

**République Algérienne Démocratique et Populaire**

**Ministère de l'Enseignement Supérieur  
et de la Recherche Scientifique**

**Université 08 Mai 45 Guelma  
Faculté des sciences de l'ingénieur**



**Département d'Informatique**

**Présenté pour l'obtention du diplôme de  
Magister en Informatique  
Option : Intelligence artificielle**

Titre du Mémoire

---

**Les machines à vecteurs supports dans la  
catégorisation de textes arabes**

---

Présenté par :  
**Djelailia Karim**

Devant le jury :

<b>Président :</b>	M.C. H. Séridi	Université de Guelma
<b>Rapporteur :</b>	M.C. H.F. Merouani	Université de Annaba
<b>Examineurs :</b>	M.C. Y. Tlili-Guiassa	Université de Annaba
	M.C. M.K Kholadi	Université de Constantine

## Remerciements

Je remercie chaleureusement M. Séridi Hamid maître de conférence à l'université de Guelma d'avoir accepté d'être le rapporteur de mon mémoire.

Je remercie Mme Tili-Guiassa Yamina maître de conférence à l'université de Annaba et M. Kholladi Mohamed Khireddine maître de conférence à l'université de Constantine d'avoir accepté d'être les examinateurs de mon travail.

Je suis très reconnaissant à Madame Merouani de m'avoir guidé tout au long de l'élaboration de mon travail et qui m'a encouragé dans les moments difficiles. Je remercie également Madame Tlili de son aide précieuse quant à l'acquisition du corpus d'apprentissage.

Je remercie encore une fois, M. Séridi Hamid pour son encouragement et ses grands efforts pour nous avoir assuré les conditions idéales pour l'accomplissement de notre année théorique.

Je remercie Professeur Kareem Darwish de l'Université du Caire pour nous avoir envoyé son programme al-stem et de nous avoir autorisé de le modifier et de l'utiliser.

Je tiens également à remercier mon ami et camarade Kelaiaia Abdessalem pour sa collaboration et son aide si précieuse.

Je tiens à remercier professeur Jean Marc Petit de l'INSA de Lyon de m'avoir guidé et d'avoir accepté de lire mon travail pour sa présentation à COSI 2007.

Je remercie également toute l'équipe COSI pour m'avoir permis une participation à la conférence d'Oran en Juin 2007. Je cite particulièrement Professeur Lakhdar Sais de l'Université d'Artois, Professeur Houari Nourine de l'Université de Clermont Ferrand et Professeur Benamrane Nacéra de l'Université d'Oran.

Je ne dois pas oublier de remercier tous ceux qui m'ont aidé et soutenu durant ces trois dernières années : Ma petite et grande famille et mes amis.

# Dédicace

A la mémoire de mon père

A ma mère

A ma petite famille (Nora, Rawiya et Fady)

A mes frères et sœurs

A mes amis

Il est peu de réussites faciles, et d'échecs définitifs.

Marcel Proust

# Sommaire

## Sommaire

### Résumé

### Table des figures

### Liste des tableaux

### Abréviation et acronymes

<b>Introduction générale</b>	1
<b>Partie I : Etat de l'art</b>	3
<b>Chapitre 1 : La catégorisation de textes</b>	4
Introduction	5
1-1 Définition de la catégorisation de textes	5
1-2 Processus de catégorisation	5
1-3 Représentation de textes	7
1-4 Caractéristiques de la catégorisation de textes	8
1-4.1 Influence de la nature des textes	8
1-4.2 Avantage de l'apprentissage	8
1-4.3 Les types de classification	8
1-5 Evaluation d'un système de classification	9
1-5.1 Classification bi-classe : Précision, rappel et mesure $F_\beta$	9
1-5.2 Classification multi-classe : Précision, rappel et mesure $F_\beta$	11
1-5.3 Micro et macro mesures	11
1-6 Difficultés de la catégorisation de textes	11
1-7 Algorithmes d'apprentissage utilisés dans la catégorisation de textes	11
1-8 Application de la catégorisation de textes	12
1-9 Lien avec la recherche documentaire	12
1-10 Corpus d'évaluation	13
1-11 Conclusion	13
<b>Chapitre 2 : La représentation de textes</b>	14
Introduction	15
2-1 Choix des termes	15
2-1-1 Les termes sont exactement les mots contenus dans les textes	15
2-1-2 Représentation des textes par des phrases	16
2-1-3 Représentation des textes par des racines lexicales « stemmes et lemmes »	16
2-2 Codage des termes	17
2-2-1 Représentation en vecteur binaire	17
2-2-2 Vecteur fréquentiel	18
2-2-3 Représentation TF-IDF	19
2-3 Réduction de dimension	20

2-3-1	Réduction locale de dimension	21
2-3-2	Réduction globale de dimension	21
2-3-3	Sélection de termes	21
	Le seuil de fréquence d'un terme	21
	Le gain d'information ( « <i>information gain</i> » )	22
	L'information mutuelle ( « <i>mutual information</i> » )	22
	La statistique du $\chi^2$	22
	La force du terme ( « <i>term strength</i> » )	22
2-3-4	Extraction de termes	23
	L'indexation par la sémantique latente (LSI)	23
	Principe	23
	Exemple	25
2-4	Quel est le meilleur nombre de termes à conserver ?	27
2-5	Conclusion	27
<b>Chapitre 3 : spécificités de la langue Arabe et travaux sur la catégorisation de textes Arabes</b>		<b>28</b>
Introduction		29
3.1	Particularité de la langue arabe	30
3.1.1	Morphologie arabe	31
3.1.2	Structure d'un mot	32
3.1.3	Catégories des mots	32
3.1.3.1	Le verbe	32
3.1.3.2	Les noms	33
3.1.3.3	Les particules	34
3.2	Problèmes du traitement automatique de l'arabe	34
3.2.1	Détection de racine	34
3-2.2	l'agglutination	35
3.3	Travaux sur la catégorisation de textes Arabes	36
3.4	Conclusion	37
<b>Chapitre 4 : Techniques utilisées dans la catégorisation de textes</b>		<b>38</b>
Introduction		39
4-1	Algorithmes d'apprentissage et données textuelles	39
4-2	Etude des classifieurs utilisés en catégorisation de textes	40
4-2-1	Le modèle de Rocchio	41
4-2-2	Le classifieur Bayésien naïf	42
4-2-3	Les machines à vecteurs supports	43
	1- Cas linéairement séparable	44
	2- Cas linéairement non séparable	49
4-2	Conclusion : Choix du meilleur algorithme de classification de texte	51
<b>Partie II : Expérimentation</b>		<b>53</b>
<b>Chapitre 5 : corpus utilisé et processus de prétraitement proposé</b>		<b>54</b>

5-1 Le corpus	55
5-2 Processus d'expérimentation	57
5-3 Le prétraitement	
5-3.1 Obtention de textes bruts	58
5-3.2 Tokenisation et indexation	58
5-3.3 La normalisation	59
5-3.4 Translittération	59
5-3.5 Radicalisation ou stemming	61
5-3.6 Elimination des mots outils	62
5-4 Outil pour le prétraitement développé	62
<b>Chapitre 6 : expérimentations</b>	<b>64</b>
Introduction	65
6-1 Présentation de l'environnement utilisé	65
6-1.1 Obtention de la représentation TF-IDF	66
6-1.2 Construction du classifieur SVM	66
6-1.3 Réduction de dimension	67
6-1.4 Chargement des résultats	67
6-1.5 Construction du classifieur Naïve Bayes	68
6-2 Expérimentations réalisées	68
<b>Chapitre 7 : Résultats et discussions</b>	<b>70</b>
7-1 Résultats	71
7-2 Différentes mesures	73
7-2.1 Précision	73
7-2.2 Rappel	74
7-2.3 F-mesure	75
7-2.4 Quelques remarques sur le stemming et la réduction de dimension	76
7-2.5 Comparaison entre Bayes et SVM	76
7-3 Discussion générale	77
<b>Conclusion et perspectives</b>	<b>78</b>
<b>Références et bibliographie</b>	<b>80</b>
<b>Annexe</b>	<b>89</b>

## **Résumé :**

Notre mémoire traite la problématique de la catégorisation de texte en langue arabe, une approche de classification supervisée. La base d'apprentissage étant un corpus en langue arabe de documents étiquetés. La représentation utilisée est la représentation vectorielle, avec la technique TF-IDF. Nous évoquons à travers ce mémoire, l'influence de la sélection d'attributs et la langue du corpus d'entraînement dans la qualité des résultats du classifieur. Les SVM (acronyme de Support vector machine) est la méthode de classification que nous utilisons dans notre expérimentation. Le choix de la langue Arabe est motivé par la rareté des travaux menée dans ce domaine pour cette langue. Vu ses particularités morphosyntaxiques (langue fortement dérivationnelle, à caractère flexionnel et agglutinante). Nous visons à confirmer ou infirmer que la qualité des résultats obtenus pour d'autres langues avec les SVM et basés sur les techniques de radicalisation des termes (stemming), pour réduire la dimension de l'espace de représentation (problème inhérent à la technique de représentation en sacs de mots –bag of words-), sont ou ne sont pas liés à la nature de la langue du corpus. Il est donc, question de distinguer entre les résultats obtenus avec un prétraitement rigoureux de ceux obtenus avec un prétraitement rudimentaire consistant à une simple tokenisation.

**Mots-clés:** Catégorisation de textes, corpus, stemming, TF-IDF, SVM

## **Abstract :**

In our work we use an approach for supervised classification to treat the problem of categorization of Arabic texts. The learning is based on a corpus of Arabic-language labelled documents. The representation used is the vector representation, with the TF-IDF technique. Through this work, we talk about the influence of the selection of attributes and language of the training corpus in the quality of the results of the classifier. The SVM (an acronym for Support Vector Machine) is the method of classification that we use in our experimentation. The choice of Arabic language is motivated by the scarcity of works conducted in this area for this language. Given its special morpho-syntactic nature (strong derivational, flexional and agglutinative language), we aim to confirm or deny that the quality of results for other languages with SVM based on the techniques of radicalization of words, to reduce the dimension of the space representation (a problem inherent to the bags of words representation), is or is not related to the nature of the language of the training corpus.

Therefore, there is an issue of distinguishing between the results obtained with a rigorous pre-treatment of those obtained with a rudimentary pre-treatment consisting of a simple tokenization.

**Key words:** text categorization, corpus, stemming, TF-IDF, SVM



## ملخص:

يعالج بحثنا هذا مشكلة تصنيف النصوص العربية ، منهج للتصنيف المراقب. إن التدريب يركز على مجموعة من النصوص المعلمة باللغة العربية. التمثيل المستخدم هو التمثيل الشعاعي، مع تقنية تواتر الكلمات-عكس تواتر الوثائق. نتحدث من خلال هذا العمل عن تأثير اختيار الصفات ولغة مجموعة التدريب في نوعية نتائج المصنف.

آلات أشعة الدعم هي طريقة التصنيف التي نستخدمها في تجاربنا. اختيار اللغة العربية هو بدافع ندرة الأعمال التي أجريت في هذا المجال لهذه اللغة. نظرا للخاصية النحوية والصرفية (قوة اشتقاقية وإعرابية و التصاق الكلمات ). نحن نهدف إلى تأكيد أو نفي بأن نوعية النتائج المحصل عليها في لغات أخرى مع آلات أشعة الدعم استنادا إلى تقنيات اشتقاق جذور الكلمات لتقليص أبعاد فضاء التمثيل (مشكلة متصلة بأسلوب التمثيل بكيس الكلمات)، ترتبط أو لا بطبيعة لغة مجموعة التدريب.

سنميز بين النتائج التي تم التوصل إليها مع صرامة المعالجة المسبقة من تلك التي نتحصل عليها مع معالجة مسبقة بداءيه تتكون من تكسير النصوص إلى كلمات فقط.

الكلمات الدالة: تصنيف النصوص، مجموعة نصوص، اشتقاق جذور الكلمات، تواتر الكلمات-عكس تواتر الوثائق، آلات أشعة الدعم.

## Table des figures

Fig 1.1 : Processus de catégorisation de textes	6
Fig 2.1 : Illustration de la loi de Zipf	19
Fig 4.1 Schéma illustratif du principe de construction d'une SVM	44
Fig 4.2 : Illustration de l'hyperplan séparateur et de la marge	45
Fig 4.3 : Transformation des données de l'espace initial vers un espace de plus grande dimension pour que les données deviennent linéairement séparables	49
Fig 5.1 : Processus d'expérimentation	58
Fig 7.1 : Comparaison de la précision sur le corpus radicalisé et brut avec ou sans réduction de dimension	73
Fig 7.2 : Comparaison du rappel sur le corpus radicalisé et brut avec ou sans réduction de dimension	74
Fig 7.3 : Comparaison de la F-mesure sur le corpus radicalisé et brut avec ou sans réduction de dimension	75

## Liste des tableaux

Tableau 1.1 : Matrice de contingence bi-classe	9
Tableau 1.2 : Matrice de contingence bi-classe -exemple	10
Tableau 2.1 exemple de représentation en vecteurs binaires	17
Tableau 2.2 exemple de représentation en vecteurs fréquentiels	18
Tableau 2.3 exemple de représentation en vecteurs fréquentiels normalisés	18
Tableau 2.4 exemple de représentation TF-IDF	20
Tableau 2.5 : Matrice $A^T$ de l'exemple pour la LSI	26
Tableau 2.6 : Matrice $A_k^T$ de l'exemple pour la LSI	26
Tableau 3.1 : alphabet arabe	30
Tableau 3.2 : Variation de la lettre ج jim	30
Tableau 3.3 : Ambiguïté causée par l'absence de voyelles pour les mots سلم et أكل	31
Tableau 3.4 : Différentes dérivations des racines منح et فتح	31
Tableau 3.5 : Les affixes les plus fréquents en langue arabe	34
Tableau 3.6 : Différentes radicalisations du terme ايمان	35
Tableau 3.7 : Exemple de déclinaison du verbe irrégulier قال dire	35
Tableau 3.8 : Exemple de segmentation du mot المهم	36
Tableau 5.1 : Les corpus disponibles pour l'arabe	56
Tableau 5.2 : Caractéristiques de CCA	57
Tableau 5.3 : exemple de tokenisation	59
Tableau 5.4 : système de translittération proposé par Tim	61

Buckwalter	
Tableau 5.5. : Les affixes considérés dans le stemming (Light 10 et Al-stem)	62
Tableau 6.1 : les plugins disponibles pour RapidMiner	65
Tableau 6.2 : Les classes utilisées dans les expérimentations	69
Tableau 7.1 : Tableau récapitulatif des différentes expérimentations	76
Tableau 7.2 : Comparaison des résultats obtenus entre les SVM et Naive Bayes	76

## **Abréviation et acronymes**

ACM	: Association for Computing Machinery
AFP	: Agence France-Presse
ASCII	: American Standard Code for Information Interchange
CCA	: Corpus of Contemporary Arabic
CICKM	: International Conference on Intellectual Capital, Knowledge Management
CRF	: Conditional Random Field
ELRA	: European Language Resources Association
IA	: Intelligence Artificielle
LDC	: Language Data Consortium (Pennsylvania)
LSI	: Latent Semantic Indexing
NLP	: Natural Language Processing
PLSA	: Probabilistic Latent Semantic Analysis
RI	: Recherche d'Information ou IR pour Information Retrieval
RD	: Recherche Documentaire
SIGIR	: Special Interest Group on Information Retrieval
SVM	: Support Vector Machines
TAFL	: Teaching Arabic as a Foreign Language
TALN	: Traitement Automatique du Langage Naturel
TC	: Text Categorization
TF-IDF	: Term Frequency Inverse Document Frequency
TREC	: Text REtrieval Conference
XML	: eXtensible Markup Language
YALE	: Yet Another Learning Environment

# Introduction

Le développement des moyens de communication et en particulier Internet a contribué, aujourd'hui à l'émergence de la circulation d'une masse importante d'informations sous forme textuelle. On estime à près de 80 % cette masse [A. Lehman et P. Bouvet, 2001]. En parallèle les entreprises et les particuliers se trouvent souvent confrontés à l'exploitation d'informations provenant de documents textuels. Cette masse gigantesque d'information serait sans intérêt si notre capacité à y accéder n'augmentait pas efficacement.

Comprendre un texte, voilà le défi que se pose aujourd'hui l'IA moderne. L'extrême variété d'analyses d'un texte et la complexité de la nature de l'information qui y est disséminée a poussé les chercheurs à limiter leurs ambitions et à se focaliser actuellement sur des tâches moins ambitieuses leur permettant de construire les premières couches d'un système de compréhension de textes. Il s'agit alors de tâches inhérentes à la classification de textes, de résumés automatiques, de visualisation du contenu d'une base documentaire et d'extraction d'information précise.

