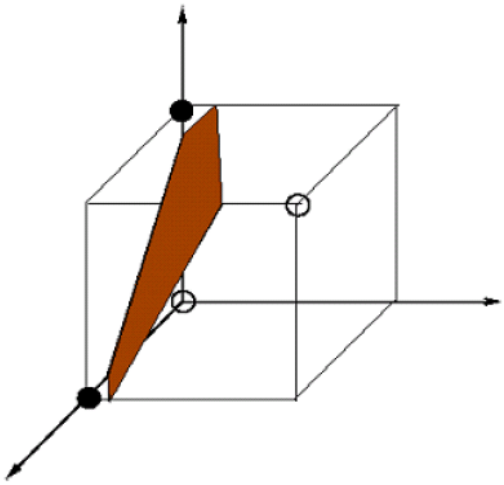
Exemple : (XOR)

Le cas de XOR n'est pas linéairement séparable, si on place les points dans un plan à deux dimension, on obtient la figure suivante Coordonnées des points : (0,0) ; (0,1) ; (1,0) ; (1,1)



Si on prend une fonction polynomiale $(x , y) \rightarrow (x , y , x.y)$ qui fait passer d'un espace de dimension 2 à un espace de dimension 3, on obtient un problème en trois dimensions linéairement séparable :

(0,0) -> (0,0,0)
(0,1) -> (0,1,0)
(1,0) -> (1,0,0)
(1,1) -> (1,1,1)



3.4. Applications des SVM :

Les SVM ont été adaptées à divers contextes : classification avec apprentissage, estimation de support de densité, régression, etc.

SVM est une méthode de classification qui montre de bonnes performances dans la résolution de problèmes variés. Cette méthode a montré son efficacité dans de nombreux domaines d'applications tels que le traitement d'image, la catégorisation de textes ou le diagnostiques médicales et ce même sur des ensembles de données de très grandes dimensions

4. Conclusion

Dans ce chapitre nous avons étudié en détail le processus de catégorisation qui comporte deux phases principales: l'apprentissage et la classification, Pour pouvoir entraîner une machine sur de telles données il faut avant toute chose spécifier un format d'entrée qui soit compris par l'algorithme d'apprentissage. Le format le plus utilisé est la représentation vectorielle "bag of words". Avant l'encodage vectoriel, certains prétraitements linguistiques doivent être réalisés: pour réduire la taille du lexique.

SVM est une méthode de classification qui montre de bonnes performances dans la résolution de problèmes variés. L'idée est de chercher les paramètres permettant d'obtenir la performance maximale. Si la mise en oeuvre d'un algorithme de SVM est en général peu coûteuse en temps, il faut cependant compter que la recherche des meilleurs paramètres peut requérir des phases de test assez longues**.**

Chapitre III

Langue Arabe

CHAPITRE 03 – Langue Arabe

1. Introduction:

Dans cette partie notre démarche consisterait à étudier les caractéristiques de la langue arabe pour ensuite effectuer un processus de transformation qui comporte deux phases: traitement linguistique afin de minimiser le plus possible l'espace de représentation tout en gardant les informations pertinentes et traitements statistiques pour calculer les poids des termes.

2. Définition :

La langue arabe est l'une de six officielles langues de nations unies, c'est une langue très différente des langues européennes, elle compte 28 consonnes et 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).

La plupart des mots arabes sont extrait à partir d'une racine de 3 caractères en ajoutant ou pénétrant des lettres ce qui engendre de nouveau mots en utilisant des schèmes. Ce qui le rend difficile à maîtriser dans le domaine du traitement automatique de la langue.

Un mot arabe s'écrit avec des consonnes et des voyelles. Les voyelles sont ajoutées au-dessus ou au-dessous des lettres. Elles sont nécessaires à la lecture et à la compréhension correcte d'un texte, elles permettent de différencier des mots ayant la même représentation.

3. Morphologie de la langue Arabe:

3.1. Généralités:

3.1.1 Types de caractères:

Du point de vue linguistique l'alphabet arabe se divise en deux types de caractères [19]:

- **Les caractères greffés** sont nommés accessoires, زوائد, parce qu'ils servent à former les différentes inflexions grammaticales des verbes et des noms, ainsi que les mots dérivés des racines (radicales). Les caractères greffés identifiés par SOUILEM .D [18] sont : ا, ب, و, ت, س, ن, م.
- Les autres caractères de l'alphabet forment l'ensemble des **caractères radicaux**. Ils ne servent à aucune fonction grammaticale, et constituent seulement des verbes racines. Il faut remarquer qu'un caractère greffé peut jouer le rôle d'un caractère radical alors qu'un radical ne peut jamais être greffé.

Les lettres de base	Noms	Valeurs Phonétiques (IPA)
ء	hamza	[ʔ]
ا	ʾalif	[æː]
ب	bāʾ	[b]
ت	tāʾ	[t]
ث	thāʾ	[θ]
ج	jīm	[ʒ] / [dʒ] / [g]
ح	ḥāʾ	[ħ]
خ	khāʾ	[χ]
د	dāl	[d]

ذ	dhāl	[ð]
ر	rāʾ	[r]
ز	zāy	[z]
س	sīn	[s]
ش	shīn	[ʃ]
ص	ṣād	
ض	ḍād	[ḍ]
ط	ṭāʾ	[ṭ]
ظ	ẓāʾ	[ð̣]
ع	ʿayn	[ʕ]
غ	ghayn	[ɣ]
ف	fāʾ	[f]
ق	qāf	[q]
ك	kāf	[k]
ل	lām	[l]
م	mīm	[m]
ن	nūn	[n]
ه	hāʾ	[h]
و	wāw	[w] / [uː]
ي	yāʾ	[j] / [iː]

Tableau 3.1 *Les lettres de base de la langue Arabe.*

Les caractères additionnels	Noms	Valeurs
آ	ʾalif madda	[ʔaː]
ة	tāʾ marbūṭa	[a], [at]
ى	ʾalif maqṣūra	[aː]
لا	lām ʾalif	[laː]
أ	ʾalif hamza	[aː], [uː], [ʔ]
إ		[iː]
ؤ	wāw hamza	[uː]
ئ ,ـئ ,ـئ	hamza médiane	[iː], [ʔ]

Tableau 3.2 *Les caractères additionnels de la langue Arabe*

3.1.2 Notion de racine:

Une famille de mots peut ainsi être générée d'un même concept sémantique à partir d'une seule racine à l'aide de différents schèmes. Ce phénomène est caractéristique à la morphologie arabe. On dit donc que l'arabe est une langue à racines réelles à partir desquelles on déduit le lexique arabe selon des schèmes qui sont des adjonctions et des manipulations de la racine.

La racine est formée de trois lettres, la première étant nommée le 'FA', ف, du verbe, la deuxième le 'AYN', ع, du verbe et la troisième le 'LAM', ل, du verbe.

3.1.3 Le schème

Le schème est un modèle constitué d'un ensemble de symboles dans lequel vient se couler la racine pour former le mot. Il joue un rôle important dans la dérivation en langue Arabe, celle-ci permettant à partir d'une racine consonantique – représentant la notion de base – d'obtenir les parentés sémantiques.

Exemple :

Mot Arabe	*racine*	*prononciation*
كتب	Fa'ala (فعل)	Kataba
كتابة	Fe'ala (فعالة)	Ketaba
كاتب	F'ael (فاعل)	Kateb
مكتوب	Maf'ool (مفعول)	Maktoub
كتاب	F'aal (فعال)	Ktaab
مكتبة	Maf'ala (مفعلة)	Maktaba
مكتب	Maf'al (مفعل)	

Tableau 3.3 Différents schèmes pour les mots كتب (écrire)

3.2. Structure d'un mot:

Le lexique arabe comprend trois catégories de mots : verbes, noms et particules. En arabe un mot peut signifier toute une phrase grâce à sa structure composée qui est une agglutination d'éléments de la grammaire, la représentation suivante schématise une structure possible d'un mot (de droite vers la gauche).

Post fixes	***Suffixes***	***Corps schématique***	***Préfixes***	***Antéfixes***

Figure 3.1 Structure du mot

- **Antéfixes** **:** sont des prépositions ou des conjonctions.
- **Préfixes et suffixes :** expriment les traits grammaticaux et indiquent les fonctions : cas du nom, mode du verbe et les modalités (nombre, genre, personne,…)
- **Post fixes** **:** sont des pronoms personnels.

3.2.1 Catégories des mots:

3.2.1.1 Les verbes:

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.

L'Arabe possède un système complet de dérivés verbaux, les 'formes augmentées' pour exprimer l'intensité, le but, la réciprocité, etc. Ils sont créés par modification, par redoublement de la deuxième consonne, par

adjonction et même par intercalation d'affixe. Le tableau 3.4 présente les dix formes les plus utilisées A. Saron [20].