Le but visé par ce mémoire étant d'étayer les différentes méthodes de recherche thématique. Ceci est basé essentiellement sur la classification d'un texte (text categorization ou TC). La première question que l'on se pose en présence d'un texte est « Quel est le thème abordé par ce texte ? » ou plus généralement en ayant sous la main une masse considérable de documents, la première tâche à effectuer pour faciliter la recherche d'information est de l'organiser en catégories pour pouvoir extraire les documents ayant un rapport avec la thématique visée. Bien évidemment, on peut faire ce travail manuellement. Mais cette tâche devient impossible dès lors que la masse de textes que nous détenons se compte par milliers. On ressent alors la nécessité de disposer de moyens pour catégoriser automatiquement ces documents. C'est le but poursuivi par la classification automatique de documents. On pourrait imaginer les différentes applications qui l'utiliseraient. On peut citer à titre d'exemple la catégorisation de courrier électronique pour une entreprise en vue de dégager les besoins de ses clients. L'organisation d'une base documentaire dans un projet de recherche. Le tri de pages Web en vue de leur consultation pour différentes raisons, le tri du courrier électronique en Spams et e-mails et la liste ne s'arrêterait pas là.

Pour extraire de l'information à partir d'un texte, les premiers chercheurs se sont heurtés à la difficulté de procéder à une analyse en profondeur vu la complexité du langage humain. Les premiers problèmes étant le contexte, la synonymie et la polysémie. On se retourne alors à une analyse de surface en vue de dégager des informations utiles pour des buts bien précis.

Dans ce contexte, on se base sur une représentation de textes plus souple se prêtant à un traitement automatique. Généralement c'est la représentation vectorielle où dite en « sac de mots » qui est utilisée car elle convient bien aux différents algorithmes de calcul. Les attributs étant les mots présents dans les textes.

La littérature se référant au domaine de la catégorisation automatique de textes nous fournit une masse gigantesque de travaux ayant traité cette problématique. Mais rares sont les travaux qui se sont focalisés sur l'importance des attributs sélectionnés dans la qualité des résultats

obtenus ou plus encore la contribution de la méthode de sélection des attributs et la langue utilisée dans les documents dans les résultats de cette opération.

Tous ces éléments nous ont motivé à évaluer cet impact en utilisant l'algorithme le plus communément utilisés dans la catégorisation de textes, en l'occurrence les machines à vecteurs supports (son acronyme anglo-saxon SVM : Support vector machine).

## **Organisation du document**

Le document est organisé en deux parties :

Dans la première partie nous exposerons l'état de l'art sur la catégorisation de textes. Elle est composée de quatre (04) chapitres. Dans le premier chapitre nous aborderons la catégorisation de textes en évoquant la problématique de la recherche documentaire et la recherche d'information en général. En second chapitre nous parlerons de la représentation des documents et l'extraction de termes et nous établirons les différents codages possibles de cette représentation. Au niveau du chapitre 3 nous discuterons de la nature des textes écrits en Arabe et des différences fondamentales des textes écrits en Arabe par rapport aux langues latines et l'on présentera les différents outils d'extraction de termes appliqués à la langue Arabe. Nous traiterons dans le quatrième chapitre des différents algorithmes de catégorisation de textes et l'on comparera trois d'entre eux. Nous mettrons un grand bémol sur les SVM.

Dans la seconde partie du document nous détaillerons l'expérimentation que nous avons menée sur un corpus en langue Arabe. Elle est composée de trois (03) chapitres. En cinquième chapitre, nous présenterons le corpus utilisé et le prétraitement effectué et l'on parlera au sixième chapitre des moyens logistiques que nous avons utilisés et nous terminerons au septième chapitre par les résultats obtenus avec une discussion sur leur qualité et enfin, nous conclurons ce travail et nous évoquerons nos perspectives.

# **Partie I**

## **Etat de l'art**

# **Chapitre 1**

## **La catégorisation de textes**

## Introduction

Le problème actuel que se pose tout chercheur d'information n'est plus comment accéder à l'information mais comment la trouver ? La réponse à cette question est du ressort de toute la discipline de la recherche d'information (RI) dont la catégorisation de texte est l'un des outils.

### 1-12 Définition de la catégorisation de textes

Nous nous plaçons dans un contexte où un ensemble de catégories ou étiquettes sont définies à l'avance. C'est-à-dire, définies par les usagers. Dans ce contexte on parle de catégorisation de textes. On parle de clustering (segmentation) dans le cas où les catégories ne sont pas définies à l'avance et c'est au système de déterminer les groupes de documents qui sont liés. Ceci n'est pas l'objet de cette étude.

La catégorisation de textes consiste en la recherche d'une relation fonctionnelle entre un ensemble de textes noté  $D$  et un ensemble de catégories noté  $C$  [Sebastiani, 2002]. Ceci est basé essentiellement sur un algorithme d'apprentissage.

Formellement, un processus de catégorisation se définit comme une fonction  $\Phi$  telle que :

$$\Phi : D \times C \rightarrow \{\text{Vrai, Faux}\} \quad (1.1)$$

C'est-à-dire, étant donné un couple  $(d_i, c_j)$ , on cherche une valeur booléenne qui prend la valeur vraie si  $d_i \in c_j$  et fausse sinon.

On suppose que les étiquettes sont des valeurs symboliques et qu'aucune information supplémentaire concernant leur signification n'est considérée. On n'utilise alors que les informations endogènes afin de pouvoir catégoriser un texte. Les informations exogènes telles que les dates de publication, le type de documents et les sources de publication ne sont pas prises en considération.

### 1-13 Processus de catégorisation

Le processus de catégorisation est un système (cf. figure 1.1) qui reçoit en entrée un texte et en sortie, lui associe une ou plusieurs catégories. Ceci est effectué en respectant un ensemble d'étapes. Ces étapes concernent la représentation des textes, le choix de l'algorithme d'apprentissage, et l'évaluation des résultats en vue de prévoir le degré de généralisation du classifieur.

Dans ce processus, deux phases sont à distinguer à savoir, l'apprentissage et la classification :



1- **L'apprentissage** : Le but de cette étape est d'aboutir à un modèle de généralisation. Il consiste en :

- un ensemble de textes d'apprentissage étiquetés,
- extraction des termes les plus pertinents à partir des textes d'apprentissage,
- mise en relation textes/termes (représentation vectorielles des documents). (d'autres représentations sont possibles nous en discuterons dans le chapitre suivant),
- application de l'algorithme d'apprentissage en vue d'aboutir à la fonction  $\Phi$  de catégorisation.

2- **La classification** : Cette phase qu'on pourrait qualifier de phase d'exploitation consiste en :

- la recherche des termes du document à classer,
- l'application de la fonction  $\Phi$  de catégorisation.

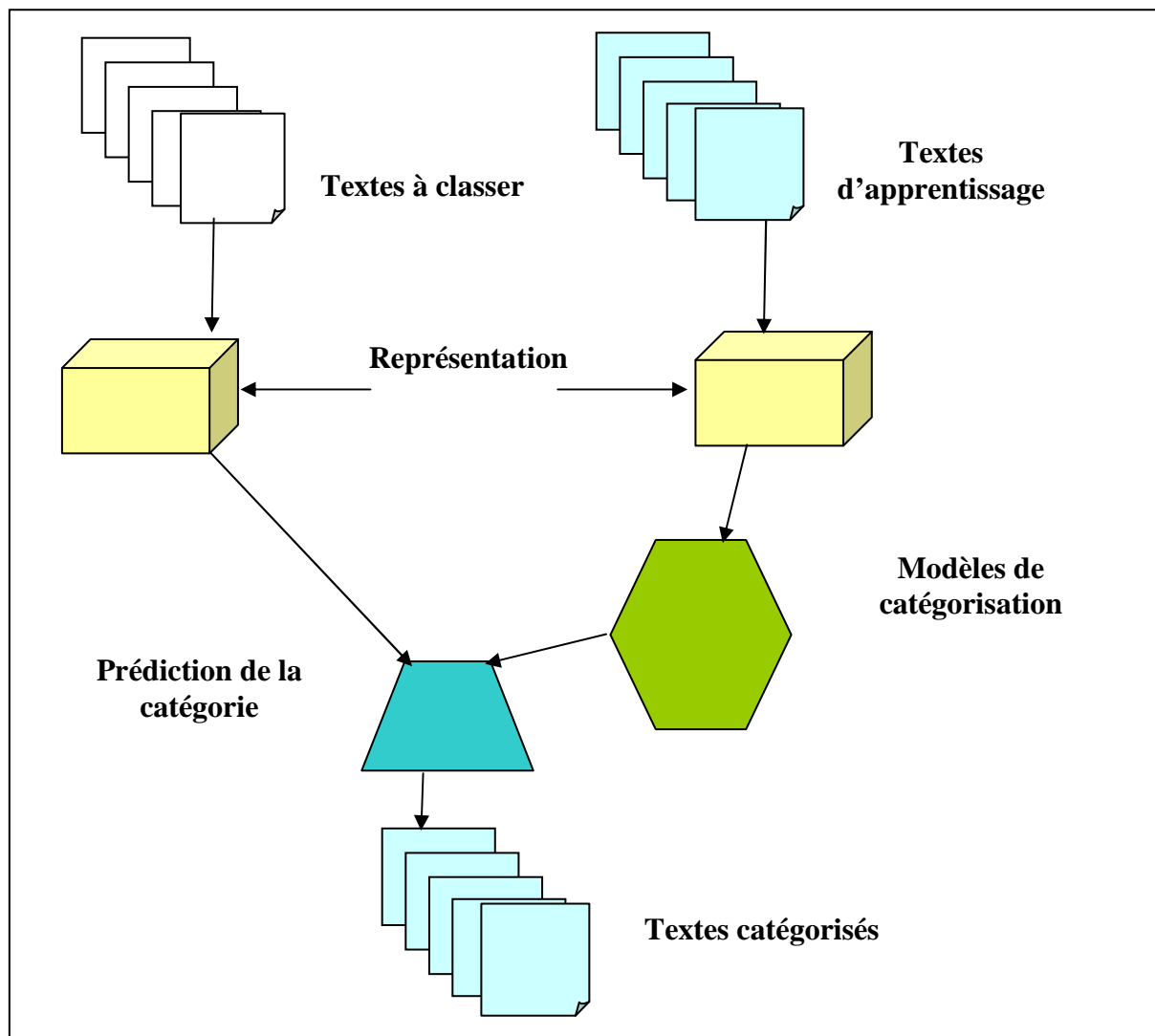


Fig 1.1 : Processus de catégorisation de textes

## 1-14 Représentation de textes

Comme pour l'image, le son etc... les textes à catégoriser doivent être codés que ce soit lors de l'apprentissage ou lors de la phase de classification car la machine ne sait pas encore traiter des données qui ne sont pas structurées. En traitement numérique ceci est réalisé en produisant un tableau croisé « individus-variables » :

- l'individu est un texte (document) noté  $d_j$  étiqueté lors de la phase d'apprentissage et à classer lors de la classification,
- les variables sont les termes notés  $t_k$  extraits des documents.

Le contenu du tableau (éléments  $w_{kj}$ ) point d'intersection d'un texte  $j$  et d'un terme  $k$  représente le poids de ce terme dans le document.

Le plus important en catégorisation de textes est le choix des descripteurs (termes).

Plusieurs méthodes ont été utilisées pour le choix des descripteurs et des poids associés à ceux-ci :

[Yang, 1997] et [Salton et Mc Gill, 1983, Aas et Eikvil, 1999] utilisent, à titre d'exemples, les mots comme descripteurs, tandis que d'autres préfèrent utiliser les lemmes (racines lexicales) [Sahami, 1999] ; ou encore des *stemmes* (la suppression d'affixes) [de Loupy, 2001]. Il existe une autre approche de la représentation des textes : les **n-grammes** : les séquences de  $n$  caractères [Cavnar et Trenkle, 1994]. Aussi d'autres ont essayé la représentation par des phrases [Fuhr et Buckley, 1991, Schütze et al., 1995, Tzeras et Hartmann, 1993]

L'exploitation des documents textuels dans l'espace des termes (mots, lemmes, stemmes, n-grammes, phrases, etc.) n'est pas possible sans réduction préalable de la dimension de cet espace, car celle-ci est en générale trop grande (de l'ordre des milliers). L'objectif des techniques de réduction est de déterminer un espace restreint de descripteurs conservant au mieux l'information originelle pour différencier les étiquettes de classement. La sélection de termes peut être effectuée soit localement, pour chaque catégorie  $c_i$ , un ensemble de termes  $T_i'$  avec  $|T_i'| \ll |T_i|$  est choisi pour représenter  $c_i$  [Apté et al., 1994, Lewis et Ringuette, 1994], soit globalement, un ensemble de termes  $T'$  avec  $|T'| \ll |T|$  est choisi pour représenter la totalité des classes  $C$  [Yang et Pedersen, 1997, Mladenić, 1998, Caropreso et al., 2001].

Le choix de descripteurs et la réduction de dimension seront traités en détail au niveau du chapitre deux, consacré aux méthodes de représentation.

# 1-15 Caractéristiques de la catégorisation de textes

## 1-4.1 Influence de la nature des textes

La nature des textes à classer influence grandement la qualité des résultats d'un classifieur. En effet, les textes informatifs telles que les dépêches de presse sont écrits dans un style direct et précis et les résultats basés sur l'extraction des termes sont plus probants alors que les textes littéraires sont écrits dans un style plus imagé ce qui rend la tâche de classification plus difficile.

## 1-4.2 Avantage de l'apprentissage

Les premiers travaux sur la catégorisation de textes se sont focalisés sur la recherche de règles de production (systèmes experts) de type : **Si Antécédent alors Conclusion** où l'antécédent portait sur la présence ou l'absence de certains mots et la conclusion est la classe de document. Ces travaux se sont heurtés à la complexité de la tâche d'écriture de règles surtout pour des corpus importants de plus, l'impossibilité d'exporter le système sur d'autres corpus. L'avantage de l'apprentissage est que c'est le système lui-même qui apprend à classer les documents ce qui permettra de le généraliser sur d'autres corpus sans beaucoup de changements.

## 1-4.3 Les types de classification

- **Classification bi-classe** : Ce type de classification correspond au contexte de filtrage de documents c'est-à-dire prendre la décision si un document est ou non pertinent pour une classe donnée.
- **Classification multi-classe disjointe** : dans ce contexte, le nombre de classe est supérieur à 1 et le système doit répondre à la question « à quelle classe (au singulier) appartient le document ? ».
- **Classification multi-classe** : A l'opposé, de la précédente, un document peut appartenir à plus d'une classe.  
En général, on construit un classifieur bi-classe pour chaque classe et le document est soumis à chacun des classifieurs en vue de sa catégorisation. Plusieurs classifieurs peuvent affecter le document à la classe pour laquelle ils sont conçus ce qui permettra une classification multi-classe. C'est le cas notamment des machines à vecteurs supports.
- **Le ranking** : Dans ce cadre, au lieu que le système prenne une décision « dure », il affecte un score de pertinence au document pour une classe donnée. Cette fonction peut être formalisée comme suit :  $SC : D \times C \rightarrow [0,1]$ . Elle est surtout utilisée dans le cas d'accès aux pages Web pour trier les documents par ordre de pertinence.

La plus part des systèmes, surtout probabilistes, permettent d'obtenir la valeur de cette fonction. Cependant la définition d'un seuil de pertinence  $S_p$  permettrait de prendre cette décision. Il est formalisé comme suit :

$$\begin{cases} \text{Si } SC(d,c) > S_p & \text{alors } \Phi(d,c) = \text{vrai} \\ \text{sin on } & \Phi(d,c) = \text{Faux} \end{cases} \quad (1.2)$$

## 1-16 Evaluation d'un système de classification

### 1-5.1 Classification bi-classe : Précision, rappel et mesure $F_\beta$

Nous utilisons un corpus de test pour lequel nous connaissons la vraie catégorie (pertinent ou pas) d'un document  $d$  par rapport à une classe  $c$

Nous construisons alors un tableau de contingence (Tableau 1.1) qui contient les mesures suivantes :

VP : Le nombre de documents pertinents classés correctement

VN : Le nombre de documents non pertinents classés correctement

FP : Le nombre de documents pertinents mal classés

FN : Le nombre de documents non pertinents mal classés

Classe trouvée \ Classe réelle	Pertinents	Non pertinents
	Pertinents	<b>VP</b>
Non pertinents	FP	<b>VN</b>

Tableau 1.1 : Matrice de contingence bi-classe où les valeurs en gras représentent les documents classés correctement

- **Rappel** : Il est défini comme étant la proportion de documents pertinents correctement classés

Formellement il est calculé comme suit :

$$\rho = VP / (VP + FP) \quad (1.3)$$

- **Précision** : Elle est définie comme étant la proportion de documents réellement pertinents parmi ceux classés par le système comme étant pertinents.

Formellement elle est calculée comme suit :

$$\pi = VP / (VP + FN) \quad (1.4)$$

Si la précision est faible, les documents retournés ne sont pas tous pertinents pour la requête de l'utilisateur. Par contre si le rappel est faible, les documents qu'on souhaite voir ne sont pas tous retournés.

L'idéal est d'avoir une précision et un rappel de 1. Ces deux exigences sont souvent contradictoires et la force de l'un est au détriment de l'autre.

Dans cet ordre, on peut aussi définir, la notion de bruit (B) et de silence (S), qui sont respectivement les notions complémentaires de précision et de rappel.  $B = 1 - \pi$ ;  
 $S = 1 - \rho$ .