Classe	Racine (c_1, c_2, c_3)	Description	Exemple
Semi-normale	Redoublée	Les secondes et troisièmes consonnes sont identiques	جدّ : JDD
	Hamzé	L'une des trois consonnes et une Hamza	قرأ : QRE
Anormale	Assimilée	La première consonne est WAW ou YA	وصل : WSL
	Concave	La seconde consonne est WAW ou YA	كون : KWN
	Défectueuse	La troisième consonne est WAW ou YA	دعو : DCW
	Repliée groupée	Racine à la fois concave et défectueuse	شوى : *WY
	Repliée séparée	Racine à la fois assimilée et défectueuse	وقى : WQY
Normale	Saine	Racine n'appartenant ni à la classe semi-normale ni à la classe anormale	كتب : KTB

Tableau 3.4 *Classification des racines trilitères.*

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).

Dans tous les mots qui dérivent de cette racine, on trouvera ces trois lettres K, T, B (voir Tableau 3.3).

La conjugaison des verbes se fait en ajoutant des préfixes et des suffixes, un peu comme en français

La conjugaison des verbes se fait en ajoutant des préfixes et des suffixes, un peu comme en français, la langue arabe dispose de trois temps:

1. **Le passé** **:** présente l'action passée et se distingue par des suffixes.

   Exemple: شَـربَ

شَـربَ + تُ = شَـربْـتُ	*j'ai bu*.
شَـربَ + تْ = شَـربَتْ	*elle a bu*.
شَـربَ + نا = شَـربْـنا	nous avons bu

2. **Le présent :** présente l'action en cours d'accomplissement, ses éléments sont préfixés et/ou des suffixés.

   Exemple: شَـربَ

أ + شَـربَ = أشْـرَبُ	*je bois*.
يـ + شَـربَ = يَـشْـرَبُ	*il bois*.
تـ + شَـربَ + ـيـنَ = تَـشْـرَبِـينَ	*tu bois* au féminin singulier.

3. **Le future** **:** correspond à une action qui se déroulera au futur et qui est marquée par l'antéposition de س *sa* ou سوف *sawfa.*

   Exemple: شَـربَ

سـ + أشْـرَبُ = سَـأشْـرَبُ	*je vais boire*.
سـ + تَـشْـرَبـانِ = سَـتَـشْـرَبـانِ	*vous allez boire*, au dual.

3.2.1.2 Les noms:

Les substantifs arabes sont de deux catégories, ceux qui sont dérivés de la racine verbale et ceux qui ne le sont pas comme les noms propres et les noms communs.

Dans le premier cas, le fait que le nom soit dérivé d'un verbe, exprime donc une certaine sémantique qui pourrait avoir une influence dans la sélection des phrases saillantes d'un texte pour le résumé.

La déclinaison des noms se fait selon les règles suivantes:

- **Le féminin singulier:** On ajoute le ة, exemple صغير *petit* devient صغيرة *petite*
- **Le féminin pluriel :** De la même manière, on rajoute pour le pluriel les deux lettres ات, exemple صغير *petit* devient صغيرا ت *petites*
- **Le masculin pluriel :** Pour le pluriel masculin on rajoute les deux lettres ين ou ون dépendamment de la position du mot dans la phrase (sujet ou complément d'objet), exemple : الراجع *revenant* devient الراجعين ou الراجعون *revenants*
- **Le Pluriel irrégulier:** Il suit une diversité de règles complexes et dépend du nom. Exemple : طفل *un enfant* devient أطفال *des enfants*. Le phénomène du pluriel irrégulier dans l'arabe pose un défi à la morphologie, non seulement à cause de sa nature non concaténative, mais aussi parce que son analyse dépend fortement de la structure comme pour les verbes irréguliers.

Certains dérivés nominaux associent une fonction au nom :

• **Agent** (celui qui fait l'action),
• **Objet** (celui qui a subit l'action),
• **Instrument** (désignant l'instrument de l'action),
• **Lieu**.

Pour les pronoms personnels, le sujet est inclus dans le verbe conjugué. Il n'est donc pas nécessaire (comme c'est le cas en français) de précéder le verbe conjugué par son pronom. On distinguera entre singulier, duel (deux) et pluriel (plus de deux) ainsi qu'entre le masculin et le féminin.

3.2.1.3 Les particules:

Ce sont principalement les mots outils comme les conjonctions de coordination et de subordination. Les particules sont classées selon leur sémantique et leur fonction dans la phrase, on en distingue plusieurs types (introduction, explication, conséquence, …). Elles servent à situer des faits ou des objets par rapport au temps ou au lieu.

4. Quelques travaux de catégorisation de textes arabes:

Dans la catégorisation de textes arabe il y a des travaux déjà réalisés, on peut citer:

- El-Kourdi et. al. [21] qui utilise l'algorithme Bayes naïve. La précision obtenue est égale à 67. 83 %:
  - Recall 71.96
  - Precision 67.88
  - F-measure 67.83

- *Siraj* de *Sakhr*.[22] Le système est disponible mais il n'y a pas de documentation ou spécification qui expliquent la méthode.
- Sawaf et. al. [23] qui utilise l'entropie maximum. La précision obtenue est égale à 62.70%:
  - Recall 84.20
  - Precision 50.00
  - F-measure 62.70
- El-Halees [24] utilise une méthode basée sur les règles d'association pour catégoriser les textes arabes. La précision obtenue est égale à 74.41%:
  - Recall 74.48
  - Precision 74.34
  - F-measure 74.41

- **Alaa M. El-Halees** [25], qui utilise l'entropie maximale pour la catégorisation des textes arabes, la précision obtenue est égale à 80,41%.
  - Recall 80.43
  - Precision 80.34
  - F-measure 80.41

5. Conclusion:

L'arabe est la langue parlée à l'origine par les Arabes qui est très différente des langues européennes. C'est une langue sémitique et flexionnelle dont l'alphabet est un abjad. Elle compte 28 consonnes et s'écrit et se lit de droite à gauche.

Un mot arabe s'écrit avec des consonnes et des voyelles. Les voyelles sont ajoutées au-dessus ou au-dessous des lettres

La plupart des mots arabes sont extrait à partir d'une racine de 3 caractères en ajoutant ou pénétrant des lettres ce qui engendre de nouveau mots en utilisant des schèmes.

Les mots arabes se divisent en trois groupes : les verbes (passé, présent et future), les noms (dérivés, non dérivés) et les particules.

Chapitre VI

Conception du système proposé

CHAPITRE 04 – Conception du système proposé

1. Introduction:

Dans ce chapitre nous présenterons notre approche pour la catégorisation de textes. Qui comporte deux étapes. Premièrement: le processus de prétraitement qui se divise en deux sous étapes : prétraitement linguistique et prétraitement statistique. Deuxièmement: une formalisation des SVM pour l'appliquer à la catégorisation de textes.

2. Architecture du Système:

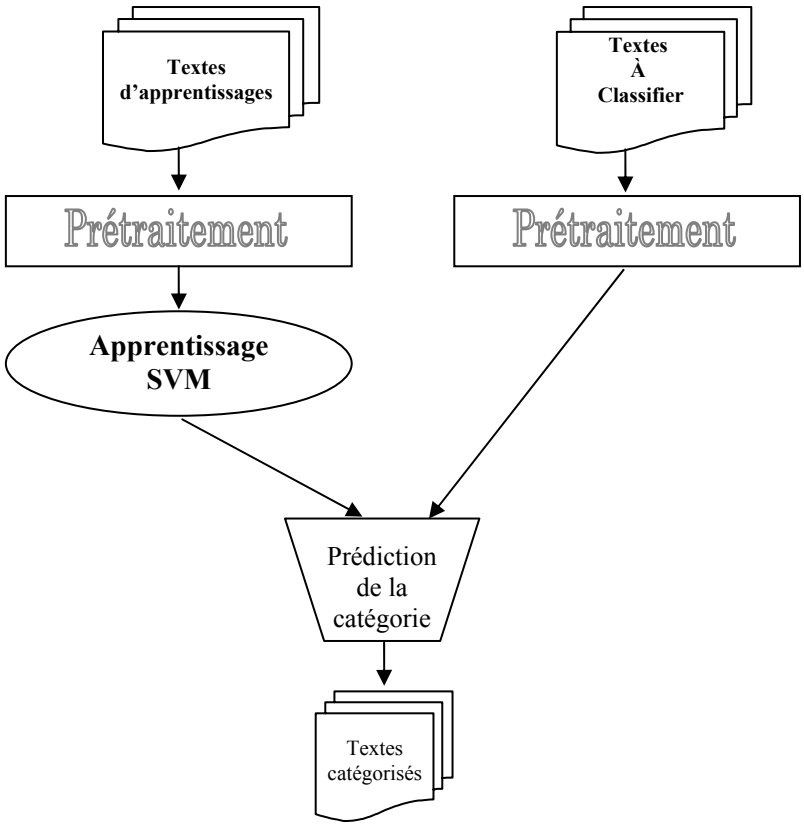


Figure 4.1 Architecture Globale du Système

Ce schéma décrit la mise en œuvre fonctionnelle de notre système de catégorisation.