- **Mesure  $F_\beta$**  : Pour synthétiser la double information portée par la précision et le rappel, plusieurs mesures ont été développées. Nous ne retiendrons ici que la mesure  $F_\beta$  décrite dans [Van Rijsbergen, 1979] qui est une mesure courante de la littérature sur la classification de documents. On définit la mesure  $F_\beta$  comme la moyenne harmonique entre le rappel et la précision :

$$F_\beta = \frac{(\beta^2 + 1) * \pi * \rho}{\beta^2 * (\pi + \rho)} \quad (1.5)$$

La valeur du paramètre  $\beta$  permet d'accorder plus ou moins de poids à la précision d'un système. Habituellement, la valeur de  $\beta$  est fixée à 1 et la mesure est ainsi notée  $F_1$ .

**Illustration par un exemple** : Soit La classe C contenant 50 documents et la classe non C contenant 50 documents. Un système de classification donné qui donne les résultats qui figurent dans le tableau 1.2 aurait les mesures suivantes :

$$\rho = VP / (VP + FP) = 35 / (35 + 15) = 0,70$$

$$\pi = VP / (VP + FN) = 35 / (35 + 25) = 0,58$$

$$F_1 = 2\rho\pi / (\rho + \pi) = 0,64$$

Classe trouvée \ Classe réelle	Pertinents	Non pertinents
	Pertinents	<b>35</b>
Non pertinents	15	<b>25</b>

Tableau 1.2 : Matrice de contingence bi-classe -exemple

## 1-5.2 Classification multi-classe: Précision, rappel et mesure $F_\beta$

Nous pouvons généraliser la notion de précision et de rappel, vue précédemment, dans le cas multi-classe en calculant ces paramètres pour chacune des classes en opérant par la technique un contre tous ; c'est-à-dire pour une classe donnée, le restant des classes ne forme qu'une seule classe ce qui ramène à un problème de classification bi-classe. Les paramètres du

classifieurs seront obtenus par le calcul de la moyenne des différents paramètres obtenus pour chacune des classes.

### **1-5.3 Micro et macro mesures**

Cependant, pour évaluer la performance d'un classifieur, on procède par deux méthodes appelées macro et micro mesures (macro and micro averaging). La performance d'une macro mesure c'est la moyenne des performances des classifieurs pour chacune des catégories. La micro mesure est le résultat du calcul des performances à partir de la somme des VP, VN, FP, FN.

La micro mesure privilégie donc le document c'est-à-dire que tous les documents ont la même importance alors que la macro mesure privilégie la catégorie, c'est-à-dire que toutes les catégories sont d'une même importance.

## **1-6 Difficultés de la catégorisation de textes**

Le langage naturel est par nature un langage non univoque (on peut exprimer les mêmes idées avec des termes différents). On parle alors de problème de synonymie. On peut aussi utiliser les mêmes termes pour dire des choses différentes. C'est le problème de la polysémie. En plus, les idées peuvent être exprimées de façon implicite. La grande dimensionnalité des descripteurs (notion que nous développerons au niveau de la représentation de textes) est aussi une autre difficulté qui rend caduque la majorité des algorithmes d'apprentissage. Enfin la catégorie à laquelle appartient le texte est définie par un expert humain ce qui lui confère un caractère subjectif (Un autre expert peut prendre une décision différente).

Une classification opérée pour un corpus peut devenir caduque lorsque le nombre de documents augmente. On doit revoir les classes et les critères de classification.

## **1-7 Algorithmes d'apprentissage utilisés dans la catégorisation de textes**

Comme, il a été dit précédemment, la grande dimensionnalité des données issues du prétraitement d'un corpus rend inopérants certains algorithmes d'apprentissage. C'est pourquoi, seuls certains algorithmes sont utilisés dans ce domaine. Parmi les méthodes d'apprentissage les plus utilisées, on cite, l'analyse factorielle discriminante [Lebart et Salem, 1994], la régression logistique [Hull, 1994], les réseaux de neurones [Wiener et al., 1995, Schütze et al., 1995, Stricker, 2000], les plus proches voisins [Yang et Chute, 1994, Yang et Liu, 1999], les arbres de décision [Lewis et Ringuette, 1994, Apté et al., 1994], les réseaux bayésiens [Borko et Bernick, 1964, Lewis, 1998, Androutsopoulos et al., 2000, Chai et al., 2002,], les machines à vecteurs supports [Joachims, 1998, Joachims, 1999, Joachims, 2000, Dumais et al., 1998, He et al., 2000] et, plus récemment, les méthodes dites de *boosting*

[Schapire et al., 1998, Iyer et al., 2000, Schapire et Singer, 2000, Escudero et al., 2000, Kim et al., 2000, Carreras et Márquez, 2001, Liu et al., 2002].

Ces classifieurs se différencient selon leur mode de construction (les classifieurs sont-ils construits manuellement, ou bien automatiquement par induction à partir des données ?) et selon leurs caractéristiques, (le modèle appris est-il *compréhensible*, ou bien s'agit-il d'une *fonction numérique* calculée à partir de données servant d'exemples ?).

Généralement, le choix du classifieur est 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 ou les classifieurs à base de règles.

On explicitera les méthodes les plus couramment utilisées au niveau du chapitre quatre, consacré aux algorithmes d'apprentissage.

## 1-8 Application de la catégorisation de textes

On peut citer, l'identification de la langue [Cavnar et Trenkle, 1994], la reconnaissance des auteurs [Forsyth, 1999, Teytaud et Jalam, 2001] et la catégorisation de documents multimédia [Sable et Hatzivassiloglou, 2000], et bien d'autres.

La catégorisation de textes peut être une fin en soi, c'est le domaine de l'indexation où un texte est décrit par un ensemble d'index.

La catégorisation de texte peut être un support pour d'autres applications. On peut citer :

- **le filtrage** : Il consiste à déterminer si un document est ou non pertinent pour une catégorie donnée.  
Exemple : détection de courriers non désirés (Spams) [Androutsopoulos et al., 2000, [Cohen, 1996](#)].
- **le routage** : Il consiste à affecter un texte à une catégorie parmi n catégories.  
Exemple : Un outil de routage peut déterminer à quelle destination envoyer un document selon le centre d'intérêt de chaque individu inscrit dans sa base de données [Liddy et al., 1994].

## 1-9 Lien avec la recherche documentaire

En recherche documentaire (RD), une requête est lancée en vue d'obtenir les documents les plus pertinents, ce qui revient à catégoriser les textes en textes pertinents et non pertinents pour une classe donnée.

En RD on procède comme suit [Lewis 1992b]:

- indexer les documents : c'est la représentation de textes en vue de leur exploitation,

- Formuler les requêtes par soit :

- a- un descripteur de thème : ex. « astrophysique »,
- b- une requête construite à l'aide des mots du langage courant en utilisant des opérateurs logique, de proximité, de troncature : ex. « (Astronomie et trous noirs\*) ou (corps près sombre\*) »,
- c- une expression en langage naturel : ex. « tension artérielle très élevée »,
- d- un document entier, utilisé comme exemple du sujet sur lequel on veut obtenir d'autres informations,
- e- un graphe de concepts : les concepts, représentés par des termes, peuvent être liés par des relations sémantiques de natures diverses (réseaux sémantiques).

- comparaison entre la requête et les documents utilisant une fonction de similarité,
- feedback : L'utilisateur reformule sa requête dans le cas où les documents ne satisfont pas ses besoins.

La catégorisation de texte est étroitement liée à la recherche documentaire. La recherche documentaire intervient en trois phases du cycle de vie d'un classifieur [Sebastiani, 2002]:

- a- lors de l'*indexation* des textes,
- b- lors du choix d'une *méthode d'appariement* entre un texte étiqueté et un autre texte, à étiqueter,
- c- lors de l'évaluation d'un classifieur.

## 1-10 Corpus d'évaluation

En catégorisation de textes, les données d'entraînement et de test sont disponibles en vue d'évaluer et de comparer entre elles les méthodes proposées par les chercheurs. L'agence de presse Reuters a mis à la disposition des chercheurs dans ce domaine à partir de 1989, des collections d'articles normalisés. Plusieurs versions ont été proposées et c'est le corpus Reuters-21578 qui est le plus utilisé de nos jours.

Dans l'étude que nous avons menée nous avons utilisé un corpus en langue Arabe. Il sera présenté au niveau du chapitre traitant du corpus utilisé pour l'évaluation de l'algorithme choisi.

## 1-11 Conclusion

Aux termes de ce chapitre, nous croyons avoir fixé les idées sur la problématique de catégorisation de texte. Nous avons soulevé la méthode générale et le prétraitement des données textuelles en vue de leur utilisation dans la catégorisation. Nous avons noté que le choix de la méthode d'apprentissage est primordial. Nous avons exposé les applications utilisant la catégorisation de texte. Nous avons aussi, évoqué les difficultés rencontrées surtout celles générées par la polysémie et la synonymie. Nous avons démontré le lien entre la catégorisation de texte et la recherche documentaire.

Nous introduisons au chapitre suivant les différents modèles de représentation de textes.



# **Chapitre 2**

## **La représentation de textes**

## Introduction

Les ordinateurs actuels ne sont pas encore dotés de moyens de compréhension du langage naturel. C'est pourquoi, en text mining et plus précisément en catégorisation de textes, les données textuelles doivent être transformées en vue de leur traitement par les algorithmes d'apprentissage automatique.

La méthode la plus communément utilisée est de transformer le texte en vecteur (représentation vectorielle). Chacune des dimensions de ce vecteur est un terme du texte. Une collection de textes peut être rassemblée en une matrice dont les lignes sont les documents de la collection et les colonnes sont les termes qui apparaissent au moins une fois dans les documents. On note par  $t_i$  le terme  $i$  de la collection et  $d_j$  le document  $j$  de la collection.  $w_{ij}$ , représente la fréquence du terme  $i$  dans le document  $j$  de la collection.

La représentation susmentionnée est dite représentation en « sacs de mots » et ne préserve pas l'ordre des mots dans les textes ce qui engendre une perte de sémantique. Une autre représentation possible et qui préserverait l'ordre des mots est la représentation séquentielle. Les textes ne sont plus des vecteurs mais des séquences de mots. Cette représentation a l'avantage de traiter les textes tels qu'ils se présentent mais les études ont montré qu'elle n'apporte pas d'améliorations significatives aux résultats de classification si ce n'est qu'elles exigent des moyens de stockage et d'indexation plus complexes. Cette représentation ne sera pas discutée au niveau de ce mémoire.

Dans ce chapitre nous discuterons du choix des termes à faire apparaître dans la matrice, du calcul de la fréquence  $w_{ij}$  et de la réduction de la dimension de l'espace.

### 2-1 Choix des termes

Le terme ne s'interprète pas nécessairement par mot. Au niveau de cette section nous présenterons les différents modèles de choix de termes utilisés dans la catégorisation de textes.

On transforme un document  $d_j$  en vecteur  $d_j = (w_{1j}, w_{2j}, \dots, w_{|T|j})$  où  $T$  est l'ensemble de termes ou de descripteurs qui apparaissent au moins une fois dans le corpus ou la collection de documents d'apprentissage. Le poids  $w_{ij}$  correspond à la contribution du terme  $t_i$  dans la sémantique du texte  $d_j$ . Notons que nous avons parlé de termes contenus dans le corpus et non dans le document car c'est le tout qui va être représenté par la matrice.

#### 2-1.1 Les termes sont exactement les mots contenus dans les textes

Tout d'abord définissons ce qu'est un mot. Un mot est une suite de caractères appartenant à un dictionnaire.

L'Arabe étant une langue agglutinante. Définir un mot comme étant une suite de caractères non délimiteurs encadrés par des caractères délimiteurs [Gilli 1988] ne convient pas à la langue Arabe de par sa nature agglutinante.

Exemple : واشترأهم - et il les a achetés

En Français nous produisons, selon la définition précédente les termes (mots) suivant : et, il, les, a, achetés alors qu'en Arabe il n'y a qu'un seul terme. Nous détaillerons dans le chapitre consacré à la langue Arabe, les techniques spécifiques à cette langue en vue d'obtenir les termes d'indexation.

Les composantes des vecteurs sont une fonction de l'occurrence des mots dans les textes. Cette représentation exclut toute notion de distance entre les mots et toute analyse grammaticale. C'est pour cette raison qu'elle est dite représentation en sac de mots.

## 2-1.2 Représentation des textes par des phrases

Les phrases sont des entités beaucoup plus informatives que les mots. C'est pourquoi certains auteurs comme [Fuhr et Buckley, 1991, Schütze et al., 1995, Tzeras et Hartmann, 1993], les proposent comme index de documents. L'expression « *Intelligence artificielle* » prise entièrement est plus indicative que les mots « *intelligence* » et « *artificielle* » pris séparément.

Logiquement avec cette représentation on doit atteindre des résultats plus significatifs. Cependant, les expériences ont montré que si les qualités sémantiques sont conservées, les qualités statistiques sont dégradées. Le grand nombre de combinaisons de mots possibles entraîne des fréquences faibles et aléatoires [Lewis 1992b].

Certains auteurs comme [Caropreso et al., 2001], ont utilisés la notion de phrase statistique. Celle-ci est un ensemble de mots contigus mais pas nécessairement ordonnés c'est-à-dire sans signification grammaticale.

Des affinements de ces méthodes peuvent apparaître dans la notion de syntagmes nominaux et verbaux.

Exemple de syntagme nominal : رئيس الجمهورية

Exemple de syntagme verbal : خرج الولد

## 2-1.3 Représentation des textes par des racines lexicales « stemmes et lemmes »

Les flexions des mots « Chacune de ses formes » (أكل، يأكل، أكلا، مأكولات) représente un index à part entière dans la représentation en sac de mots et donc une dimension. Alors que ces mots ont tous la même racine et contribuent de la même façon dans le sens général du texte. Les techniques de stemming et de lemmatisation cherchent à résoudre cette difficulté en représentant le mot par sa racine lexicale.

Pour la recherche de racines lexicales ou stemming (dessuffixation : élimination des affixes), il existe un outil proposé par porter [Porter 1980] pour la langue Anglaise. Des outils semblables pour la langue Arabe sont disponibles. On explicitera celui que nous avons utilisé dans le prétraitement que nous avons effectué sur le corpus utilisé.

Cependant cette technique donne parfois des racines qui n'ont aucun lien avec le mot original.

**Exemple :**

Le mot إيمان une des racines possible étant ایم et qui a le sens : deux veuves qui ne ressemble en rien avec le mot d'origine.

*La lemmatisation est une technique qui cherche à remplacer les mots par leurs racines sans toutefois perdre le sens général du mot. Cette technique est beaucoup plus utilisée dans le traitement automatique du langage naturel (TALN). Elle permet de conserver le sens des mots. Elle s'appuie en général sur des constructions linguistiques et donc beaucoup plus compliquée à mettre en œuvre que le stemming qui, lui repose sur des techniques flexionnelles et dérivationnelles [de Loupy, 2001] donc plus simple à mettre en œuvre.*

En text mining en général et plus particulièrement en catégorisation de textes, le stemming donne des résultats satisfaisants eu égard à sa simplicité. La précision donnée par la lemmatisation n'améliore pas efficacement les résultats. C'est pourquoi en lui préfère des outils de stemming qui sont plus simple à implémenter et moins coûteux.

D'autres techniques comme, celles basées sur les n-grammes sont utilisées. Mais il a été démontré que les résultats obtenus ne sont pas meilleurs [Caropreso et al., 2001]. Cependant, il a été démontré que la reconnaissance de la langue, une des application de la TC (acronyme anglo-saxon de catégorisation de texte), par la technique des n-grammes est meilleure qu'avec d'autres approches on a atteint un taux de reconnaissance de près de 99,8% [Cavnar et trenkle, 1994]

## 2-2 Codage des termes

Dans cette section, nous donnons un aperçu sur les méthodes existantes du choix de la fréquence  $w_{ij}$  comme mentionné précédemment.

Différentes représentations vectorielles sont possibles :

### 2-2.1 Représentation en vecteur binaire

C'est une représentation dite « par mots clefs », elle consiste à coder les termes par leur présence ou absence (1 pour la présence ; 0 pour l'absence). C'est une méthode relativement simple. Elle est surtout utilisée dans l'indexation des documents

Exemple :  $d1 =$  أكل أحمد تفاحات. تلك التفاحات كانت غير طازجة  
 $d2 =$  أكل علي التفاحات طازجة

Les vecteurs associés figurent au tableau 2.1 ci-après

	أكل	أحمد	تفاح	ات	تلك	ال	كان	ت	غير	طازجة	علي
d1	1	1	1	1	1	1	1	1	1	1	0
d2	1	0	1	1	0	1	0	0	0	1	1

Tableau 2.1 exemple de représentation en vecteurs binaires

De manière formelle, si  $d_{binaire}$  est la représentation binaire du document  $d$  de composantes  $d_{binaire}^i$  pour  $i \in [1..|V|]$ , où  $|V|$  représente la taille du vocabulaire, nous pouvons écrire :

$$\forall i \in [1..|V|], \begin{cases} d_{binaire}^i = 1 & \text{si le } i\text{eme terme apparait dans } d \\ 0 & \text{sinon} \end{cases} \quad (2.1)$$

Cette représentation est peu informative car elle ne renseigne ni sur la fréquence d'un mot qui peut constituer une information importante pour la RI (IR en Anglais : information retrieval), ni sur la longueur des documents.

## 2-2.2 Vecteur fréquentiel

Une extension de la représentation binaire prenant en compte le nombre d'occurrences des termes s'appelle représentation fréquentielle.

Exemple : sur la base des mêmes documents présentés précédemment, nous construisons le tableau 2.2 ci-après :

Les vecteurs associés sont :

	أكل	أحمد	تفاح	ات	تلك	ال	كان	ت	غير	طازجة	علي
d1	1	1	2	2	1	1	1	1	1	1	0
d2	1	0	1	1	0	1	0	0	0	1	1

Tableau 2.2 exemple de représentation en vecteurs fréquents

Formellement :

$$\forall i \in [1..|V|], d_{freq}^i = \text{nombre d'apparition du } i\text{eme terme dans } d \quad (2.2)$$

L'inconvénient de cette méthode est que les documents longs auront un vecteur de norme plus grande que les petits documents. Ceci est un inconvénient majeur dans les applications utilisant des normes à base de produits scalaires où il serait plus judicieux de travailler avec des versions de vecteurs fréquents normalisés. Cette normalisation consiste à représenter la fréquence d'un terme par sa proportion dans le document.

Ainsi, la version normalisée de l'exemple précédent peut être représentée par le tableau 2.3 :

	أكل	أحمد	تفاح	ات	تلك	ال	كان	ت	غير	طازجة	علي
d1	0.083	0.167	0.167	0.167	0.167	0.083	0.083	0.083	0.083	0.083	0
d2	0.167	0	0.167	0.167	0	0.167	0	0	0	0.167	0.167

Tableau 2.3 exemple de représentation en vecteurs fréquents normalisés

## 2-2.3 Représentation TF-IDF

### Loi de Zipf

- 1- Plus un terme  $t_k$  est fréquent dans un document, plus il informe sur le sujet du document.
- 2- Plus un terme  $t_k$  est fréquent dans un corpus, moins il est utilisé comme discriminant entre les documents. C'est le cas notamment des mots outils « stop-words » : tels les articles, les prépositions, les mots courants du langage ...

Mieux encore

*« un mot est informatif dans un document si il y est présent souvent mais qu'il n'est pas présent trop souvent dans les autres documents du corpus ».*

Un mot est important s'il n'est ni trop fréquent, ni trop rare comme illustré dans la figure 2.1.

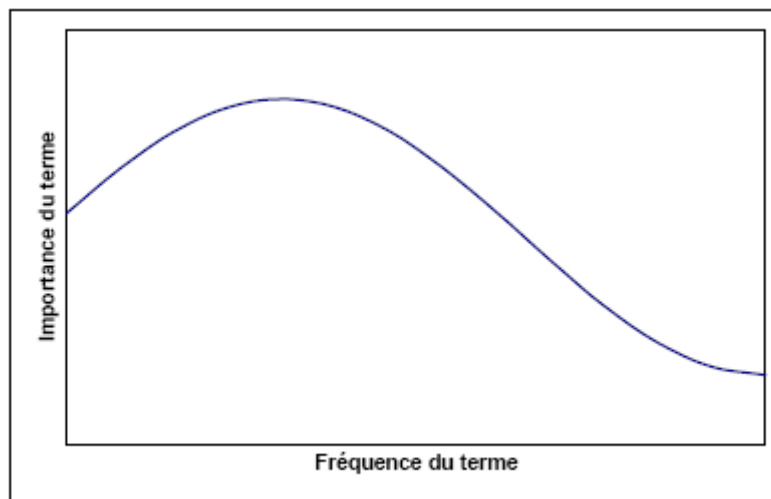


Fig 2.1 : Illustration de la loi de Zipf

Plusieurs codages ont été étudiés pour la prise en compte de la loi de Zipf. Ils reposent tous sur l'hypothèse que la composante du vecteur représentant un document noté  $d_{zipf}^i$  pour le  $i$ ème mot est calculée par le produit entre un facteur qui concerne le poids du terme dans le document et un autre qui concerne le poids du terme dans le corpus :

$$d_{zipf}^i = \text{poids\_dans\_le\_document} \times \text{poids\_dans\_le\_corpus}$$

(pondération locale x Pondération globale)

Le modèle le plus classique est celui pour lequel la première valeur est égale à la fréquence du mot dans le document (notée  $tf_i^d$  pour term frequency : : généralement on utilise les fréquences normalisées) et la seconde valeur est égale à  $\log(|D|)/df_i$  où  $|D|$  est le nombre de documents du corpus et  $df_i$  est le nombre de documents qui contiennent le mot  $i$  ( $df$  signifie document frequency).

Dans ce cas là, cette représentation sera appelée représentation TF-IDF (term frequency inverse document frequency). Elle correspond à la représentation suivante :

$$\forall i \in [1..|V|], \quad d_{tf-idf}^i = tf_i^d \times \log\left(\frac{|D|}{df_i}\right) \quad (2.3)$$

La représentation TF-IDF est une représentation très utilisée en RI aussi bien en recherche documentaire qu'en classification [Salton, 1988],[Joachims, 1999]).

Dans l'étude que nous avons menée, c'est cette représentation que nous avons adoptée.

### Exemple :

Reprenons le même exemple. Sa représentation TF-IDF est présentée dans le tableau 2.4

	أكل	أحمد	تفاح	ات	تلك	ال	كان	ت	غير	طازجة	علي
d1	0	0.116	0	0	0.116	0	0.058	0.058	0.058	0	0
d2	0	0	0	0	0	0	0	0	0	0	0.116

Tableau 2.4 exemple de représentation TF-IDF

On remarque bien que les termes **علي** et **أحمد** sont les plus discriminants ce qui confirme notre intuition.

On note que plusieurs autres méthodes respectant les principes de la loi de Zipf ont été utilisées en représentation de textes apportant chacune une amélioration par rapport au codage tf-idf. On cite la mesure tfc qui prend en compte la longueur des documents avec sa variante ltc qui prend en considération la réduction de l'effet des grandes différences de fréquences et enfin la mesure d'entropie basée sur la théorie de l'information.

Il est à noter que tf-idf est la mesure la plus utilisée eu égard à sa simplicité et que les améliorations apportées par les méthodes ci-dessus sont insignifiantes par rapport à la complexité des calculs à entreprendre lors de leurs réalisations.

## 2-3 Réduction de dimension

La grande dimension de l'espace des documents est un problème central dans la catégorisation de textes. Le nombre de descripteurs (dimensions) peut facilement atteindre plusieurs dizaines de milliers pour un corpus de taille raisonnable. De ce fait, on est obligé de sélectionner un espace de représentation beaucoup plus réduit par l'élimination des descripteurs les moins informatifs et ce, pour deux raisons :

- 1- le coût de traitement d'un grand espace intervient dans la complexité de l'algorithme d'apprentissage choisi. Imaginons la construction d'un arbre de décision sur une base de quelques milliers de descripteurs !!!,
- 2- on ne peut construire de règles fiables à partir de termes de faible occurrence.

De ce fait, on peut conclure que l'élimination peut porter sur :

- 1- les mots très fréquents car ils sont présents partout dans le corpus d'où leur faible apport informatif. C'est le cas notamment des mots outils (Stop words) à l'instar des conjonctions et des adverbess...,
- 2- les mots très rares ne concernant qu'une faible proportion de documents (cf. loi de Zipf). Ils peuvent également être issus d'une mauvaise orthographe.

Cependant, même avec l'élimination de ces deux catégories de mots, l'espace de représentation reste élevé. Il faut alors, utiliser une méthode statistique pour extraire les mots les plus pertinents. Dans [Sebastiani, 2002], on peut trouver une étude détaillée de ces techniques. Elles sont classées en deux catégories selon qu'elles agissent localement (documents par documents) ou globalement (dans le corpus en entier) ou selon la nature des résultats de la sélection (S'agit-il d'une sélection de termes ou d'une extraction de termes)

### 2-3.1 Réduction locale de la dimension

Il s'agit de proposer pour chaque catégorie  $c_i$  Un ensemble de termes  $T_i'$  dont la cardinalité est nettement inférieure à la cardinalité de l'ensemble initial [Apté et al., 1994, Lewis et Ringuette, 1994, Schütze et al., 1995, Wiener et al., 1995, Ng et al., 1997, Li et Jain, 1998, Sable et Hatzivassiloglou, 2000].

Avec cette technique, chaque catégorie  $c_i$  possède son propre ensemble de termes et chaque document  $d_j$  sera représenté par un ensemble de vecteurs  $d_j$  différents selon la catégorie. Habituellement,  $10 < |T_i'| < 50$ .

### 2-3.2 Réduction globale de dimension

Dans ce cas, le nouvel ensemble de termes  $T'$  est choisi en fonction de toutes les catégories. Ainsi, chaque document  $d_j$  sera représenté par un seul vecteur  $d_j$  quelque soit la catégorie [Yang et Pedersen, 1997, Mladeni'c et Grobelnik, 1998, Caropreso et al., 2001, Yang et Liu, 1999].

### 2-3.3 Sélection de termes

Les techniques de réduction de dimensions par sélection visent à proposer un nouvel ensemble  $T'$  avec  $|T'| \ll |T|$ . Parmi ces techniques figurent le calcul de l'information mutuelle [Lewis, 1992a, Moulinier, 1997, Dumais et al., 1998], la statistique du  $\chi^2$  [Schütze et al., 1995], ou des méthodes plus simples utilisant uniquement les fréquences d'apparitions [Wiener et al., 1995, Yang et Pedersen, 1997] ; d'autres méthodes ont également été testées [Moulinier, 1996, Sahami, 1999]

**Le seuil de fréquence d'un terme** : Il s'agit de calculer la fréquence de chaque terme dans le corpus, les termes dont le seuil de fréquence ne dépasse pas un certain seuil fixé (en général plus de 3) seront éliminés.

**Le gain d'information** ( «*information gain*» ) : On mesure en quelque sorte le pouvoir de discrimination d'un mot. Le nombre de bits d'information obtenue 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.

Formellement le gain G d'information d'un terme t est mesuré comme suit :

$$G(t) = -\sum_{i=1}^m \Pr(c_i) \log \Pr(c_i) + \Pr(t) \sum_{i=1}^m \Pr(c_i/t) \log \Pr(c_i/t) + \Pr(\bar{t}) \sum_{i=1}^m \Pr(c_i/\bar{t}) \log \Pr(c_i/\bar{t}) \quad (2.4)$$

**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 apparaisse 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.

Formellement, on procède comme suit : On construit 2 tables de contingences pour chaque terme t avec une classe c. A est le nombre de fois où t et c co-occurrent. B le nombre de fois où t apparaît sans c. C le nombre de fois où c apparaît sans le terme t et N le nombre de documents. L'information mutuelle entre t et c notée I est estimée comme suit :

$$I(t, c) \approx \log\left(\frac{A \times N}{(A + C) \times (A + B)}\right) \quad (2.5)$$

**La statistique du  $\chi^2$**  : 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.

Une 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.

Formellement, elle est calculée comme suit :

$$\chi^2(t, c) \approx \left(\frac{N \times (AD - CB)^2}{(A + C) \times (A + B) \times (A + B) \times (C + D)}\right) \quad (2.6)$$

où D est le nombre de fois où t et c n'apparaissent pas du tout.

**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 cosinusoidale 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.

Formellement elle est mesurée comme suit :

$$s(t) = \Pr(t \in y / t \in x) \quad (2.7)$$

## 2-3.4 Extraction de termes

L'objectif des techniques d'extraction de termes est de proposer un sous-ensemble  $T'$  avec  $|T'| \ll |T|$  mais, à la différence des techniques de sélection, le sous-ensemble  $T'$  est une synthèse (combinaison linéaire des descripteurs) qui devrait maximiser la performance. On recherche des variables synthétiques pour éliminer les problèmes liés aux synonymies, polysémies et homonymies en proposant des variables artificielles, jouant le rôle de nouveaux « termes ».

L'une des approches est appelée le « Latent Semantic Indexing (LSI) », proposée par [Deerwester et al., 1990]. La LSI est fondé sur l'hypothèse d'une structure latente des termes, identifiables par les techniques factorielles. Il consiste en une décomposition en valeurs singulières de la matrice dans laquelle chaque document est représenté par la colonne des occurrences des termes qui le composent. Le LSI est très proche de l'analyse des correspondances [Morin, 2002], introduite par [Benzecri, 1976] et [Escofier, 1965] pour traiter les données textuelles.

**L'indexation par la sémantique latente :** (LSI : Latent semantic indexing).

Un texte étant décrit par un vecteur dont chaque composante correspond à un mot d'un vocabulaire  $V$ . La LSI effectue une projection des textes et des mots dans un espace factoriel. On part de l'hypothèse que la similarité entre deux vecteurs-documents implique une proximité sémantique des documents correspondants et que une forte similarité entre deux vecteurs-mots indique la synonymie des deux mots [Deerwester et al., 1990].

### 1- Principe

Soit la matrice  $A$  dont chaque colonne est un vecteur-document (donc autant de colonnes qu'il y a de documents : notons ce nombre  $N$ ). Chaque ligne correspond à un mot du vocabulaire que l'on suppose de taille  $P = |V|$ . Les composantes  $A(m,t)$  de la matrice  $A$  sont en relation avec le nombre d'occurrences du mot  $m$  dans le texte  $t$  (comme décrit dans la section sur la représentation des textes ( TF-IDF)

On effectue ensuite l'ACP de la matrice  $A$ . Rappelons que l'on a ici deux types de données mêlées dans cette matrice (des mots et des textes). On veut donc extraire des axes principaux pour ces deux types de données.

Pour cela, on effectue une décomposition en valeurs singulières de  $A$ .  $A$  est donc une matrice de taille  $M \times N$ . La décomposition en valeurs singulières fournit les matrices  $U$ ,  $\Sigma$  et  $V$  telles que :

$$A = U \Sigma V^T$$

$U$  est une matrice  $M \times M$  alors que  $\Sigma$  et  $V$  sont des matrices  $N \times N$ . On a (propriétés de la décomposition en valeurs singulières) :

- i.  $U^T U = V^T V = I_N$  où  $I_N$  est la matrice identité  $N \times N$  ;
- ii.  $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_N)$  avec  $\sigma_i > 0$  pour  $1 \leq i \leq r$  et  $\sigma_i = 0$  pour  $i > r$ .  $r$  est nommé le rang de la matrice  $A$ . Si  $r < N$ , on dit que la matrice  $A$  est singulière ;
- iii. les  $\sigma_i$  sont les racines carrées des valeurs propres de  $A^T A$  et  $AA^T$  ;
- iv. les vecteurs propres de  $AA^T$  constituent les colonnes de  $U$  et se nomment les vecteurs singuliers à gauche de  $A$  ;
- v. les vecteurs propres de  $A^T A$  constituent les colonnes de  $V$ , lesquels sont dénommés les vecteurs singuliers à droite de  $A$ .

Pour comprendre l'intérêt de ce qui précède et le lien avec une ACP, il est important de noter que :

1.  $AA^T$  est une matrice dont chaque terme correspond au produit scalaire entre une ligne de  $A$  (un mot) et une ligne de  $A$  (un autre mot). Donc, chacun de ses termes représente le cosinus entre ces deux mots. Donc, chacun des mots  $m_1, m_2$  de cette matrice représente la similarité entre les deux mots  $m_1$  et  $m_2$ , la similarité étant mesurée par la répartition de ces deux mots parmi les textes ;
2.  $A^T A$  est une matrice dont chaque terme correspond au produit scalaire entre une colonne de  $A$  (un texte) et une colonne de  $A$  (un autre texte). Donc, chacun de ces termes mesure la similarité entre deux textes ;
3. donc,  $AA^T$  et  $A^T A$  sont donc des matrices de similarité. La décomposition en vecteurs propres de ces matrices est connue pour apporter des informations très intéressantes. Ces vecteurs propres sont dans les matrices  $U$  et  $V$  obtenues par décomposition en valeurs singulières. Quant à elles, les valeurs propres sont dans  $\Sigma$ .

En n'utilisant que les  $k$  premières valeurs singulières avec leur vecteurs propres à gauche et à droite respectifs, on obtient une approximation  $A_k$  de la matrice  $A$  de départ :

$$A_k = U_k \Sigma_k V_k^T$$

où  $U_k$  et  $V_k$  sont composés des  $k$  premières colonnes de  $U$  et  $V$  et  $\Sigma_k = (\sigma_1, \dots, \sigma_k)$ .  $A_k$  est la meilleure approximation de rang  $k$  de  $A$  au sens des moindres carrés :

$\beta$

$$A_k = \arg \min_{\beta \in \text{matrices de rang } k} \|A_k - \beta\|^2.$$

Ainsi, d'une certaine manière,  $A_k$  capture les particularités les plus importantes de la matrice  $A$  initiale. Le fait de laisser de côté un certain nombre de dimensions permet de se débarrasser (au moins en partie) des problèmes liés à la polysémie et à la synonymie.