Notre système est composé de deux grandes étapes:

1. ***L'apprentissage***: c'est principalement par apprentissage automatique que l'on tente de résoudre le problème de la catégorisation automatique de textes. Dans cette optique, plusieurs algorithmes mis au point pour des problèmes quelconques en apprentissage automatique ont été adaptés et appliqués dans notre domaine de recherche. L'objectif est de trouver une liaison fonctionnelle, que l'on appelle également ***modèle de prédiction***, entre les textes à classer et l'ensemble des catégories. Pour estimer le modèle de prédiction, il faut disposer d'un ensemble de textes préalablement étiquetés, dit ensemble d'apprentissage, à partir duquel on estime les paramètres du modèle de prédiction le plus performant possible, c'est-à-dire qui produit le moins d'erreurs en prédiction. Dans notre travail l'algorithme d'apprentissage utilisé est les SVMs (Support Vector Machine) [16].
2. ***La catégorisation d'un nouveau texte*** en appliquant le modèle de prédiction générée dans la phase d'apprentissage pour prédire la classe de ce texte. Le modèle, en entrée, reçoit un texte et, en sortie, lui associe une étiquette (classe).

2.1 Processus de prétraitement:

Tous texte soumis au système, que se soit en phase d'apprentissage, ou en phase de classification, est traité dans le but d'en extraire les fréquences des occurrences des termes définissant le domaine d'intérêt.

L'idée consiste à représenter les textes dans un espace approprié.

Elle comporte des processus de :

- **Prétraitement linguistique**: Élimination des mots inutiles (mots vides, caractères spéciaux,…) et extraction des radicaux (lemmatisation).
- **Prétraitement statistique**: calcule les poids des attributs des vecteurs représentant les textes.

Notre processus de prétraitement est présenté par le schéma suivant:

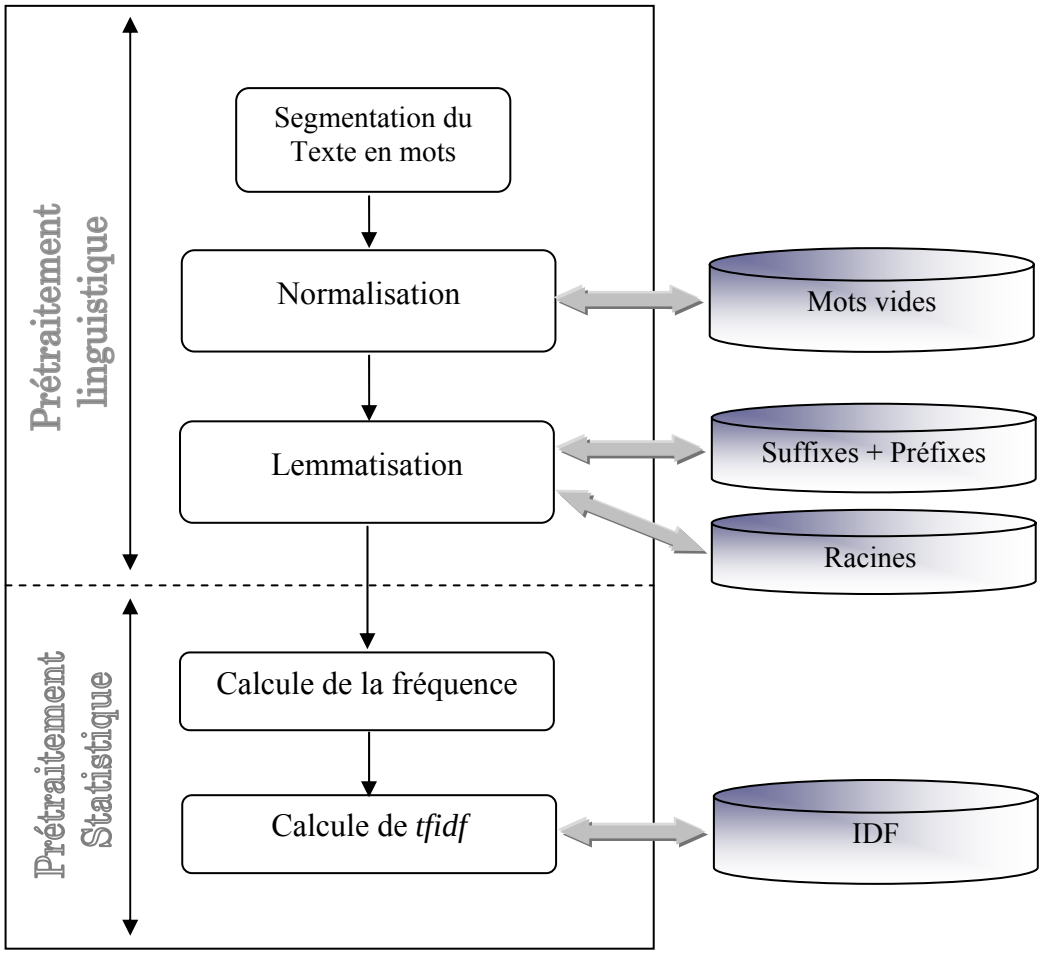


Figure 4.2 Processus de Prétraitement

1. **Base des données des mots vides:** contient les mots arabes non significatifs
2. **Base des données des suffixes et préfixes:** contient les suffixes et préfixes les plus utilisés dans la langue arabe
3. **Base des données des racines:** contient la liste des racines de la langue arabe
4. **IDF:** Fréquence des Textes par rapport à un terme.

2.1.1 Prétraitement linguistique:

Un problème central pour les approches statistiques de la catégorisation de textes est la grande dimension de l'espace de représentation. L'objet de prétraitement linguistique est la réduction de cet espace en éliminant les mots outils (stop words, caractères spéciaux,…), les "mots vides", et supprimer les éléments flexionnels (antéfixes, préfixes, suffixes, post fixes) dans le mot, pour chercher le racine.

Notre processus de prétraitement linguistique est présenté par L'algorithme suivant:

a. Segmentation du texte en mots:

Découpage du texte en mots en fonction des délimiteurs (espace, fin de ligne,…),

Code VB:

```
Texte = Split (Texte, délimiteur)
Délimiteur = { " ","/n"}
```

b. Normalisation:

1 - Suppression des mots non arabes, des nombres et des caractères spéciaux.

Code VB:

```
Texte1 = CleanWords (Texte)
Texte : Texte contient des mots non arabes, des
        nombres et des caractères spéciaux
Texte1: Texte nettoyé
```

2 – Suppression des mots dont leurs longueurs est inférieur à trois: ceci est pour les mots qui n'appartiennent pas à la base des données des racines.

Code VB:

```
If Len(text(i)) < 3 And FindMatch(buffer, text(i)) = 0
Then text(i) = " "
FindMatch(buffer, text(i)) = 0 : vérifier si le mot
appartient à la base des données des racines.
```

3 -

- ✓ Remplacement de آ, أ et إ par ا.
- ✓ Remplacement de ة par ه

Code VB:

```
Replace (Texte, ReplaceThis, WithThis)
Exp:
        Texte= "أكل"
        ReplaceThis = "إ"
        WithThis    = "ا"
Replace (Texte, ReplaceThis, WithThis) = "اكل"
```

4 – Suppression des mots vides consiste à éliminer tous les mots non significatifs. Pour chaque mot reconnu, on le compare avec un des éléments dans la base des données des mots vides qui contient tous les mots vides. Si un mot en fait parti, il ne sera pas pris en considération pour le calcul de sa fréquence.

Code VB:

```
If FindMatch(buffer1, text(i)) <> 0 then
:
:
End if
```

FindMatch(buffer1, text(i)) <> 0 : vérifier si le mot n'appartient pas à la base des données des mots vides.

c. Lemmatisation:

1 – Suppression des articles de définition

Code VB:

```
If Left(texte(i), 2) = "ال" Then
 str1 = Right(text(i), Len(text(i)) - 2)
Exp:
    Texte= "الاكل"
    Left(texte(i), 2) = "ال"
    Len(text(i)) = 5
    Right(text(i), Len(text(i)) - 2) = Right("الاكل", 5 - 2)
                                     = "اكل"
   str1 = "اكل"
```

2 – Suppression récursive des suffixes et préfixes: Pour un mot significatif, on applique une lemmatisation légère qui consiste à essayer de déceler si des préfixes ou suffixes ont été ajoutés au mot.

Le tableau suivant présente les affixes les plus utilisés en langue arabe:

Suffixes	***Préfixes***
هن، كن، هما، ك، كم، هم، نا، ها، تي، ي، ه، تان، تين، يون، تما، تا، وا، ين، ون، ان، ات، و، ي، ا، ن،ت، نا، تن، تم	ا، ن، ي، ت، كال، فال، بال، وال، وبال، فس، لل، ول، وب، ال، ولل، ب، و، ف، ك، وس، فل، فب، ل، سي، ست

Tableau 4.1 *les préfixes et suffixes les plus fréquents*

3 – Extraction de la racine de mot : pour extraire la racine de mot on cherche les schèmes appropriés à partir de la liste des schèmes arabes (**فعول ،فاعل ،مفعول، مفعال، مفعل**,…)

Code VB:

Lem (mot, buffer)
Buffer: base des données des racines
Norm (mot)

Exp:

Lem ("المرابطون", buffer) = "ربط"
Schème =(مفاعل)
Norm ("نوم")= "نام"

4 – Exemple: Pour le mot "المرابطون"

1. On cherche le mot dans la liste des mots vide le mot "المرابطون" n'appartient pas à la liste des mots vides
2. On cherche le mot dans la liste des racines le mot "المرابطون" n'appartient pas à la liste des racines
3. Extraction de la racine:

Entrée	**ن**	**و**	**ط**	**ب**	**ا**	**ر**	**م**	**ل**	**ا**
Elimination des suffixes et préfixes									
			ط	**ب**	**ا**	**ر**	**م**		
Chercher le schème approprié									
schème			**ل**	**ع**	**ا**	**ف**	**م**		
			ط	**ب**		**ر**			
le mot "ربط" appartient pas à la liste des racines donc la racine ="ربط"									

2.1.2. Prétraitement statistique:

Chaque fois qu'il est question de définir un problème de façon à assurer un traitement automatique, il est impossible de passer outre l'étape où il faut choisir la façon dont on va représenter le problème. Dans le cas de la catégorisation automatique de textes, on doit opter pour une façon efficace de représenter les instances à traiter, soit les textes. Un grand nombre de chercheurs dans le domaine ont choisi d'utiliser une représentation vectorielle dans laquelle chaque texte est représenté par un vecteur de *n* termes pondérés. Les *n* termes sont tout simplement les *n* différents mots apparaissant dans les textes de l'ensemble d'entraînement. Cette approche est aussi appelée *«bag-of-words»*. On peut utiliser d'autres types d'attributs pour caractériser les vecteurs dont certains seront présentés plus loin. Il existe aussi plusieurs façons d'associer un poids à un terme. Il peut être tout simplement binaire (1 si le mot est présent dans le texte, 0 sinon). Il peut aussi représenter le nombre d'occurrences du mot dans le texte. Cependant, en procédant ainsi, on donne une importance trop grande aux termes qui apparaissent très souvent à travers toutes les classes et qui sont peu représentatifs d'une classe en particulier. Pour résoudre ce problème on a utilisé la méthode TFIDF (acronyme pour «term frequency inverse document frequency») [8]. Celle-ci donne plus d'importance aux mots qui apparaissent souvent à l'intérieur d'un même texte, ce qui correspond bien à l'idée intuitive que ces mots sont plus représentatifs du document. Mais sa particularité est qu'elle donne également moins de poids aux mots qui appartiennent à plusieurs documents, pour refléter le fait que ces mots ont un faible pouvoir de discrimination entre les classes.

Le poids d'un terme t_k dans un texte d_j est calculé ainsi :

$$tfidf\left(t_k, d_j\right) = \#\left(t_k, d_j\right).\log\frac{|Tr|}{\#\left(t_k\right)}$$

Où

- $\#\left(t_k, d_j\right)$ est la fréquence de t_k dans d_j

  $\#\left(t_k, d_j\right) = Nb\,(t_k)/Nb(t_i) : t_i \in d_j$

  $Nb\,(t_k)$: *nombre d'occurrence du terme* t_k dans le Texte dj

  $Nb(t_i) : t_i \in d_j$: *nombre de termes* dans le Texte dj

- $|Tr|$ est le nombre de textes d'entraînements de la collection d'apprentissage.
- $\#\left(t_k\right)$ est le nombre de textes d'entraînements dans lesquels t_k apparaît au moins une fois.

1. Calculer le nombre d'occurrence de chaque terme dans le texte et le sauvegarder dans le fichier freq.txt sous le format:

<attribut>:<Nb> < attribut >:<Nb> … < attribut >:<Nb>

<attribut > : la valeur correspondante du terme dans le vecteur des termes

<Nb> : Nombre d'occurrence d'attribut

Code VB:

```
if texti(i) <> " " Then
  chang = o
     v = o
     w = o
     stri = texti(i)
     For j = LBound(texti) To k
        If stri = texti(j) And texti(j) <> " " Then
          v = v + 1
          texti(j) = " "
        End If
     Next j
     For j = LBound(vecteur) To UBound(vecteur)
        If stri = vecteur(j) Then
```

```
                    w = j
                    j = UBound(vecteur)
                    chang = 1
                End If
            Next j
            If chang = 1 Then
                str2 = str(w) & ":" & str(v)
                str2 = Replace(str2, " ", "")
                RichTextBox1.text = RichTextBox1.text + " " + str2
            End If
        End If
    Open "c:\freq.txt" For Output As #1
    Print #1, RichTextBox1.text
    Close #1
```

2. pour chaque terme dans le vecteur des termes on calcule le nombre textes d'entraînements dans lesquels t_k apparaît au moins une fois, et le sauvegarder dans le fichier idf.txt.

Code VB:

```
n = compt(i)
    str1 = tempstring1(j)
    k1 = NB(j)
    str4 = tfidf(str1, i, n, n1, k1)
    If str4 <> " " Then
        str2 = str(n1)
            If strpred <> str2 Then
                str2 = Trim(str2)
                str3 = str(i)
                str3 = Trim(str3)
                str2 = str3 + ":" + str2
                str4 = Clean(str4)
                str1 = Replace(str1, str2, str4)
                tempstring1(j) = str1
                strpred = str2
            End If
     End If
Open "c:\idf.txt" For Output As #1
Print #1, RichTextBox1.text
Close #1
```

tfidf (str1, i, n, n1, k1) : fonction qui calcule le poids P du terme i.

n : le nombre de textes d'entraînements dans lesquels i apparaît au moins une fois.

n1: nombre d'occurrence du terme i dans le Texte dj.

k1: nombre de termes dans le Texte dj.

2.2 Apprentissage avec SVM:

2.2.1 Constitution de corpus:

La constitution de ce corpus a été un travail long et difficile, malgré nos recherches, nous n'avons pas trouvé de corpus arabe étiqueté en classes comparables. Dans cette section nous présenterons le corpus de textes qui a été utilisé au cours de notre apprentissage et test. Une attention particulière a été portée quant au choix de ces collections. Une de nos préoccupations était de choisir des ensembles de textes facilement accessibles et disponibles collectées de sites très connus.

Il était alors intéressant de s'assurer que la méthode proposée pouvait être utile dans différentes situations, dans des contextes variés. Pour cela on a choisi cinq catégories de domaines différents (Sociologie, Touriste, Science et Santé, Recette, Sport).

Un autre aspect assez important était la taille de ces corpus. Pour que nos résultats soient statistiquement significatifs, nos ensembles de tests devaient contenir un nombre suffisant de Textes.

Une remarque est à faire concernant la division de nos corpus en ensembles d'entraînement et de test. Les publications faisant état de travaux en catégorisation automatique de textes tentent lorsque c'est possible d'utiliser des divisions entraînement/test standard pour faciliter la comparaison de leurs résultats. Le premier ensemble de textes utilisé provient des sites de presses connues:
http://www.aljazeera.net, http:// www.alarabiya.net, http:// www.elaph.com, http://www.newstin.ae, www.alriyadh.com, http://news.naseej.com ,. Qui rassemblent des articles dans presque tous les domaines connus. Le site http:// *www.swmsa.com* inclus une collection importante de documents dans le domaine sociologie et les deux sites, http:// www.kooora.com, *http://www.aljazeerasport.net* rassemblent des documents dans le domaine sport.

Pour nos expérimentations, des ensembles d'entraînement de taille variable allant de 100 à 1000 Textes, pour chaque catégorie, ont été constitués en choisissant les premiers Textes, l'ensemble des tests comporte des ensembles de textes de taille variable allant à 300 Textes, pour chaque catégorie.

2.2.2 : Algorithme d'apprentissage:

La plupart des techniques du ML possèdent un (trop) grand nombre de paramètres d'apprentissage à fixer par l'utilisateur (structure d'un réseau de neurones, cœfficient de mise à jour du gradient,. . .). De plus, avec ces méthodes, le nombre de paramètres γ_i à calculer par l'algorithme d'apprentissage est en relation linéaire, voire exponentielle, avec la dimension de l'espace d'entrée. La formulation élégante de SVM laisse très peu de place aux paramètres utilisateurs et le nombre de paramètres γi est linéaire en la taille du training set.

SVM est donc une méthode de classification particulièrement bien adaptée pour traiter des données de très haute dimension telles que les textes et les images.

La réalisation d'un programme d'apprentissage par SVM se ramène à résoudre un problème d'optimisation impliquant un système de résolution dans un espace de dimension conséquente. L'utilisation de ces programmes revient surtout à sélectionner une bonne famille de fonctions noyaux et à régler les paramètres de ces fonctions. Ces choix sont le plus souvent faits par une technique de validation croisée, dans laquelle on estime la performance du système en la mesurant sur des exemples n'ayant pas été utilisés en cours d'apprentissage.

2.2.3 Problème d'apprentissage

On s'intéresse à un phénomène f (éventuellement non déterministe) qui, à partir d'un certain jeu d'entrées x, produit une sortie y = f(x).