### Projection dans l'espace factoriel

Soit un texte  $t$  que l'on souhaite projeter dans l'espace factoriel. Ce texte est décrit à partir du vocabulaire  $V$ . Sa projection est  $t_p = t^T U_k \Sigma_k^{-1}$

### Interprétation

L'exploitation est riche. On peut s'intéresser à différentes questions :

- vi. pour un mot donné  $m$ , quels sont les mots (( proches )) i.e. dont l'apparition dans les textes est à peu près la même. Cela va permettre d'exhiber des synonymes ;

- vii. pour un texte donné  $t$ , quels sont les textes proches ? Cela va permettre de trouver des textes traitant du même sujet ;
- viii. pour un mot ou un ensemble de mots donnés, quels sont les textes pour lesquels ce(s) mot(s) sont les plus significatifs ? Cela va permettre une interrogation par mots-clés.

### **Recherche de synonymes**

Soit un mot  $m \in V$  pour lequel on veut trouver les mots qui sont utilisés de la même manière. On constitue un texte qui ne contient qu'un seul mot,  $m$  et on projette ce texte dans l'espace factoriel. Il ne reste plus qu'à chercher les mots de  $V$  dont la projection est proche par une approche (( plus proches voisins )).

### **Recherche de textes identiques**

Soit un texte  $t$ , qui fait partie de l'ensemble des textes initiaux ou pas. On décrit ce texte par sa composition en mots du vocabulaire  $V$  et on le projette. De même, on calcule la projection de chacun des  $N$  textes et on détermine les plus proches.

### **Interrogation par mots-clés**

On suppose que les mots-clés font partie de  $V$ . Dès lors, on constitue un texte  $t$  comprenant ces mots, on le projette et, à nouveau, on recherche les textes dont la projection est la plus proche. Bien entendu, les composantes des pseudo textes sont pondérées si nécessaire, comme l'ont été les composantes de  $A$ .

### **Mots et leur traduction**

On dispose d'un même ensemble de textes dans deux langues. La LSI va projeter chaque mot d'une langue près de sa traduction dans l'autre langue, ainsi que chaque paire de textes l'un près de l'autre. Cela permet de trouver la traduction d'un terme et d'effectuer une interrogation dans une langue et d'obtenir le résultat dans l'autre langue.

Enfin, de nouvelles variantes de cette technique ont été développées par [Hofmann *et al.* 1999] [Saul et Pereira, 1997] donnant lieu à ce qu'on appelle PLSA (Probabilistic latent semantic analysis)

## **2- Exemple [Bladi et al., 2003]**

Une collection de documents notée  $C$  contenant 10 documents dont les 05 premiers sont relatifs au système d'exploitation Linux et les 05 autres au domaine de la génétique. Nous nous intéressons aux termes qui apparaissent au moins 02 fois dans la collection en enlevant les mots communs. Les valeurs singulières de la matrice

Les termes utilisés dans l'analyse sont soulignés. La matrice termes-documents est notée  $A^T$ , présenté dans le tableau 2.5 La matrice reconstruite pour  $k=2$  est notée  $A_k^T$ , présenté dans le tableau 2.6

C

- d1: Indian government goes for open-source software
- d2: Debian 3.0Woody released
- d3: Wine 2.0 released with fixes for Gentoo 1.4 and Debian 3.0
- d4: gnu POD released: iPod on Linux... with GPLed software
- d5: Gentoo servers running an open-source mySQL database
- d6: Dolly the sheep not totally identical clone
- d7: DNA news: introduced low-cost human genome DNA chip
- d8: Malaria-parasite genome database on theWeb
- d9: UK sets up genome bank to protect rare sheep breeds
- d10: Dolly's DNA Damaged

$A^T$

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
<b>Open-source</b>	1	0	0	0	1	0	0	0	0	0
<b>Software</b>	1	0	0	1	0	0	0	0	0	0
<b>Linux</b>	0	0	0	1	0	0	0	0	0	0
<b>Released</b>	0	1	1	1	0	0	0	0	0	0
<b>Debian</b>	0	1	1	0	0	0	0	0	0	0
<b>Gentoo</b>	0	0	1	0	1	0	0	0	0	0
<b>Database</b>	0	0	0	0	1	0	0	1	0	0
<b>Dolly</b>	0	0	0	0	0	1	0	0	0	1
<b>Sheep</b>	0	0	0	0	0	1	0	0	1	0
<b>Genome</b>	0	0	0	0	0	0	1	1	1	0
<b>DNA</b>	0	0	0	0	0	0	2	0	0	1

Tableau 2.5 : Matrice  $A^T$  de l'exemple pour la LSI

$A_k^T$

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
<b>Open-source</b>	0.34	0.28	0.38	0.42	0.24	0.00	0.04	0.07	0.02	0.01
<b>Software</b>	0.44	0.37	0.50	0.55	0.31	-0.01	-0.03	0.06	0.00	-0.02
<b>Linux</b>	0.44	0.37	0.50	0.55	0.31	-0.01	-0.03	0.06	0.00	-0.02
<b>Released</b>	0.63	0.53	0.72	0.79	0.45	-0.01	-0.05	0.09	-0.00	-0.04
<b>Debian</b>	0.39	0.33	0.44	0.48	0.28	-0.01	-0.03	0.06	0.00	-0.02
<b>Gentoo</b>	0.36	0.30	0.41	0.45	0.26	0.00	0.03	0.07	0.02	0.01
<b>Database</b>	0.17	0.14	0.19	0.21	0.14	0.04	0.25	0.11	0.09	0.12
<b>Dolly</b>	-0.01	-0.01	-0.01	-0.02	0.03	0.08	0.45	0.13	0.14	0.21
<b>Sheep</b>	-0.00	-0.00	-0.00	-0.01	0.03	0.06	0.34	0.10	0.11	0.16
<b>Genome</b>	0.02	0.01	0.02	0.01	0.10	0.19	1.11	0.34	0.36	0.53
<b>DNA</b>	-0.03	-0.04	-0.04	-0.06	0.11	0.30	1.70	0.51	0.55	0.81

Tableau 2.6 : Matrice  $A_k^T$  de l'exemple pour la LSI

Nous pouvons remarquer que les termes software et Linux ont des vecteurs identiques. Aussi les vecteurs Debian ou Gentoo sont proches au vecteur Linux. Ainsi que les vecteurs DNA et Dolly qui sont proches au vecteur Génome.

De cette manière, si on s'intéresse à la réduction de l'espace de représentation. Un terme des ensembles proches pourrait être retenu et les autres éliminés.

## 2-4 Quel est le meilleur nombre de termes à conserver ?

L'objectif des méthodes de réduction de termes est de fournir une liste de termes plus courte mais porteuse d'information. Les termes sont en général ordonnés du terme le plus important au moins important selon un certain critère. La question qui se pose concerne le nombre de termes à conserver [Stricker, 2000]. Ce nombre dépend souvent du modèle, puisque, par exemple, les machines à vecteurs supports sont capables de manipuler des vecteurs de grandes dimensions alors que, pour les réseaux de neurones, il est préférable de limiter la dimension des vecteurs d'entrées. Pour choisir le bon nombre de descripteurs, il faut déterminer si l'information apportée par les descripteurs en fin de liste est utile, ou si elle est redondante avec l'information apportée par les descripteurs du début de la liste. Dans son utilisation des machines à vecteurs supports, [Joachims, 1998] considère l'ensemble des termes du corpus Reuters, après suppression des mots les plus fréquents et l'utilisation de racines lexicales (les stemmes). Il reste alors 9.962 termes distincts qui sont utilisés pour représenter les textes en entrée de son modèle. Il considère que chacun de ces termes apporte de l'information, et qu'il est indispensable de les inclure tous.

Au contraire, [Dumais et al., 1998] utilisent également les machines à vecteurs supports mais ne conservent que 300 descripteurs pour représenter les textes. Ils obtiennent néanmoins de meilleurs résultats que Joachims sur le même corpus ; cela laisse à penser que tous les termes utilisés par Joachims n'étaient pas nécessaires. Dans leur article sur la sélection de descripteurs, [Yang et Pedersen, 1997], critiquent [Koller et Sahami, 1997] qui étudient l'impact de la dimension de l'espace des descripteurs en considérant des représentations allant de 6 à 180 descripteurs. Pour [Yang et Pedersen, 1997], une telle étude n'est pas pertinente, car l'espace des descripteurs doit être de plus grande dimension ("an analysis on this scale is distant from the realities of text categorization") ; à l'opposé, d'autres auteurs considèrent qu'un très petit nombre de descripteurs pertinents suffit pour construire un modèle performant. Par exemple, [Wiener et al., 1995] ne retiennent que les vingt premiers descripteurs en entrée de leurs réseaux de neurones. Entre ces deux ordres de grandeurs, d'autres auteurs choisissent de conserver une centaine de termes en entrée de leur modèle [Lewis, 1992b, Ng et al., 1997].

Finalement, il n'est pas prouvé qu'un très grand nombre de descripteurs soit nécessaire pour obtenir de bonnes performances, puisque, même avec des modèles comme les machines à vecteurs supports qui sont, en principe, adaptées aux vecteurs de grandes dimensions, les résultats sont contradictoires. Ceci est sans doute dû à ce que les descripteurs sont corrélés mutuellement, et à la façon dont les différents algorithmes gèrent ces corrélations.

## 2-5 Conclusion

Au niveau de ce chapitre nous avons abordé les différentes représentations de textes et nous avons détaillé la plus communément utilisée et qui consiste en la représentation par les mots (bag of words) avec la fréquence TF-IDF. Nous avons aussi mentionné que cette représentation possède une caractéristique essentielle où réside toute la difficulté et qui est celle de la grande dimensionnalité. A cet effet, nous avons présenté une panoplie de techniques pour sa réduction.

**Chapitre 3**  
**spécificités de la langue**  
**Arabe**  
**et travaux sur la**  
**catégorisation de textes**  
**Arabes**

## Introduction

La langue Arabe est l'une des 06 langues officielles des nations unies et est parlée par plus de 300 millions d'habitants [Aljlal et Frieder, 2002]

Le nombre de sites en langue Arabe ne cesse de croître. En 2000, on a recensé près de 200000 sites ce qui représente 7% de l'ensemble des sites publiés sur le Web [Abdelali et al., 2004]

La langue Arabe est une langue difficile à maîtriser dans le TALN eu égard à ses propriétés morphologiques et syntaxiques [Aljlal et Frieder, 2002] et [Larkey et al., 2002]. 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.

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...

*A la différence des autres langues comme le Français ou l'Anglais, dont les étiquettes grammaticales proviennent d'une approche distributionnelle caractérisée par une volonté "d'écarter toute considération relative au sens", les étiquettes de l'Arabe viennent d'une approche où le sémantique côtoie le formel lié à la morphologie du mot, sans référence à la position de ce dernier dans la phrase [Débili F., Achour H., Souici E, 2002]*

Ce phénomène est matérialisé par les notions de schèmes et de fonctions qui occupent une place importante dans la grammaire de l'Arabe. Par exemple le mot Français *ferme*, est hors contexte, un substantif, un adjectif ou un verbe. Alors que le mot Arabe RaLaKa غلق est un verbe à la 3<sup>e</sup> personne masculin singulier de l'accompli actif, par contre sa forme non voyellée غلق (dans l'exemple donné ne sont représentées que les consonnes RLK) admet quatre catégories grammaticales :

- substantif masculin singulier (RaLKun : une fermeture),
- verbe à la 3<sup>e</sup> personne masculin singulier de l'accompli actif (RaLaKa : il a fermé ou RaLLaKa il a fait fermé),
- verbe à la 3<sup>e</sup> personne masculin singulier de l'accompli passif (RuLiKa : il a été fermé),
- verbe à l'impératif 2<sup>e</sup> personne masculin singulier (RaLLiK: fais fermer).

Les voyelles jouent un rôle proche des accents en Français pour un mot comme « *peche* » qui peut être interprété comme *pêche, pèche et péché*. Par contre, en Arabe chaque lettre de chaque mot devrait posséder sa voyelle ce qui n'est en général pas le cas.

On constate donc l'étendue du rôle que jouent les voyelles dans les mots Arabes, non seulement parce qu'elles enlèvent l'ambiguïté, mais aussi parce qu'elles donnent l'étiquette grammaticale d'un mot indépendamment de sa position dans la phrase.



### 3.1 Particularité de la langue Arabe

L'alphabet de la langue Arabe compte 28 consonnes (cf. tableau 3.1) et peut être étendue à 90 en rajoutant Les voyelles, les caractères spéciaux et les différentes formes d'une même lettre (début, milieu ou fin du mot). [Aljlayl et frieder, 2002]

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). Le Tableau 3.2 montre les variations de la lettre ج (jim). Toutes les lettres se lient entre elles sauf ( , و , ر , ز , د , ذ ) qui ne se joignent pas à gauche.

Lettre Arabe	Correspondant Français	Prononciation	Lettre Arabe	Correspondant Français	prononciation
ا	A	Alef	ض	d	Dad
ب	B	Ba'	ط	t	Tah
ت	T	Ta'	ظ	Th	Thah
ث	Th	Tha'	ع	"	Ayn
ج	J	Jim	غ	gh	Ghayn
ح	H	Hha	ف	f	Fa
خ	Kh	Kha	ق	q	Qaf
د	D	Dal	ك	k	Kaf
ذ	D	Thal	ل	l	Lam
ر	R	Ra	م	m	Mim
ز	Z	Zayn	ن	n	Nun
س	S	Sin	ه	h	Ha
ش	Sh	Shin	و	w	Waw
ص	S	Sad	ي	y	Ya

Tableau 3.1 : alphabet Arabe

A la fin d'une lettre non joignable	A la fin	Au milieu	Au début
ج	ج	ج	ج

Tableau 3.2 : Variation de la lettre ج jim

Les voyelles en Arabe sont ajoutées au-dessus ou au-dessous des lettres (ـَ، ـُ، ـِ، ـِ). Elles sont nécessaires à la lecture et à la compréhension correcte d'un texte. Elles permettent de différencier des mots ayant la même représentation. Cependant, les voyelles ne sont utilisées que pour des textes sacrés et didactiques. Les textes courants rencontrés dans les journaux et les livres n'en comportent habituellement pas. De plus certaines lettres comme ا Alef peuvent symboliser le آ، أ ou إ; de même que pour les lettres ي et ه qui symbolisent respectivement ي et ه [Xu et al., 2002].

Dans l'exemple du tableau 3.3 pour les racines أكل et سلم, on remarque que la voyellisation contribue à la levée de l'ambiguïté

Mot sans voyelles	1 <sup>ère</sup> interprétation		2 <sup>ème</sup> interprétation		3 <sup>ème</sup> interprétation	
أكل	أكل	Il a mangé	أكل	Nourriture	أكل	Il a été mangé
سلم	سلم	Il a remis	سلم	Paix	سلم	Echelle

Tableau 3.3 : Ambiguïté causée par l'absence de voyelles pour les mots أكل et سلم

### 3-1.1 Morphologie Arabe

Trois catégories de mots forment le lexique Arabe : verbes, noms et particules.

Les racines des verbes et des noms sont souvent à trois consonnes radicales [Baloul et al., 2002] et peuvent être à quatre et parfois à cinq consonnes radicales. Un même concept sémantique peut engendrer une famille de mots 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 selon des schèmes qui sont des adjonctions et des manipulations de la racine. On peut ainsi dériver un grand nombre de noms, de formes et de temps verbaux.

Schémes	FTH	فتح	Notion d'ouvrir	MNH	منح	Notion de donner
R1aR2aR3a	FaTaHa	فتح	Il a ouvert	MaNaHa	منح	Il a donné
R1âR2iR3	FâTiH	فاتح	conquérant	MâNiH	مانح	Donateur
maR1R2uR3	maFTuH	مفتوح	Ouvert	maMNUH	ممنوح	Donné
R1aR2R3atun	FaTHatun	فتحة	Ouverture	MaNHatun	منحة	Bourse, donne
R1âR2iR3un	FâTiHun	فاتحون	Conquérants	MâNiHun	مانحون	Donateurs
...						

Tableau 3.4 : Différentes dérivations des racines فتح et منح

Dans le tableau 3.4, les lettres en majuscule (Ri) désignent les consonnes de base qui composent la racine. Les voyelles (â, a, i,..) désignent les voyelles et les consonnes en minuscule (m,..) sont des consonnes de dérivation utilisées dans les schèmes.

La majorité des verbes Arabes ont une racine composée de 3 consonnes. On recense près de 10000 racines lexicales [Darwish, 2003]. L'Arabe comprend environ 150 schèmes ou patrons dont certains plus complexes, tel le redoublement d'une consonne ou l'allongement d'une voyelle de la racine, l'adjonction d'un ou de plusieurs éléments ou la combinaison des deux. Une autre caractéristique est le caractère flexionnel des mots : les terminaisons permettent de distinguer le mode des verbes et la fonction des noms [Baloul et al., 2002].

### 3-1.2 Structure d'un mot

Une phrase en Arabe peut être agglutinée en un seul mot. La représentation suivante schématise une structure possible d'un mot. Notons que la lecture et l'écriture d'un mot se fait de droite vers la gauche.

Post fixe	suffixe	Corps schématique	préfixe	antéfixe
-----------	---------	-------------------	---------	----------

- Les antéfixes sont des prépositions ou des conjonctions (...و،س،...)
- Les préfixes et suffixes expriment les traits grammaticaux et indiquent les fonctions : cas du nom, mode du verbe et les modalités (nombre, genre, personne,...)
- Les postfixes sont des pronoms personnels.