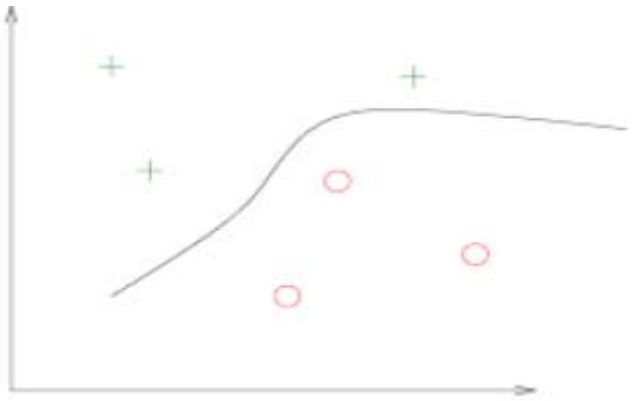
Le but est de retrouver cette fonction f à partir de la seule observation d'un certain nombre de couples entrée-sortie {(xi; yi) : i = 1, …, n} afin de « prédire » d'autres évènements.

On considère un couple (X, Y) de variables aléatoires à valeurs dans X x Y.

Seul le cas Y = {-1, 1} (classification) nous intéresse ici (on peut facilement étendre au cas sachant qu'on peut observer un échantillon S = {(X1, Y1),... ,(Xn, Yn)} de n copies indépendantes de (X, Y), on veut: construire une fonction h : X → Y telle que P(h(X) != Y) qui soit minimale.

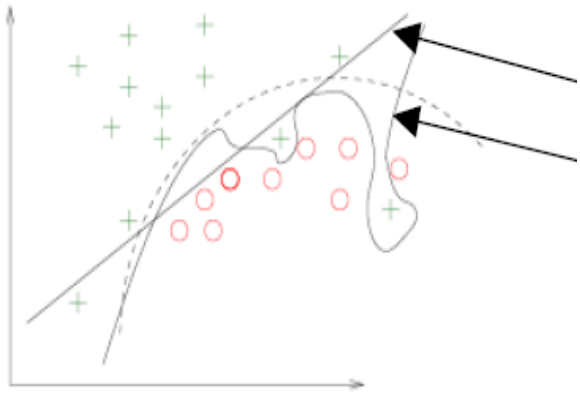
Trouver une frontière de décision qui sépare l'espace en deux régions (pas forcément connexes).

Connaissant h, on peut en déduire la classification des nouveaux points c'est à dire trouver une frontière de décision.



Le problème est de trouver une frontière assez éloignée des points de différentes classes. C'est ce qui constituera l'un des problèmes majeurs de la classification grâce aux SVMs.

Sur et sous apprentissage :



Si les données sont générées par un modèle quadratique :
Le modèle linéaire est en situation de sous apprentissage.
Le modèle de haut degré est en situation de sur apprentissage (Apprentissage par coeur)

Il faut donc trouver un compromis entre adéquation aux données et complexité pour pouvoir généraliser.

Classification à valeurs réelles

Plutôt que de construire directement

$$h : \boldsymbol{X} \rightarrow \{-1, 1\},$$

On construit :

$$f : \boldsymbol{X} \rightarrow \boldsymbol{R} \text{ (ensemble des réels).}$$

La classe est donnée par le signe de f ;

$$h = \text{signe}(f) .$$

L'erreur se calcule avec $P(h(X) != Y) = P(Yf(X) \leq 0)$.

Ceci donne une certaine idée de la confiance dans la classification.
Idéalement,

$|Yf(X)|$ est proportionnel à $P(Y|X)$.$Yf(X)$ représente la marge de f en (X,Y).

Le but à atteindre est la construction de f et donc h. Nous allons voir comment y parvenir.

Maximisation de la marge

La marge est la distance du point le plus proche à l'hyperplan.

Dans un modèle linéaire (cf. figure ci-dessus), on a:

$$f(x) = w.x + b.$$

L'hyperplan séparateur (frontière de décision) a donc pour équation:

$$w.x + b = 0.$$

La distance d'un point au plan est donnée par:

$$d(x) = |w.x + b| / ||w||$$

L'hyperplan optimal est celui pour lequel la distance aux points les plus proches (**marge**) est maximale.

Soient x1 et x2 eux points de classes différentes :

$$(f(x1) = +1 \text{ et } f(x2) = -1)$$
$$(\boldsymbol{w}.\boldsymbol{x}1) + b = +1 \text{ et } (\boldsymbol{w}.\boldsymbol{x}2) + b = -1$$

Donc:

$$(\boldsymbol{w}.(\boldsymbol{x}1 - \boldsymbol{x}2)) = 2$$

D'où :

$$(w/||w|| . (\boldsymbol{x}1 - \boldsymbol{x}2)) = 2/||w||.$$

On peut donc en déduire que maximiser la marge revient à minimiser ||w|| sous certaines contraintes que nous verrons dans les paragraphes suivants.

Problème primal

Un point (x; y) est bien classé si et seulement si yf(x) > 0

Comme le couple (w, b) est défini à un coefficient multiplicatif près, on s'impose yf(x) ≥ 1

On en déduit (en s'appuyant également sur le paragraphe précédent), le problème de minimisation sous contraintes suivantes :

$$\begin{cases} \min \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \forall i,\ y_i(\boldsymbol{w}.\boldsymbol{x}_i + b) \geq 1 \end{cases}$$

Problème dual

On passe du problème primal au problème dual en introduisant des multiplicateurs de Lagrange pour chaque contrainte.

Ici on a une contrainte par exemple d'apprentissage

$$\begin{cases} \max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j \\ \forall i, \alpha_i \geq 0 \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases}$$

C'est un problème de programmation quadratique de dimension n (nombre d'exemples).

On défini ainsi la matrice suivante appelée « matrice hessienne » : (xi.xj)i,j qui représente la matrice des produits des entrées X (La notation matricielle permettant de résoudre plus facilement le problème en informatique).

2.2.4 Apprentissage et classification

Comme il a été introduit plus haut, nous réalisons une classification supervisée à l'aide de SVM. Ceci comprend une phase d'apprentissage et une phase de classification.

Pour l'apprentissage Nous nous appuyons sur une implémentation des SVM appelée SVM^{light} [26]. Il s'agit de 2 programmes C, un pour l'apprentissage et un autre pour la classification.

L'étape d'apprentissage prend en entrée, pour chaque couple de classes, un fichier (.DAT) avec les vecteurs de description de chaque texte et le label correspondant (+1 ou -1). Ce programme fournit en sortie le SVM appris à partir des exemples. Le programme de classification prend en entrée la SVM produite pendant l'apprentissage ainsi qu'un fichier texte contenant les vecteurs caractéristiques des textes à classer.

Pour cela il est nécessaire de transformer les textes d'apprentissage en une représentation souhaitable pour l'algorithme a'apprentissage SVM^{light}, dont

chaque classe est représentée par les textes auxquels appartient "In" et les textes auquel n'appartient pas "Out"

La classe science est présentée par In-science qui regroupe tous les textes qu'il appartient plus out-science qui regroupe tous les textes qu'il n'appartient pas à la classe science.figure.4.3.

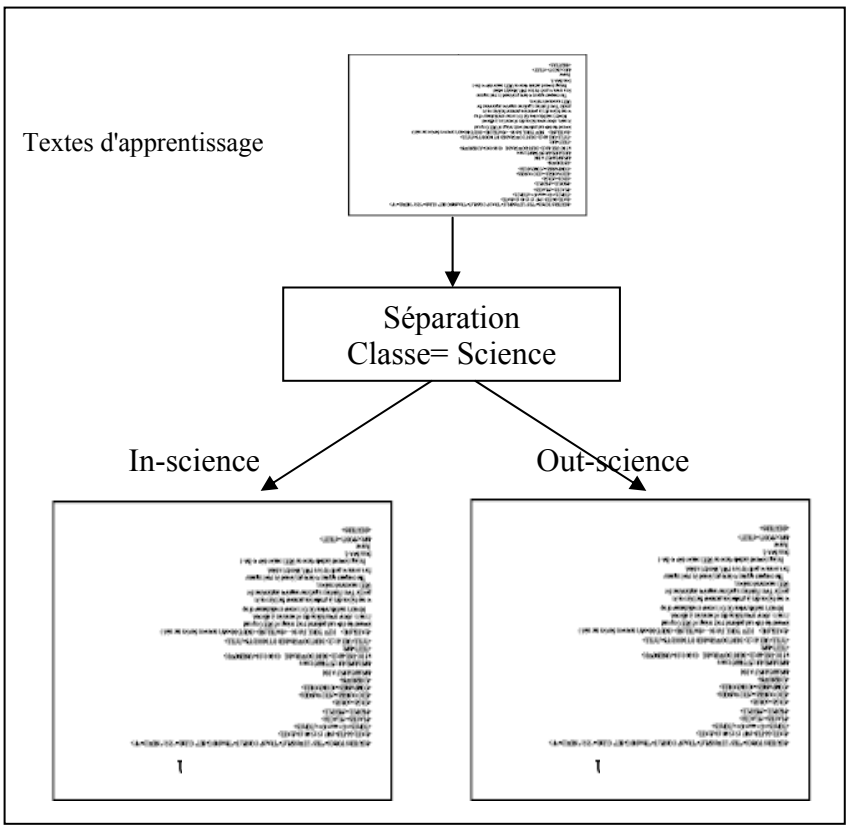


Figure 4.3 séparation des textes pour la classe Science

La deuxième étape consiste à transformer les textes d'apprentissage (chaînes de caractères) en une représentation convenable pour l'algorithme d'apprentissage SVM^{light} [26]. Donc chaque texte est représenté par un vecteur qui comporte les attributs et leurs valeurs TFIDF.

2.2.4.1 Réglage des paramètres

Nous avons utilisé pour l'apprentissage les paramètres suivants :

Paramètre de régularisation C:

La valeur du paramètre C est un hyper-paramètre qui régit la performance du SVM. Ce paramètre sert à fixer le compromis entre la minimisation de l'erreur d'apprentissage et la maximisation de la marge. En pratique, le comportement du SVM est sensible à la valeur de C uniquement si les données d'apprentissage ne sont pas séparables. Dans ce cas, il existe des valeurs critiques qui peuvent compromettre la performance du classifieur. Une très grande valeur de C (quelques milliers) peut faire que la fonction objective minimisée par le SVM ne soit plus convexe et empêcherait sa convergence. Une très faible valeur de C tend à diminuer la capacité du classifieur .

Dans ce travail, après quelques expériences, le paramètre C a été fixé à 100 car il offre les meilleures performances.

Réglage de epsilon "largeur de tube de régression" :

Erreur de critère de terminaison eps =[y [w*x+b] - 1]

Nous avons fixé *epsilon à* 10^{-3}

2.2.4.2 Choix du noyau

Trois noyaux ont été utilisés dans notre travail :

1. Polynomial $k(x, y) = (x.y + 1)^d$ / d : le degré du noyau
2. RBF (Radial Basis Function) $k(x, y) = e - \sigma(x\_y)^2$
3. Sigmoïde $k(x, y) = \tanh(x.y + 1)$ / *Tanh*: La fonction tangente hyperbolique

Pour évaluer l'influence du paramètre de chaque noyau sur la qualité de la classification, nous avons calculé le F-measure en fonction de nombre de textes d'apprentissages.

Les résultats sont présentés dans le tableau suivant :

Noyaux / catégories	Noyau polynomial			Noyau RBF			Noyau Sigmoïde
	P1	P2	P3	σ =1	σ =2	σ =3	
Sport	90,15	86,39	85,52	86,72	88,13	90,12	88,70
Touriste	85,53	84,39	84,01	84,74	84,48	86,07	87,35
Science et santé	77,77	74,74	73,27	73,98	71,44	66,19	75,07
Recette	93,51	92,22	90,89	93,35	92,31	85,91	95,23
sociologie	78,79	79,35	78,59	81,68	78,60	77,85	75,71

Tableau 4.2 F- measure des catégories calculé pour chaque type de noyau - Polynomial - RBF – Sigmoïde

Ces résultats sont présentés dans la figure suivante:

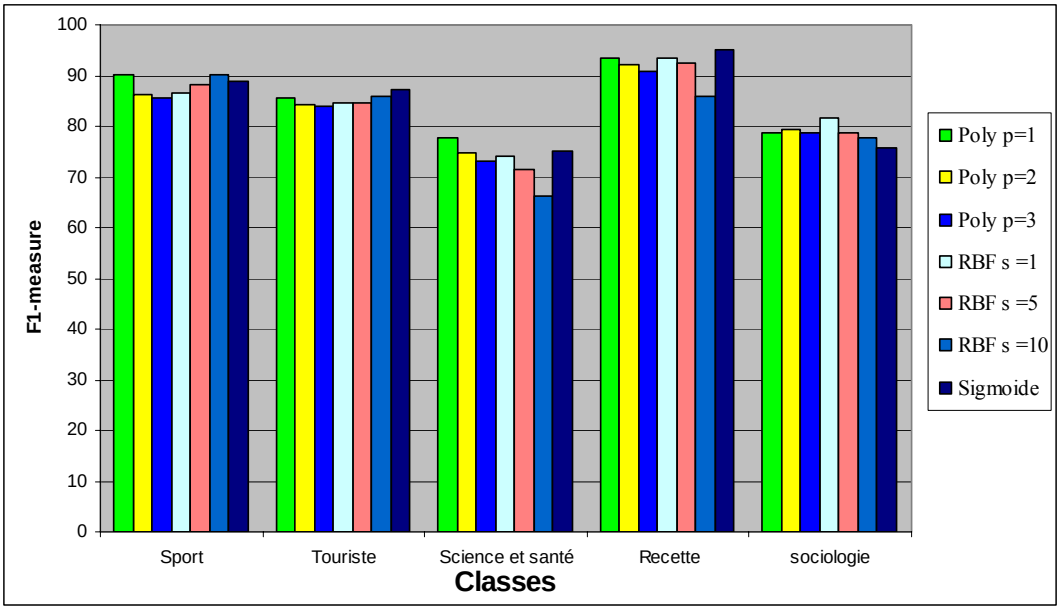


Figure 4.4 F- measure des catégories calculé pour chaque type de noyau - Polynomial - RBF – Sigmoïde

Nous présentons dans le tableau suivant les F-measures moyennes calculé pour chaque type de noyau

noyaux	RBF			polynomial			Sigmoïde
paramètres	σ =1	σ =2	σ =3	p=1	p=2	p=3	
F-measure	*84,09*	*82,99*	*81,23*	*85.15*	*83,42*	*82,45*	*84,41*

Tableau 4.3 F- measure moyenne calculée pour chaque type de noyau - Polynomial - RBF – Sigmoïde

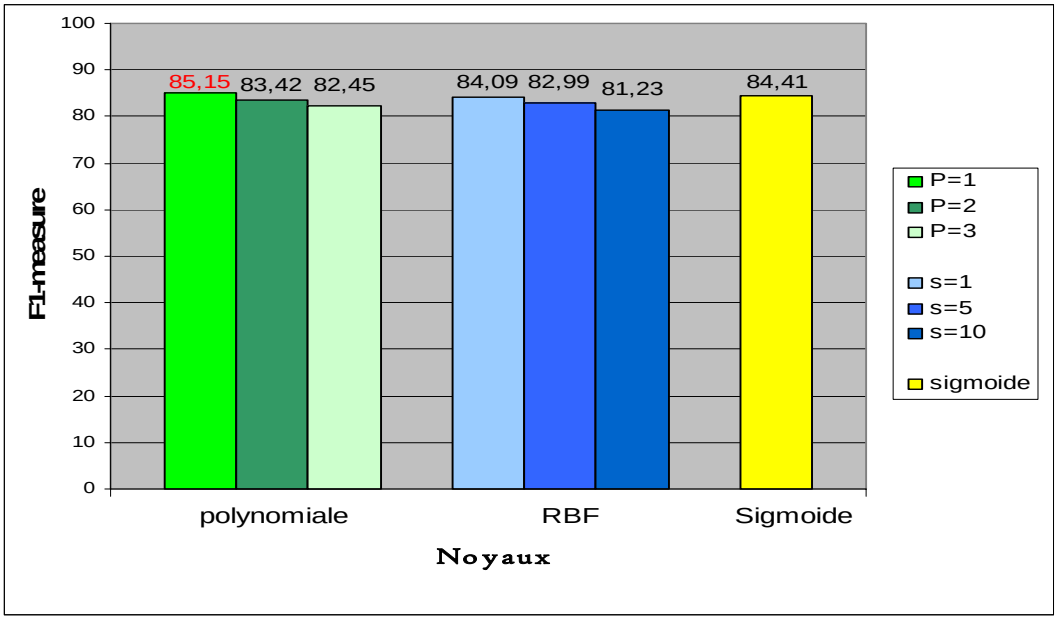


Figure 4.5 F- measure moyenne calculée pour chaque type de noyau
- Polynomial - RBF – Sigmoïde

Globalement, les taux sont sensiblement similaires. Cependant, nous constatons que le choix du noyau et ses paramètres influent sur le taux de classification. .Le noyau polynomiale de premier degré p=1 fournit les meilleurs résultats de catégorisation comparativement aux autres noyaux 84.66%.