#### Exemple

أَتَكْتُبُونَهَا

Ce mot exprime la phrase en Français : "Est ce que vous l'écrivez ?" La segmentation de ce mot donne les constituants suivants :

	أ	ت	كتب	ون	ها
Antéfixe	:	أ	conjonction d'interrogation		
Préfixe	:	ت	préfixe verbal du temps de l'inaccompli.		
Corps schématique	:	كتب	dérivé de la racine: كتب selon le schème R1R2uR3u		
Suffixe	:	ون	suffixe verbal exprimant le pluriel		
Post fixe	:	ها	pronom suffixe complément du nom		

### 3-1.3 Catégories des mots

L'Arabe considère 3 catégories de mots :

- le verbe : 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,
- le nom : l'élément désignant un être ou un objet qui exprime un sens indépendant du temps,
- 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.

#### 3-1.3.1 Le verbe

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 et/ou des préfixes. 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).

Par exemple : ج + ر + خ *Kh+R+J* donne le verbe خرج *KhaRaJa*. (sortir).

Dans tous les mots qui dérivent de cette racine, on trouvera ces trois consonnes Kh, R, J

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.

- L'accompli : correspond au passé et se distingue par des suffixes (par exemple pour le pluriel féminin on a خرجن *KhaRaJna*, *elles sont sorties* et pour le pluriel masculin on a خرجوا *KhaRaJuu*, *ils sont sortis*),
- L'inaccompli présent: présente l'action en cours d'accomplissement. Ses éléments sont préfixés ( يخرج *yaKhRuJu* *il sort*; تخرج *taKhRuJu*, *elle sort*),
- L'inaccompli futur : correspond à une action qui se déroulera au futur et est marqué par l'antéposition de س *sa* ou سوف *sawfa* au verbe ( سيخرج *sayaKhRuJu* *il sortira*, سوف يخرج *sawfa yaKhRuJu* *il va sortir*).

### 3-1.3.2 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.

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 صغيرات *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 [Kiraz, 1996] comme pour les verbes irréguliers.

Certain dérivés nominaux associent une fonction au nom :

- agent (celui qui fait l'action),
- objet (celui qui a subi 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 féminin.

### 3-1.3.3 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 jouent un rôle important dans l'interprétation de la phrase [Kadri et Benyamina, 1992]. Elles servent à situer des faits ou des objets par rapport au temps ou au lieu. Elles jouent également un rôle clé dans la cohérence et l'enchaînement d'un texte.

Comme exemple de particules qui désignent un temps *بعد*, *قبل*, *منذ* *pendant*, *avant*, *après*, un lieu *حيث* *où*, ou de référence *الذين* *ceux*,....

Les particules peuvent avoir des préfixes et suffixes ce qui rajoute une complexité quant à leur identification.

## 3-2 Problèmes du traitement automatique de l'Arabe

Un des aspects complexes de la langue Arabe est l'absence des voyelles dans le texte, qui risque de générer une certaine ambiguïté à deux niveaux :

- sens du mot,
- difficulté à identifier sa fonction dans la phrase, (différencier entre le sujet et le complément,...).

### 3-2.1 Détection de racine

Pour détecter la racine d'un mot, il faut connaître le schème par lequel il a été dérivé et supprimer les éléments flexionnels (antéfixes, préfixes, suffixes, post fixes) qui ont été ajoutés.

Nous utilisons la liste de préfixes et de suffixes proposé par [Darwish, 2003] voir Tableau 3.5. Plusieurs d'entre eux ont été utilisés par [Chen et Gey, 2002] pour la radicalisation de mots Arabes. Ils ont été déterminés par un calcul de fréquence sur le corpus d'articles Arabes de l'Agence France Press (AFP).

Préfixes							
والـ	بـ	تـ	بـ	مـ	لـ	فـ	لا
فالـ	يـ	سـ	لـ	فـ	لـ	وا	با
بالـ	مـ	نـ	وـ	الـ	وـ	فا	
Suffixes							
اتـ	وہـ	تہـ	ہمـ	یہـ	ینـ	ةـ	ا
واـ	انـ	نمـ	هنـ	تکـ	یہـ	ہـ	
ونـ	تیـ	کمـ	هاـ	ناـ	یہـ	یـ	

Tableau 3.5 : Les affixes les plus fréquents en langue Arabe

L'analyse morphologique devra donc séparer et identifier des morphèmes semblables aux mots préfixés comme les conjonctions wa- و et fa- ف, des prépositions préfixées comme bi- ب et li- ل, l'article défini ال, des suffixes de pronom possessif. La phase d'analyse morphologique détermine un schème possible. Les préfixes et suffixes sont trouvés en enlevant progressivement des préfixes et des suffixes et en essayant de faire correspondre toutes les racines produites par un schème afin de retrouver la racine [Darwish, 2002].

Lorsqu'un mot peut être dérivé de plusieurs racines différentes, la détection de la racine est encore plus difficile, en particulier en absence de voyelles [Attia, 2000]. Par exemple dans le tableau 3.6, pour le mot Arabe ایمان AymAn les préfixes possibles sont : "Ø", "A ا" et "Ay اي" et les suffixes possibles sont : "Ø" et "An ان", sans compter que ce mot peut aussi représenter un nom propre ایمان *Imène*.

Stem	Préfixe	Schème	Suffixe	Racine	Signification
AyMaN ایمان	Ø	R1yR2aR3	Ø	AMN امن	Croyance
YMaN یمان	ا A	R1R2aR3	Ø	YMN یمن	Convenant
MAN مان	اي Ay	R1R2R3	Ø	MAN مان	Va-t-il approvisionner
AYM ایم	Ø	R1R2R3	ان An	AYM ایم	Deux veuves

Tableau 3.6 : Différentes radicalisations du terme ایمان

Certains verbes sont considérés comme irréguliers, ce sont ceux qui portent des consonnes particulières dites faibles ( و, ا, ي ). Ils sont appelés ainsi parce que, lors de leur déclinaison, chacune de ces lettres est soit conservée, soit remplacée ou éliminée [Kadri et Benyamina, 1992].

On voit bien dans le tableau 3.7 que la racine du verbe قال n'est pas conservée.

Caractère ا est remplacé par	قال	Dire
ا	قال	Il a dit
و	يقول	Il dit
ي	قيل	Il a été dit
Ø	قل	dis

Tableau 3.7 : Exemple de déclinaison du verbe irrégulier قال *dire*

### 3-2.2 l'agglutination

Une difficulté en traitement automatique de l'Arabe est l'agglutination par laquelle les composantes du mot sont liées les unes aux autres. Ce qui complique la tâche de l'analyse morphosyntaxique pour identifier les vrais composants du mot.

Par exemple, le mot ألمُّهم ALaMuhum, *leur douleur* dans sa forme voyellée n'accepte qu'une seule segmentation : ألمُّ + همُّ (ALaMu+hum).

Dans sa forme non voyellée المهم (ALMHM), le même mot accepte au moins les trois segmentations présentées dans le tableau 3.8

Segmentation possible		Traduction en Français
أ + لم + هم	a + LM + hm	Les a-t-il ramassé
ألم + هم	ALM + hm	Leur douleur
	ALM + hm	Il les a fait souffrir
أل + مهم	al + MHM	L'important

Tableau 3.8 : Exemple de segmentation du mot المهم

L'amplification de l'ambiguïté de segmentation s'opère selon deux façons [Débili et al., 2002]:

- d'abord, il y a plus d'unités ambiguës dans un texte non voyellé que dans son correspondant voyellé,
- mais aussi, les unités ambiguës acceptent plus de segmentations dans le texte non voyellé.

De plus le fait de précéder la radicalisation par la troncature des préfixes avant les suffixes (et réciproquement) peut influencer les résultats. En considérant l'exemple ci-après, sur un texte où la notion de douleur est importante, le fait d'avancer la suppression des préfixes avant les suffixes les mots comme المهم *leur douleur (pour le pluriel)*, المهمما *leur douleur (pour le duel)* exprimeront une toute autre notion.

### 3-3 Travaux sur la catégorisation de textes Arabes

Les études sur la catégorisation des textes en langue Arabe ne sont qu'à leur début. Il n'y a pas encore des travaux avancés qui exploitent ou adaptent les techniques de catégorisation actuelles.

El Kourdi [El Kourdi et al., 2004] utilise l'algorithme Naive Bayes pour la classification des documents. La précision qu'il a obtenu est loin des scores obtenus pour l'Anglais et les langues européennes. Elle n'est que de 68,78%. Sakhr a monté Siraj, un système de classification automatique de documents Arabes. Malheureusement aucun support technique n'est disponible pour ce système. Ses performances ne sont même pas mentionnées. Un autre système Proposé par Sawaf [Sawaf et al., 2001] qui utilise les méthodes de classification telle que le calcul d'entropie maximale pour classer les articles, obtient des scores de précision de l'ordre de 62,70%. Halees [El-Halles, 2006] décrit une méthode basée sur les règles d'association pour le classement des documents avec une précision de 74,41%.

D'autres travaux moins consistants basés sur des corpus montés localement tels que :

- Les travaux réalisés par Siam [Siam et al., 2006], utilisant le light stemming avec le modèle Rocchio où aucune mesure sur la précision n'a été donnée.
- Les travaux réalisés par L. Khreisat [Khreisat, 2006] où on a procédé au calcul de l'effet de la méthode de représentation en n-grammes et le calcul de la dissimilarité entre documents. Elle obtient des résultats très variables d'une classe à une autre allant de 60% à 93%.

### **3-4 Conclusion**

Dans ce chapitre nous avons présenté les difficultés inhérentes au traitement de la langue Arabe à travers la présentation de la structure des textes écrits en cette langue. Nous avons surtout soulevé le degré d'ambiguïté élevé causé par l'absence de voyelles, amplifiée par l'agglutination des mots par rapport à d'autres langues comme le Français ou l'Anglais.

Nous avons montré que bien que la radicalisation soit difficile pour les langues avec des morphologies complexes comme l'Arabe, elle est particulièrement importante et utile en particulier dans les systèmes de recherche d'information. Il est suffisant de regrouper les mots qui se ressemblent le plus sans pour autant connaître leurs racines exactes.

Nous concluons que contrairement à l'Anglais, la langue Arabe possède un système dérivationnel très riche, et c'est dans cette caractéristique que réside la difficulté de traiter cette dernière.



# **Chapitre 4**

## **Techniques utilisées dans la catégorisation de textes**

## Introduction

L'apprentissage automatique (machine learning) est l'outil par excellence pour résoudre les problèmes de classification. En catégorisation de textes, plusieurs algorithmes d'apprentissage ont été adaptés et ont donné de bons résultats. Dans ce chapitre nous parlerons de ces algorithmes en général et on exposera trois d'entre eux.

### 4.1 Algorithmes d'apprentissage et données textuelles

Comme, il a déjà été dit précédemment, les machines ne peuvent traiter directement les données textuelles. C'est pourquoi une préparation de ces données doit être effectuée afin de pouvoir leur appliquer un algorithme d'apprentissage. En catégorisation de textes, cette préparation consiste à produire des données statistiques à partir des données textuelles. C'est la représentation vectorielle TF-IDF qui est généralement adoptée.

Parmi la panoplie de classifieurs existants, on peut faire des regroupements et distinguer des grandes familles. Par exemple, on peut discerner les classifieurs probabilistes qui utilisent l'ensemble d'entraînement, c'est-à-dire les textes déjà classés, pour estimer les paramètres de la distribution de probabilité des mots par rapport aux catégories. C'est dans cette famille qu'on retrouve, entre autres, le classifieur bayésien naïf. On trouve aussi des classifieurs se basant sur un profil, les classifieurs linéaires. Dans ce contexte, le profil est un vecteur de termes pondérés construit pour chaque catégorie, dans le but de les représenter d'une façon générale. Ce vecteur est bien sûr construit à l'aide des données d'entraînement. Quand un nouveau texte doit être classé, il est alors comparé à ce vecteur «*prototype*». Un avantage de cette approche est qu'elle produit un classifieur compréhensible par un humain, dans le sens où le profil de la catégorie peut être interprété assez facilement. Par contre, l'inconvénient principal de tous les classifieurs linéaires est que l'espace est divisé en seulement deux portions, ce qui peut être restrictif, car tous les problèmes ne sont pas nécessairement linéairement séparables. Parmi les nombreux membres de cette famille, nous retrouvons Rocchio, Widrow-Hoff et Les machines à vecteurs supports s'apparentent aux classifieurs linéaires, dans le sens où elles tentent de séparer l'espace en deux, mais certaines manipulations mathématiques les rendent adaptables à des problèmes non linéaires. Il y a aussi une famille de classifieurs qui se basent sur l'exemple. On parle alors d'apprentissage à base d'instances. Les nouveaux textes à classer sont comparés directement aux documents de l'ensemble d'entraînement. L'algorithme des k-voisins les plus proches est sans doute le plus connu de cette famille.

## 4.2 Etude des classifieurs utilisées en catégorisation de textes

Nous nous limiterons à l'étude de trois type de classifieurs. Le modèle le plus ancien étant celui de Rocchio[Rocchio, 1971], il sera étudié à cause de sa simplicité, le classifieur bayésien naïf étant le plus utilisé sera étudié et nous terminerons par les machines à vecteurs supports et on conclura sur l'efficacité de chacun d'eux.

### 4-2.1 Le modèle de Rocchio

La méthode de Rocchio est un classifieur linéaire proposé dans [Rocchio, 1971] pour améliorer les systèmes de recherche documentaire. Ce classifieur s'appuie sur une représentation vectorielle des documents [Salton et Mc Gill, 1983]. Chaque document  $d_j$  est représenté par un vecteur  $d_j$  de  $\mathbb{R}^n$ , où  $n$  est le nombre de termes après sélection et réduction. Chaque coordonnée  $t_{kj}$  se déduit du nombre d'occurrences  $\#(t_k, d_j)$  du terme  $t_k$  dans  $d_j$ , par :

$$\text{TF-IDF}(t_k, d_j) = \#(t_k, d_j) \times \log \frac{|\text{Tr}|}{\#\text{Tr}(t_k)} \quad (4.1)$$

(Cf équation (2.3))

Avec  $|\text{Tr}|$  le nombre de documents du corpus d'apprentissage et  $\#\text{Tr}(t_k)$  le nombre de documents dans lesquels apparaît au moins une fois le terme  $t_k$ . Un terme  $t_k$  se voit donc attribuer un poids d'autant plus fort qu'il apparaît souvent dans le document et rarement dans le corpus complet. Chaque vecteur  $d_j$  est ensuite normalisé, par la normalisation en cosinus, afin de ne pas favoriser les documents les plus longs.

$$t_{kj} = \frac{\text{TF-IDF}(t_k, d_j)}{\sqrt{\sum_{s=1}^{|\text{Tr}|} (\text{TF-IDF}(t_s, d_j))^2}} \quad (4.2)$$

Selon la méthode de Rocchio, pour chaque catégorie  $c_i$ , les coordonnées  $t_{ki}$  du profil prototypique  $c_i = (t_{1i}, \dots, t_{|\text{Tr}|i})$  sont calculé ainsi :

$$t_{ki} = \beta \cdot \sum_{d_j \in \text{POS}_i} \frac{t_{kj}}{|\text{POS}_i|} - \gamma \cdot \sum_{d_j \in \text{NEG}_i} \frac{t_{kj}}{|\text{NEG}_i|} \quad (4.3)$$

avec  $\text{POS}_i = \{d_j \in \text{Tr} \mid \Phi(d_j, c_i) = T\}$ ,  $\text{NEG}_i = \{d_j \in \text{Tr} \mid \Phi(d_j, c_i) = F\}$

$\beta$  et  $\gamma$  sont deux paramètres choisis selon l'importance que l'on accorde aux deux ensembles  $POS_i$  et  $NEG_i$ . [Hull, 1994, Schütze et al., 1995, Dumais et al., 1998, Joachims, 1998] fixent, par exemple, la valeur  $\beta$  à 1 et celle de  $\gamma$  à 0. En règle générale, l'on peut réduire le rôle des exemples négatifs dans la construction de classifieur en choisissant une valeur élevée pour  $\beta$  et une valeur faible pour  $\gamma$ .

[Cohen et Singer, 1999] et [Joachims, 1997] utilisent  $\beta = 16$  et  $\gamma = 4$ . Les profils prototypes  $c_i$  correspondent donc aux barycentres des exemples (avec un coefficient positif pour les exemples de la classe, et négatif pour les autres). Le classement de nouveaux documents s'opère en calculant la distance euclidienne entre la représentation vectorielle du document et celle de chacune des classes ; le document est assigné à la classe la plus proche. La méthode de Rocchio présente deux caractéristiques importantes [Vinot et Yvon, 2002] :

- elle implémente une règle de décision qui dessine des séparations linéaires (hyperplans) dans l'espace de représentation des textes. [Lewis et al., 1996] montre que les performances de Rocchio avec feedback dynamique, sur des tâches de filtrage, sont comparables à celles d'un réseau de neurones entraîné par descente de gradient (Widrow-Hoff). La méthode de Rocchio devrait donc être peu adaptée quand la séparation des classes n'est pas linéaire ;
- chaque exemple contribue identiquement à la construction du centroïde de sa classe. Rocchio s'oppose ainsi d'une part aux algorithmes dirigés par les erreurs (réseaux de neurones, SVM), qui donnent plus d'importance aux exemples mal classés, et d'autre part aux algorithmes locaux, qui n'utilisent qu'une faible partie des exemples à chaque classification (par exemple les K-PPV (K plus proches voisins)).

[Vinot et Yvon, 2002] testent la sensibilité de l'algorithme de Rocchio au *bruit*. Ils ont réalisé des expériences en bruitant peu à peu les étiquettes des classes ; ils concluent que la méthode de Rocchio est exceptionnellement robuste au bruit : même avec 50% des exemples bruités, ses performances sont presque inchangées. Diverses améliorations récentes de ce modèle se sont avérées fructueuses : de nouvelles méthodes de calcul de  $d_j$  ; un choix plus raisonné des exemples négatifs intervenant dans l'équation 4.3 (Query Zoning et feedback dynamique).

[Singhal et al., 1997, Buckley et Salton, 1995] ont ainsi permis d'améliorer sensiblement les performances, le hissant, dans certaines conditions expérimentales, au niveau des meilleurs algorithmes.

[Schapire et al., 1998] concluent que la méthode de Rocchio obtient une performance comparable à celles obtenues par les méthodes les plus sophistiquées, comme celle de boosting, avec un temps d'apprentissage 60 fois plus rapide. Ces résultats vont sans doute renouveler l'intérêt porté à cette méthode ; mais d'autres auteurs tels [Lewis et al., 1996, Joachims, 1998, Cohen et Singer, 1999, Yang et Liu, 1999] considèrent qu'elle est surclassée.

La méthode de Rocchio dans [Vinot et Yvon, 2002] est appliquée aux corpus dont les classes à discriminer sont dotées d'une structure interne, par exemple lorsqu'il existe différents *sous-groupes thématiquement homogènes* au sein d'une même classe. Cette situation se rencontre dans les corpus réels : pour une tâche de filtrage de courrier électronique, les « catégories

courrier valide » et «courrier non-sollicité» recourent en fait des sous-groupes thématiquement très disparates.

[Vinot et Yvon, 2002] montrent que la méthode de Rocchio est particulièrement performante sur les tâches de routage où l'algorithme peut proposer plusieurs classes. Sa simplicité, et la faiblesse d'expressivité qui en découle, ne semblent nuire aux performances, même en présence de sous-classes thématiquement homogènes, sauf si ces thèmes sont trop éparpillés dans les différentes classes.

#### 4-2.2 Le classifieur Bayésien naïf

Comme son nom l'indique, ce classifieur se base sur le théorème de Bayes permettant de calculer les probabilités conditionnelles. Dans un contexte général, ce théorème fournit une façon de calculer la probabilité conditionnelle d'une cause sachant la présence d'un effet, à partir de la probabilité conditionnelle de l'effet sachant la présence de la cause ainsi que des probabilités a priori de la cause et de l'effet. On peut résumer son utilisation lorsqu'il est appliqué à la classification de textes ainsi : on cherche la classification qui maximise la probabilité d'observer les termes du document. Lors de la phase d'entraînement, le classifieur calcule les probabilités qu'un nouveau document appartienne à telle catégorie à partir de la proportion des documents d'entraînement appartenant à cette catégorie. Il calcule aussi la probabilité qu'un terme donné soit présent dans un texte, sachant que ce texte appartient à telle catégorie. Par la suite, quand un nouveau document doit être classé, on calcule les probabilités qu'il appartienne à chacune des catégories à l'aide de la règle de Bayes et des chiffres calculés à l'étape précédente [Mitchell, 1997].

La simplicité, la robustesse et les bonnes performances des classifieurs bayésiens pour la tâche de classification en font aujourd'hui un modèle de référence. Il est intéressant de noter qu'aujourd'hui, de nombreuses applications récentes utilisent un tel modèle (souvent appelé **filtre Bayésien** dans les applications grand public) pour la détection de Spams, le filtrage parental etc..

Plusieurs versions du modèle Naïve Bayes existent. Elles reposent sur des hypothèses statistiques légèrement différentes (notamment en ce qui concerne la longueur des documents). Elles ont été largement étudiées dans [Eyheramendy S., 2003]. Pour la tâche de classification qui est une application classique de ce modèle, les différentes variantes donnent des résultats similaires.

La probabilité à estimer est donc :

$$P(c_j | a_1, a_2, a_3, \dots, a_n) \quad (4.4)$$

où

- $c_j$  est une catégorie
- $a_i$  est un attribut

A l'aide du théorème de Bayes, on obtient :

$$\frac{P(a_1, a_2, a_3, \dots, a_n | c_j)P(c_j)}{P(a_1, a_2, a_3, \dots, a_n)} \quad (4.5)$$

On peut omettre de calculer le dénominateur, qui reste le même pour chaque catégorie.

En guise de simplification, on calcule  $P(a_1, a_2, a_3, \dots, a_n | c_j)$  ainsi :

$$\prod_{i=1}^n P(a_i | c_j) \quad (4.6)$$

On assume, en simplifiant les choses ainsi, que tous les termes sont indépendants sachant la classe du document (c'est à cause de cette simplification qu'on qualifie ce classifieur de naïf). Autrement dit, on assume que la probabilité qu'un terme apparaisse dans un texte est indépendante de la présence des autres termes du texte. On sait que cela est faux. Par exemple, la probabilité de présence du terme «الإعلام» dépend partiellement de la présence du terme «الإلهي». Pourtant, cette supposition n'empêche pas un tel classifieur de présenter des résultats satisfaisants. Et surtout, elle réduit de beaucoup les calculs nécessaires. Sans elle, il faudrait tenir compte de toutes les combinaisons possibles de termes dans un texte, ce qui d'une part impliquerait un nombre important de calculs, mais aussi réduirait la qualité statistique de l'estimation, puisque la fréquence d'apparition de chacune des combinaisons serait très inférieure à la fréquence d'apparition des termes pris séparément.

Pour estimer la probabilité  $P(a_i | c_j)$ , on pourrait calculer directement dans les documents d'entraînement la proportion de ceux appartenant à la classe  $c_j$  qui contiennent le terme  $a_i$ . Cependant, l'estimation ne serait pas très valide pour des numérateurs petits. Dans le cas extrême où un terme ne serait pas du tout rencontré dans une classe, sa probabilité de 0 dominerait les autres dans le produit ci-dessus et rendrait nulle la probabilité globale. Pour pallier ce problème, une bonne façon de faire est d'utiliser le m-estimé qui est calculé ainsi :

$$\frac{n_k + 1}{n + |\text{vocabulaire}|} \quad (4.7)$$

où

- $n_k$  est le nombre d'occurrences du terme dans la classe  $c_j$ ,
- $n$  est le compte total des termes dans le corpus d'entraînement.

### 4-2.3 Les machines à vecteurs supports

Machines à vecteurs supports, Ce nom bien qu'évoquant un outil matériel est loin de l'être. En effet, Le nom machine veut simplement dire algorithme, comme c'est communément utilisé en machine learning. C'est donc une méthode d'apprentissage automatique. Vecteur support est la notion à présenter au niveau de ce paragraphe.

La notion de machines à vecteurs supports à été introduite dans [Vapnik et Cortes, 1995]. L'idée de base soutenant cette notion est la minimisation du risque structurel. C'est-à-dire que l'hypothèse expliquant un ensemble fini d'exemple peut être recherchée dans un sous ensemble de l'ensemble d'apprentissage.

Les SVM conviennent avec les problèmes d'apprentissage à grandes dimensions. C'est le cas, précisément, des documents textuels. Leur utilisation est axée sur une catégorisation binaire (appartenance ou non appartenance à une classe donnée) d'où la notion d'exemple positif et d'exemple négatifs.

- **Cas linéairement séparable**

Un document  $d_i$  est considéré comme un point dans un espace à  $n$  dimensions (les dimensions sont les termes d'indexation). On peut aussi considérer que le document est un vecteur à  $n$  dimensions (ces deux représentations (point et vecteur) peuvent être adoptée selon le contexte).

Dans le cas de la séparabilité linéaire, les individus (documents) positifs sont séparés des individus négatifs par un hyperplan séparateur qu'on notera  $H$ . la notion d'individus positifs (respectivement négatifs) est liée à l'appartenance (respectivement la non appartenance de l'individu pour une classe donnée).

comme le montre la figure 4.1, On note  $H^+$  l'hyperplan parallèle à  $H$  et qui contient l'individu positif le plus proche de  $H$  et  $H^-$  l'individus négatif le plus proche de  $H$ .

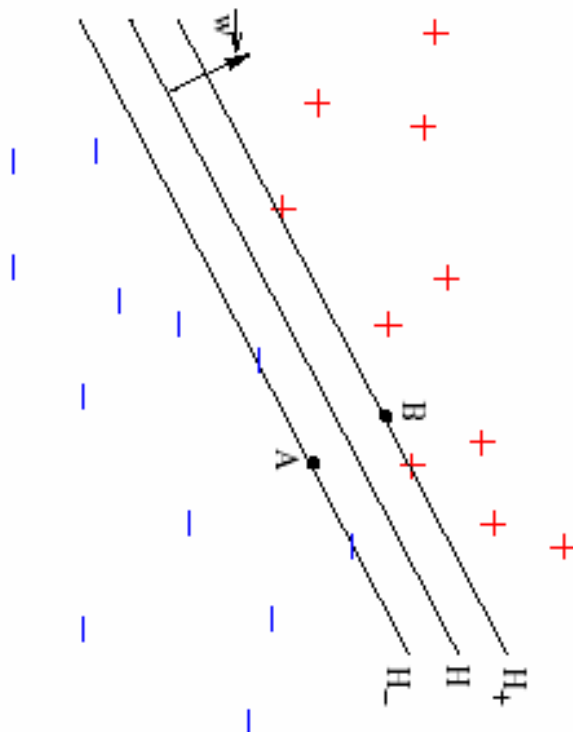


Fig 4.1 Schéma illustratif du principe de construction d'une SVM. A est l'individu négatif le plus proche de  $H$  donc situé sur  $H^-$  et B l'individu positif le plus proche de  $H$  et donc situé sur  $H^+$ .

La théorie d'apprentissage statistique développée par Vapnik en 1998, démontre que nous pouvons définir un hyperplan (relatif à l'ensemble d'apprentissage) et possédant deux propriétés essentielles :

- il est unique pour chacun des ensembles de données linéairement séparables,
- le risque de sur-apprentissage est le plus petit qui soit relativement à n'importe quel autre hyperplan séparateur.

Nous définissons, la marge du classifieur comme la distance séparant l'hyperplan et les exemples les plus proches.

L'hyperplan optimal est celui qui possède la plus grande marge. Comme le démontre la figure 4.2 ci-après.

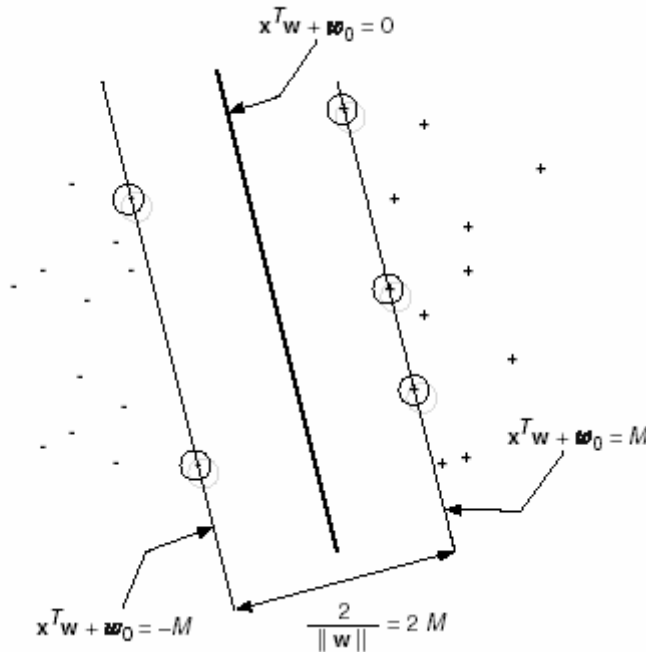


Fig 4.2 : Illustration de l'hyperplan séparateur et de la marge (notée M dans la figure) ; les points encerclés représentent les vecteurs supports

Pour calculer cette marge. Nous procédons comme suit :  
Un hyperplan a pour équation :

$$y = \langle \vec{w}, \vec{d} \rangle + b \quad (4.8)$$

$\langle \vec{w}, \vec{d} \rangle$  dénote le produit scalaire entre les vecteur  $\vec{w}$  et  $\vec{d}$ . Pour un individu  $\vec{d}$  de classe  $y$ .  
on cherche  $\vec{w}$  tel que

$$\begin{cases} \langle \vec{w}, \vec{d} \rangle + b \geq 1 & \text{si } y = +1 \\ \langle \vec{w}, \vec{d} \rangle + b \leq -1 & \text{si } y = -1 \end{cases} \quad (4.9)$$

donc on a :

$$y(\langle \vec{w}, \vec{d} \rangle + b) - 1 \geq 0 \quad (4.10)$$

On veut maximiser la largeur de la marge



Le vecteur  $\vec{w}$  est perpendiculaire à l'hyperplan H (vecteur normal de H)

Soit B un point de  $H^+$  et A le point le plus proche de B sur  $H^-$

Pour tout point O de l'espace, nous avons :  $\vec{OB} = \vec{OA} + \vec{AB}$

Par définition des points A et B,  $\vec{AB}$  est parallèle à  $\vec{w}$ , donc il existe un  $\lambda \in \mathfrak{R}$  tel que

$\vec{AB} = \lambda \vec{w}$  soit  $\vec{OB} = \vec{OA} + \lambda \vec{w}$

Nous voulons que A, B,  $H^-$  et  $H^+$  soient tels que :

$$\begin{cases} B \in H^+ \Rightarrow \langle \vec{w}, \vec{OB} \rangle + b = 1 \\ A \in H^- \Rightarrow \langle \vec{w}, \vec{OA} \rangle + b = -1 \end{cases}$$

donc  $\langle \vec{w}, \vec{OA} + \lambda \vec{w} \rangle + b = 1$

donc,  $\underbrace{\langle \vec{w}, \vec{OA} \rangle + b}_{-1} + \langle \vec{w}, \lambda \vec{w} \rangle = 1$

soit  $-1 + \underbrace{\langle \vec{w}, \lambda \vec{w} \rangle}_{\lambda \langle \vec{w}, \vec{w} \rangle} = 1$

donc  $\lambda = \frac{2}{\langle \vec{w}, \vec{w} \rangle} = \frac{2}{\|\vec{w}\|^2}$

donc :

$$\lambda = \frac{2}{\|\vec{w}\|^2} \tag{4.11}$$

La largeur de ma marge est  $\|\lambda \vec{w}\|$ ,  $\vec{w}$  est sa direction et  $\lambda$  son amplitude.

On veut maximiser la marge, on doit alors minimiser la norme de  $\vec{w}$

Minimiser  $\vec{w}$  c'est la même chose que minimiser  $\|\vec{w}\|^2$

On veut aussi vérifier les contraintes :

$$\gamma_i = y_i (\langle \vec{w}, \vec{d}_i \rangle + b) - 1 \geq 0 \quad \forall i \in \{1..n\} \tag{4.12}$$

C'est donc, un problème d'optimisation non linéaire (minimiser  $\|\vec{w}\|^2$  sous les contraintes  $\gamma_i$  qu'on résout par la méthode de Lagrange.

Cette méthode transforme un problème d'optimisation de fonctions avec contraintes en un problème d'optimisation de fonctions sans contraintes et les deux problèmes possèdent les mêmes solutions.

On utilise pour cela un opérateur appelé le lagrangien et noté  $L_P$  comme somme de fonctions à optimiser (fonction objectif) et l'opposé de chaque contrainte  $\gamma_i$  multipliée par une constante  $\alpha_i \in \mathfrak{R}^+$ . Les  $\alpha_i$  constituent les multiplicateurs de Lagrange.

Nous avons donc :

$$L_P = \frac{1}{2} \|\vec{w}\|^2 - \alpha_i \left( \sum_{i=1}^n y_i (\langle \vec{w}, \vec{d}_i \rangle + b) - 1 \right) \quad (4.13)$$

qui est équivalent à :

$$L_P = \frac{1}{2} \|\vec{w}\|^2 - \sum_{i=1}^n \alpha_i y_i (\langle \vec{w}, \vec{d}_i \rangle + b) + \sum_{i=1}^n \alpha_i \quad (4.14)$$

$L_P$  doit être minimisé par rapport à  $\vec{w}$  et  $b$ , et il faut que les dérivées par rapport à  $\alpha_i$  soient nulles.