2.2.5 Nombre de textes d'apprentissage:

On teste la performance de notre système avec l'apprentissage de notre système par différentes tailles de collections d'apprentissage

Le tableau suivant montre les F-measure calculés en fonction de nombre de textes d'apprentissages:

Nbre de Textes / Classe	500	1000	1500	2000	2500	3000	3500	4000	4500	5000
Sport	*46,72*	*62,75*	*69,71*	*74,05*	*76,58*	*78,29*	*83,28*	*85,81*	*88,67*	90,15
Touriste	*54,32*	*57,71*	*63,91*	*70,23*	*73,86*	*76,74*	*81,23*	*83,35*	*85,89*	85,53
Science et santé	*39,18*	*47,45*	*53,73*	*56,71*	*62,54*	*65,90*	*70,45*	*72,92*	*75,47*	77,77
Recette	*86,18*	*88,56*	*91,98*	*91,00*	*91,84*	*92,77*	*91,79*	*93,86*	*92,59*	93,51
Sociologie	*33,72*	*42,95*	*47,51*	*55,24*	*61,10*	*65,43*	*70,45*	*74,10*	*76,71*	78,79

Tableau 4.4 Effet de la taille de textes d'apprentissage sur les 5 catégories

Ces résultats sont présentés dans la figure suivante:

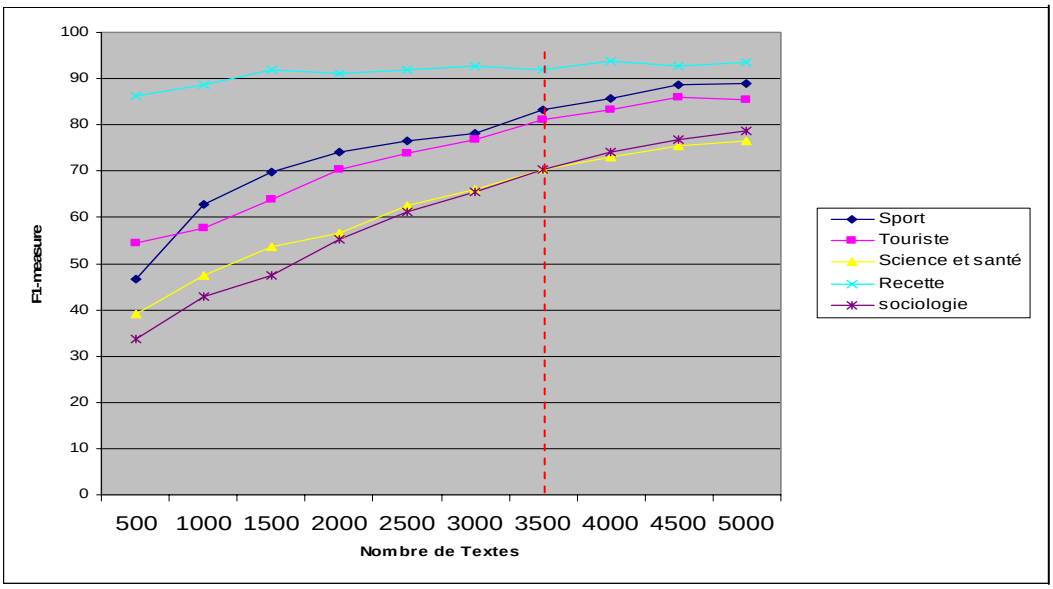


Figure. 4.6 Effet de la taille de textes d'apprentissage sur les 5 catégories

Le graphique montre la tendance de nombre d'exemples d'apprentissage utilisés. Le nombre minimum d'exemples d'apprentissage nécessaire pour les classes "sociologie, science et santé", "sport, touriste" et "recette" d'atteindre une mesure d'apprentissage F1-measure moyenne d'environ "70 %", "80%" et "90%" respectivement est de 3500 exemples (ligne pointillée montrée sur le schéma). La catégorie recette, qui est un domaine limité, entraîne très bien, même lorsqu'il y a un nombre restreint d'exemples d'apprentissage. Les catégories « sport», et «touriste » entraînent relativement bien, Les catégories « science » et « sociologie », qui sont des domaines très larges, n'entraînent pas bien avec peu de données d'apprentissage.

Chapitre V

Test et Résultats

CHAPITRE 05 – Test et Résultats

1. Les données de test:

Pour évaluer notre étude on a utilisé une collection qui comporte 1500 textes (300 pour chaque catégorie) choisis des pages Web arabe,

2. Evaluation des classifieurs:

Il est difficile d'évaluer globalement les logiciels de catégorisation et particulièrement de comparer différents classificateurs entre eux. Deux mesures sont néanmoins couramment employées, à savoir le rappel (recall) et la précision (precision) [27].

Pour évaluer notre système, on à calculer Recall et Precision. Ces probabilités peuvent être estimées à partir de la table de contingence ainsi:

		Jugement de Expert	
		Oui C'est correct	**Non C'est correct**
Jugement de classifieur	**Assigner Oui**	**VP**	**FP**
	Assigner Non	**FN**	**VN**

Tableau 5.1 Tableau de contingence

1. **VP** : Textes correctement classés comme appartenant à cette classe.
2. **FP** : Textes incorrectement classés comme appartenant à cette classe.
3. FN : Textes incorrectement rejetés, à partir de cette classe.
4. **VN** : Textes correctement rejetés, à partir de cette classe.

Le rappel (R) mesure la largeur de la catégorisation et correspond au ratio des documents bien classés par rapport à l'ensemble des documents appartenant réellement à la catégorie. Tandis que la précision (P) mesure la qualité de la catégorisation et correspond à la fraction des documents bien classés sur tous les documents affectés à la catégorie. Plus précisément, R et P sont calculés par les formules suivantes :

$$\text{Precision: } \mathbf{P} = \boldsymbol{VP / (VP + FP)}$$

$$\text{Rappel : } \mathbf{R} = \boldsymbol{VP / (VP + FN)}$$

Prenons un exemple pour une catégorie i. Un rappel de 1 signifie que tous les textes de la catégorie i du jeu de test ont bien été catégorisés par le processus de catégorisation. A l'inverse, une précision de 1 signifie que tous les textes classés dans la catégorie i appartenaient bien à cette catégorie. Un texte classé dans i par erreur baisse la précision de i. Un texte de i qui a été classé dans j, par erreur, baisse le rappel de i.

Pour évaluer un processus de catégorisation, on ne peut pas mesurer uniquement le rappel ou la précision car ces deux mesures n'ont aucune signification l'une sans l'autre. Pour prendre en compte à la fois le rappel et la précision, on utilise la plupart du temps la formule F-measure. Qui est une mesure statique standard utilisée pour évaluer le performance des classifieurs [28]. F-measure moyenne calculé à partir de Rappel et Précision [27]:

$$\mathbf{F\text{-}measure = 2RP / (R + P)}$$

3. Résultas:

On à calculer la performance pour chaque classe de cinq catégorie, le tableau suivant montre les résultats obtenu :

Catégories	Recall	Precision	F-measure
Sport	*93,15*	*94,11*	*93,63*
Touriste	*81,56*	*82,47*	*82,01*
Science et santé	*82,32*	*86,17*	*84,20*
Recette	*98,71*	*99,86*	*99,28*
Sociologie	*73,52*	*73,65*	*73,58*

Tableau 5.2 F-measure calculé de cinq catégories

A partir du tableau on peut dire que les meilleurs performances sont remarqués dans les deux catégories sport 93.63% et recette 99.28% car ces deux domaines ont un espace limité. Encor, la catégorie sociologie a une basse performance 73.58% car sociologie est un domaines très large.

Pour comparer notre système avec les autres, on a calculé Recall, Precision et F-measure total, le tableau suivant montre les résultats obtenus:

Système	Recall	Precision	F-measure
Notre système	*85,85*	*87,25*	*86,54*
ArabCat System	*80,48*	*80,34*	*80,41*
El-Halees	*74,48*	*74,34*	*74,41*
El-Kourdi et. al.	*71,96*	*67,88*	*69,83*
Sakhr's Categorizer	*73,78*	*47,35*	*57,68*
Sawaf et. al	*84,20*	*50,00*	*62,70*

Tableau 5.3 Comparaison de notre système avec d'autres

Les résultas obtenu montre que notre système a les meilleurs performances que les autres systèmes de catégorisation de textes arabes.

4. Scalabililité du système:

On test **la Scalabililité** du notre système avec l'apprentissage de notre système avec différentes taille de collections d'apprentissage et en enregistrer la performance pour chaque collection. Le nombre de textes utilisés pour le test est de 300 textes par catégorie.

Le tableau suivant montre les F-measure obtenus:

Nbre de Texte / Classe	500	1000	1500	2000	2500	3000	3500	4000	4500	5000
Sport	42,04	61,05	68,95	79,17	83,99	86,08	89,19	87,23	93,76	93,63
Touriste	47,71	54,83	59,04	68,35	71,92	77,65	83,44	80,98	81,81	82,01
Science et santé	21,88	35,96	47,76	54,23	64,08	73,09	77,12	80,36	83,05	84,20
Recette	88,00	91,52	91,11	95,26	98,04	98,66	98,90	99,51	98,01	99,28
sociologie	20,66	30,82	40,61	55,28	58,37	65,75	68,54	71,90	72,85	73,58

Tableau 5.4 F-measure calculé pour chaque collection d'apprentissage

Ces résultats sont présentés dans la figure suivante:

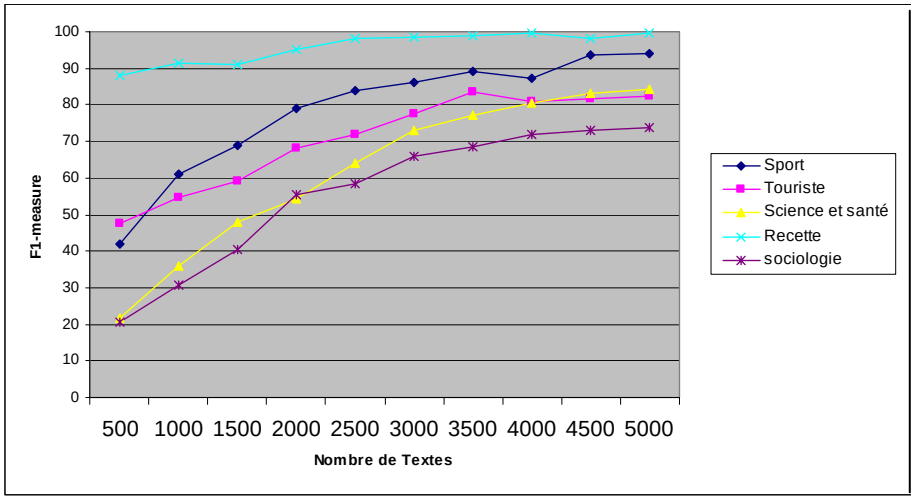


Figure 5.1 Scalabilité du système

De la figure 5.1 on remarque que la performance de notre système augmente relativement avec le nombre de texte d'apprentissage. On peut constaté que le système est scalaire linéairement avec la taille des données.

5. Discussion

Les résultats obtenus par nos modèles sur notre corpus sont des résultats encourageants.

Cette section propose de discuter les différents choix effectués pour chacune des étapes du processus afin de moduler la signification de nos résultats.

La première étape du processus consiste à définir une représentation du corpus par des termes. Nous avons choisi ici la représentation basée sur les mots (sac à mots) qui considère le mot comme une suite de caractère appartenant à un dictionnaire. Le texte est donc représenté par un vecteur, Ce choix était motivé par la simplicité de cette méthode avec la capacité des mots à capturer aisément le sens de texte.

La deuxième étape consiste à calculer les attributs des vecteurs, on utilise la méthode TFIDF (acronyme pour «term frequency inverse document frequency») qui est depuis longtemps utilisée dans les problèmes de catégorisation : bien qu'assez simple, elle donne de bons résultats.

Sa particularité est qu'elle donne plus d'importance aux mots qui apparaissent souvent à l'intérieur d'un même texte et donne également moins de poids aux mots qui appartiennent à plusieurs documents, pour refléter le fait que ces mots ont un faible pouvoir de discrimination entre les classes.

La dernière étape concerne l'apprentissage ; on a utilisé la méthode SVM "Support Vector Machine". Le succès de cette méthode est justifié par les solides bases théoriques qui la soutiennent aussi leur simple implémentation. Les résultats obtenus montre que l'utilisation des SVM donne des bons résultats et le teste de la scalabilité montre que, dans la future, notre système peut examiner des autres domaines et sous domaines.

6. Conclusion

Dans ce travail on a utilisé les machines à vecteurs de supports (SVM) pour la catégorisation des textes arabes, en débutant par le prétraitement des textes qui est une étape très importante dans le processus de catégorisation, grâce à la complexité et la richesse de la langue arabe ce processus devient très complexe.

Ensuite, la construction du modèle de catégorisation nécessite un prétraitement statistique qui consiste à transformer les textes (chaînes de caractères) en une représentation convenable pour l'algorithme d'apprentissage (un vecteur qui comporte les attributs et leurs valeurs).

Théoriquement les machines à vecteurs de supports (SVM) sont très efficaces dans le domaine de classification en générale grâce à:

- Fondement théorique solide.
- Mise en œuvre très simple.
- Temps de calcul courts.
- Qualité de prédiction généralement meilleure.

Pratiquement, les résultats obtenus montre que les SVM prouvent leur efficacité d'être un outil très important dans la catégorisation des textes arabe.

Finalement, on peut améliorer ce travail avec l'amélioration de processus de prétraitement linguistique en enrichir les bases de données qui est utilisées pour la lemmatisation des mots et en s'intéresse au aspect sémantique des mots arabe.

Bibliographie

BIBLIOGRAPHIE:

[1] Lewis, D., Naïve (Bayes) at forty: The Independent Assumption in Information Retrieval. In Machine Learning: ECML-98, 10th European Conference on Machine Learning. p 4-15 (1998).

[2] Yang, Y., An Evaluation of Statistical Approaches to Text Categorization. Journal of Information Retrieval. (1999).

[3] Nigam, K., Lafferty, J., McCallum, A., Using Maximum Entropy for Text Classification. In IJCAI-99 Workshop on Machine Learning for Information Filtering, pp. 61-67. (1999).

[4] Joachims, T., Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In Proceedings of ECML-98, 10th European Conference on Machine Learning. Pages 137-142. (1998).

[5] **Patrick Gallinari, Hugo Zaragoza, Massih Amini "***Apprentissage et données textuelles***", Université Paris 6, 4 Place Jussieu 75252 Paris cedex 05** Patrick.Gallinari@lip6.fr

[6] Hammo, B., Abu-Salem, H., Lytinen, S., Evens, M., *QARAB: A Question Answering System to Support the Arabic Language*. Workshop on Computational Approaches to Semitic Languages. ACL 2002, Philadelphia, PA, July. p 55-65 (2002).

[7] **Radwan JALAM,** *"Apprentissage automatique et catégorisation de textes multilingues"* **UNIVERSITÉ LUMIÈRE LYON 2** - juin 2003.

[8] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1) :1–47. Available from World Wide Web : `http://faure.iei.pi.cnr.it/~fabrizio/Publications/ACMCS02.pdf`.

[9] **Yang** & J.O. Pedersen. *"A comparative study on feature selection in text categorization". Proceedings of ICML-97, 14th International Conference on Machine Learning*, pp. 412-420, Morgan Kaufmann, San Francisco, US, 1997.

[10] **Patrick Gallinari, Hugo Zaragoza, Massih Amini "***Apprentissage et données textuelles***", Université Paris 6, 4 Place Jussieu 75252 Paris cedex 05** *Patrick.Gallinari@lip6.fr*

[11] **Charniak E**, *"Statistical language learning* ", MIT Press. (1993)

[12] **Jérome CALLUT**. *"Implémentation efficace des Support Vector Machines pour la classification"*, **UNIVERSIT´E LIBRE DE BRUXELLES - 2002-2003**

[13] **F. Peng and D. Schuurmans**. *"Combining naive bayes and n-gram language models for text classification"*, 2003.

[14] **J. R. Quinlan.** *"C4.5: Programs for Machine Learning. Morgan Kaufmann"*, 1993.

[15] **J. R. Quinlan. Bagging, boosting, and C4.5**. *"In Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference"*, pages 725–730, Menlo Park, 1996.AAAI Press / MIT Press.

[16] **VAPNIK, V**.*"The Nature of Statistical Learning Theory"*. Springer, New York. (1995)

[17] **Mohamadally Hasan, Fomani Boris.** *"SVM : Machines à Vecteurs de Support ou Séparateurs à Vastes Marges*", BD Web, ISTY3 Versailles St Quentin, France -2006

[18] **B. Schölkopf, A.J. Smola**, *"Learning with kernels"*, MIT Press, 2002.

[19] **SOUILEM .D. Truquet M. et Caussee B.,** *"Un système d'enseignement assisté par Ordinateur de la grammaire arabe* "S.E.A.G.A". 1989

[20] **A. Sharon 1989.** *Prosodic Constituency in the Lexicon***.** Doctoral dissertation, Stanford University.

[21] **El-Kourdi, M., Bensaid, A., Rachidi,** T **. Dissertation, Stanford University.**., *Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm*. 20th International Conference on Computational Linguistics . August 28th. Geneva (2004).

[22] www.siraj.sakhr.com

[23] **Sawaf, H., Zaplo, J**., Ney, H., *Statistical Classification Methods for Arabic News Articles*. Arabic Natural Language Processing, Workshop on the ACL'2001. Toulouse, France, July (2001).

[24] **El-Halees A.**, *Mining Arabic Association Rules for Text Classification* In the proceedings of the first international conference on Mathematical Sciences. Al-Azhar University of Gaza, Palestine, 15 -17 (2006).

[25] **Alaa M. El-Halees**, *Arabic Text Classification Using Maximum Entropy.* The Islamic University Journal (Series of Natural Studies and Engineering).Vol. 15, No.1, pp 157-167, 2007, ISSN 1726-6807,

[26] **Thorsten Joachims,** *"SVM^{light,} Support Vector Machine"* Cornell University, 2004. *http://svmlight.joachims.org/*

[27] **Lewis, D. D.** *"An evaluation of phrasal and clustered representations on a text categorization task". Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval*, pages 37–50, Kobenhavn, DK. ACM Press, New York, US.

[28] **Apte , Damerau, Weiss**, *Towards Language Independent Automated Learning of Text Categorization Models* . Research and Development in Information Retrieval P 23-30. (1994)

[29] **Ecole des Hiboux**, *"Site pour la langue et la culture Arabe", http://www.les-ziboux.rasama.org/*

[30] **الدكتور مسعد محمد زياد** ,"اللغة العربية", *http://www.drmosad.com/index.htm*

[31] **سعيد الأفغاني**_"كتاب الموجز في قواعد اللغة العربية", **دار الفكر - دمشق سورية** *http://www.islamguiden.com/arabi/index.htm*