Le gradient de  $L_P$  devant être nul par rapport à  $\vec{w}$  et  $b$  on écrit :

$$\begin{cases} \frac{\partial L_P}{\partial \vec{w}} = 0 \\ \frac{\partial L_P}{\partial b} = 0 \end{cases} \text{ d'où } \begin{cases} \vec{w} = \sum_{i=1}^n \alpha_i y_i \vec{d}_i \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases} \quad (4.15)$$

De la formulation de  $L_P$  et de ces deux équations, on tire la formulation duale du lagrangien en éliminant  $\vec{w}$  :

$$L_D = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \vec{d}_i, \vec{d}_j \rangle \quad (4.16)$$

qui doit être maximisé. Le maximum de  $L_D$  et le minimum de  $L_P$  sont obtenus pour les mêmes valeurs de  $\vec{w}$ ,  $b$  et  $\alpha_i$ .

La résolution de ce problème d'optimisation s'obtient avec des outils standards.

Pour que  $\vec{w}$ ,  $b$  et  $\alpha_i$  existent, le problème doit vérifier les conditions de Karush-Kuhn-Tucker (KKT) :

$$\left\{ \begin{array}{l} \frac{\partial L_P}{\partial \vec{w}_v} = \vec{w}_v - \sum_{i=1}^n \alpha_i y_i \vec{d}_{i,v} = 0 \quad \forall v = 1, 2, \dots, P \\ \frac{\partial L_P}{\partial b} = \sum_{i=1}^n \alpha_i y_i = 0 \\ y_i (\langle \vec{w}, \vec{d}_i \rangle + b) - 1 \geq 0 \quad \forall i = 1, 2, \dots, N \\ \alpha_i \geq 0 \quad \forall i = 1, \dots, N \\ \alpha_i (y_i (\langle \vec{w}, \vec{d}_i \rangle + b) - 1) = 0 \quad \forall i = 1, \dots, N \end{array} \right. \quad (4.17)$$

Ces cinq conditions résument tout ce qui a été dit précédemment. On remarque alors que les conditions KKT sont vérifiées. Ce qui démontre que le problème possède bien une solution

La dernière ligne peut être interprétée par :

soit  $\vec{\alpha}_i = 0$ ,

soit  $y_i (\langle \vec{w}, \vec{d}_i \rangle + b) - 1 = 0$

Les points ayant  $\vec{\alpha}_i > 0$  doivent avoir  $y_i (\langle \vec{w}, \vec{d}_i \rangle + b) - 1 = 0$ , c'est-à-dire les vecteurs supports.

Nous définissons, alors, qu'un vecteur support est un vecteur dont le multiplicateur de Lagrange associé est non nul.

Quelques remarques :

1. Les multiplicateurs de Lagrange étant positifs dans le problème posé ici, un vecteur support a donc un multiplicateur de Lagrange de valeur strictement positive,

2. Puisque l'on a  $\alpha_i (y_i (\langle \vec{w}, \vec{d}_i \rangle + b) - 1) = 0$  pour tous les points et que  $\vec{\alpha}_i \neq 0$  pour les vecteurs supports, cela entraîne que  $y_i (\langle \vec{w}, \vec{d}_i \rangle + b) - 1 = 0$  pour les vecteurs supports ; cela entraîne que les vecteurs supports sont situés exactement sur les hyperplans  $H^+$  et  $H^-$ ,

3. En fait, seuls les exemples correspondants aux vecteurs supports sont réellement utiles dans l'apprentissage. Si on les connaissait a priori, on pourrait effectuer l'apprentissage sans tenir compte des autres exemples,
4. Les vecteurs supports synthétisent en quelques sortes les aspects importants du jeu d'exemple. On peut donc compresser l'ensemble des exemples en ne retenant que les vecteurs supports.

### Classification d'une nouvelle donnée.

La classe d'une donnée  $\vec{d}$  est  $\pm 1$  et elle est fournie par le signe de  $\langle \vec{w}, \vec{d} \rangle + b$ .

En effet, si cette quantité est  $\geq 1$ , cela signifie que  $x$  est « au dessus » de  $H^+$ . sinon,  $\langle \vec{w}, \vec{d} \rangle + b \leq -1$ , ce qui signifie que  $x$  est « en dessous » de  $H^-$ . en utilisant la fonction signe  $\text{sgn}(\cdot)$ , on note  $\text{sgn}(\langle \vec{w}, \vec{d} \rangle + b)$  cette quantité.

Puisque  $\vec{w} = \sum_{i=1}^n \alpha_i y_i \vec{d}_i$  et que seuls les vecteurs supports ont un multiplicateur de Lagrange non nul, on a :

$$\text{sgn}(\langle \vec{w}, \vec{d} \rangle + b) = \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i \langle \vec{d}_i, \vec{d} \rangle + b\right) = \text{sgn}\left(\sum_{j=1}^{n_s} \alpha_j y_j \langle \vec{s}_j, \vec{d} \rangle + b\right) \quad (4.18)$$

où  $\vec{d}$  est l'instance à classer et  $\vec{d}_i$  sont les exemples d'apprentissage. Les  $\vec{s}_j$  sont les vecteurs supports.  $n_s$  est le nombre de ces vecteurs.

#### ▪ Cas linéairement non séparable

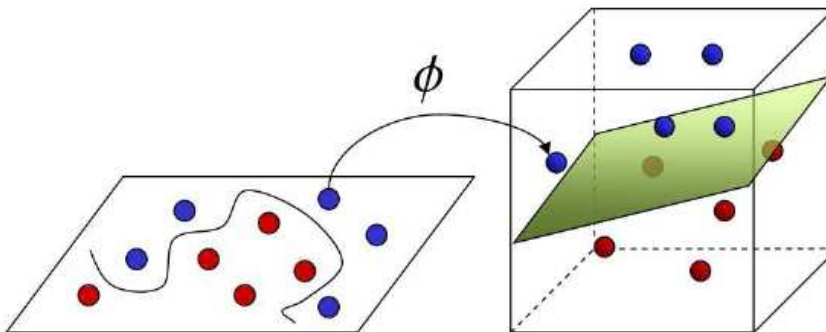


Fig 4.3 : Transformation des données de l'espace initial vers un espace de plus grande dimension pour que les données deviennent linéairement séparables

Dans le cas de données non linéairement séparables (cf. figure 4.3) l'idée est de trouver une transformation de l'espace de données dans un autre espace où les données seront à nouveau linéairement séparables. En général, ce nouvel espace de représentation est d'une dimension plus grande que l'initial. Il peut même être de dimension infinie. Le théorème de Cover [1965] indique qu'un ensemble de données transformé de manière non linéaire dans un espace de plus grande dimension a plus de chance d'être linéairement séparable que dans son espace d'origine.

### Construction d'une SVM non linéaire :

Soit  $\Phi$  la fonction qui associe l'espace de représentation initial noté  $D$  ( $\mathcal{R}^P$  :  $P$  étant la dimension de cet espace) un autre espace de représentation noté  $F$ . on note cette association :  $\Phi : \mathcal{R}^P \rightarrow F$

On procède à une transformation de l'espace initial de données dans un autre espace de caractéristiques (feature space). Chacun des axes de coordonnées de  $F$  est une combinaison non linéaire des axes de coordonnées de  $D$ . On introduit ce qu'on appelle une fonction noyau (kernel function) notée  $K$  telle que :

$$K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$$

On peut alors effectuer les calculs comme dans le cas des données linéairement séparables sans connaître explicitement la fonction  $\Phi$ , donc sans devoir transformer les données par cette fonction. Lors de l'optimisation du problème quadratique on remplace le produit scalaire

$$\langle \vec{x}_i, \vec{x}_j \rangle \text{ par } K(x_i, x_j)$$

Le dual du lagrangien qu'on doit maximiser devient alors

$$L_D = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (4.19)$$

Lors de la classification d'une nouvelle donnée  $\vec{X}$  on calcule :

$$\text{sgn}(\langle \vec{w}, \vec{x} \rangle + b) = \text{sgn}(\sum_{i=1}^n \alpha_i y_i (s_i) K(s_i, x) + b) \quad (4.20)$$

On remarque bien que rien n'est différent des SVM dans le cas des données linéairement séparables.

### Fonction noyau à utiliser :

Peut-on utiliser n'importe quelle fonction noyau ?

Les fonctions noyaux acceptables doivent satisfaire à la condition de Mercer qui dit qu'une fonction noyau  $K(x,y)$  doit respecter la contrainte suivante :

Pour toute fonction  $g(x)$  telle que  $\int g(x)^2 dx$  est finie on a  $\iint K(x, y)g(x)g(y)dxdy \geq 0$

Le respect de cette condition assure que le problème quadratique possède bien une solution.

On connaît certaines fonctions noyaux satisfaisant à cette condition. On peut citer :

- Le noyau polynomial :  $K(x, y) = (\langle \vec{x}, \vec{y} \rangle + c)^m$ ,  $m \in \mathbb{N}$  et  $c > 0$ ,

- Le noyau gaussien :  $K(x, y) = e^{-\frac{\|\vec{x}-\vec{y}\|^2}{2\sigma^2}}$ ,

- Noyau neuronal/sigmoïde :  $K(x, y) = \tanh(k \langle \vec{x}, \vec{y} \rangle - \delta)$  où  $k > 0$  et  $\delta > 0$ .

**Remarques** : cette fonction ne respecte pas la condition de Mercer pour toutes les valeurs de  $k$  et  $\delta$ .

Toute combinaison linéaire de fonctions noyaux acceptables est une fonction noyau acceptable.

On ne sait pas, face à un problème donné, quelle fonction noyau choisir.

## 4-3 Conclusion : Choix du meilleur algorithme de classification de

En catégorisation de texte, l'espace de représentation étant de grande dimension (des milliers de documents contre des centaines de milliers d'attributs voire des millions d'attributs), cette contrainte impose l'utilisation d'un algorithme efficace. Devant cette grande dimensionnalité beaucoup d'algorithmes s'avèrent inefficaces. Quelle est donc la meilleure méthode pour la catégorisation de textes?

Plusieurs méthodologies peuvent être utilisées pour répondre à cette question.

La première consiste à comparer les différentes méthodes mises en œuvre sur un même corpus d'apprentissage. C'est d'ailleurs celle-ci qui est la plus généralement utilisée dans les conférences internationales telle que TREC. L'inconvénient c'est que les auteurs doivent toujours utiliser le même découpage du corpus. Ce qui n'est pas toujours le cas. Ces différences de découpage du corpus rend impossible la comparaison des méthodes entre elles. (pour le corpus Reuters-21578 certains auteurs considèrent 90 catégories d'autres considèrent 118 catégories et même la base de test diffère d'un auteur à un autre).

Une autre approche est l'utilisation de plusieurs méthodes par un même auteur comme ça le découpage et les mesures d'efficacité sont les mêmes pour toutes les expérimentations. [Yang et Liu, 1999] comparent ainsi les machines à vecteurs supports, les plus proches voisins, les réseaux de neurones, une combinaison linéaire, et des réseaux bayésiens. [Dumais *et al.*, 1998] proposent également une série de comparaisons en mettant en compétition une variante de l'algorithme de Rocchio (appelée *find similar*), des arbres de décision, des réseaux bayésiens et des machines à vecteurs supports. Le problème vient du fait que toutes ces

méthodes sont délicates à mettre en oeuvre et leurs performances dépendent fortement des algorithmes utilisés.

Par exemple, l'implémentation des machines à vecteurs supports proposées par [Dumais *et al.*, 1998] obtient de nettement meilleurs résultats que celle proposée par [Joachims, 1998]. Les réseaux de neurones testés par [Yang et Liu, 1999] sont des perceptrons multicouche avec une couche cachée comportant 64 neurones, 1000 descripteurs en entrées et 90 neurones de sorties correspondant aux 90 catégories ; ils considèrent un seul réseau pour l'ensemble des catégories comportant plus de 64000 poids (l'algorithme d'apprentissage n'est pas précisé). Il n'est pas surprenant, dans ces conditions, que les performances obtenues ne soient pas très bonnes : de telles démarches jugent plus la capacité des auteurs à mettre en oeuvre des méthodes, que les capacités des méthodes elles-mêmes.

L'algorithme de Rocchio est considéré comme un algorithme ancien, mais [Schapire *et al.*, 1998] ont montré que cet algorithme obtient d'excellents résultats pour la catégorisation de textes à condition d'utiliser un codage efficace, de bien choisir les documents non pertinents, et d'effectuer une optimisation des poids. Leurs conclusions vont à l'encontre d'autres comparaisons qui montrent que cet algorithme n'est pas performant par rapport aux méthodes fondées sur l'apprentissage numérique [Schütze *et al.*, 1995] [Lewis *et al.*, 1996] [Cohen et Singer, 1996]. Ces différentes remarques prouvent que le succès d'une méthode dépend d'un ensemble de paramètres qui vont du codage des documents au choix des algorithmes et de leur utilisation, et qu'il est, par conséquent, extrêmement difficile de tirer des conclusions définitives sur une approche.

Cependant, il est à noter que les machines à vecteurs support ont démontré leur nette supériorité par rapport aux autres méthodes grâce à leur robustesse et leur insensibilité aux bruits.

# **Partie II**

# **Expérimentation**



# **Chapitre 5**

## **Corpus utilisé et processus de prétraitement proposé**

## 5-1 Le corpus

La catégorisation de textes est un processus long et complexe. Il est basé sur l'apprentissage dont la ressource primordiale est une base de données d'apprentissage. Dans ce domaine cette base s'avère une collection de textes étiquetés manuellement qu'on appelle corpus.

Les corpus sont une ressource importante aussi bien pour l'enseignement que pour la recherche. La langue Arabe souffre énormément du manque de cette ressource.

L'expérimentation que nous avons menée est basée sur un corpus diffusé sur le Web par Latifa Sulaiti [Al-Sulaiti, L. et Atwell, E., 2004] en langue Arabe baptisés par ses auteurs (CCA : Corpus of contemporary arabic)

Il existe cependant des corpus plus élaborés contenant plusieurs milliers de textes et quelques millions de mots édités par LDC (Language Data Consortium : Université de Pennsylvanie) et ELRA ( European language ressources association ). Les corpus les plus populaires sont ceux édités par l'AFP (Agence France Presse) sous forme de dépêches et les archives des journaux quotidiens tels que El Hayat qui sont connu pour la publication annuelle sous forme de CD-ROM de textes archivés.

Voici donc, un certain nombre de références où l'on peut se procurer des corpus de textes Arabes :

<b>Nom du corpus</b>	<b>Source</b>	<b>Forme</b>	<b>Taille</b>	<b>Utilisation</b>	<b>Origine</b>
<a href="#">Buckwalter Arabic Corpus</a> (1986-2003)	Tim Buckwalter	Written	2.5 to 3 billion words	Lexicography	Public resources on the Web
<a href="#">Leuven Corpus</a> (1990-2004)	Catholic University Leuven, Belgium	Written and spoken	3M words (spoken: 700,000)	Arabic-Dutch /Dutch-Arabic learner's dictionary	Internet sources, radio & TV, primary school books
<a href="#">Arabic Newswire Corpus</a> (1994)	University of Pennsylvania LDC	Written	80M words	Education and the development of technology	Agence France Presse, Xinhua News Agency, and Umma Press
<a href="#">CALLFRIEND Corpus</a> (1995)	University of Pennsylvania LDC	Conversational	60 telephone conversations	Development of language identification technology	Egyptian native speakers
<a href="#">NijmegenCorpus</a> (1996)	Nijmegen University	Written	Over 2M words	Arabic-Dutch / Dutch-Arabic dictionary	Magazines and fiction
<a href="#">CALLHOME Corpus</a> (1997)	University of Pennsylvania LDC	Conversational	120 telephone conversations	Speech recognition produced from telephone lines	Egyptian native speakers
CLARA (1997)	Charles University, Prague	Written	50M words	Lexicographic purposes	Periodicals, books, internet sources from 1975-present
<a href="#">Egypt</a> (1999)	John Hopkins University	Written	Unknown	MT	A parallel corpus of the Qur'an in

					English and Arabic
Broadcast News Speech (2000)	University of Pennsylvania LDC	Spoken	More than 110 broadcasts	Speech recognition	News broadcast from the radio of voice of America.
<a href="#">DINAR Corpus</a> (2000)	Nijmegen Univ., SOTETEL T, co-ordination of Lyon2 Univ	Written	10M words	Lexicography, general research, NLP	Unknown
<a href="#">An-Nahar Corpus</a> (2001)	ELRA	Written	140M words	General research	An-Nahar newspaper (Lebanon)
<a href="#">Al-Hayat Corpus</a> (2002)	ELRA	Written	18.6M words	Language Engineering and Information Retrieval	Al-Hayat newspaper (Lebanon)
<a href="#">Arabic Gigaword</a> (2002)	University of Pennsylvania LDC	Written	Around 400M	Natural language processing, information retrieval, language modelling	Agence France Presse, Al-Hayat news agency, An-Nahar news agency, Xinhua news agency
E-A Parallel Corpus (2003)	University of Kuwait	Written	3M words	Teaching translation & lexicography	Publications from Kuwait National Council
General Scientific Arabic Corpus (2004)	UMIST, UK	Written	1.6M words	Investigating Arabic compounds	<a href="http://www.kisr.edu.kw/science/">http://www.kisr.edu.kw/science/</a>
Classical Arabic Corpus (CAC) (2004)	UMIST, UK	Written	5M words	Lexical analysis research	<a href="http://www.muhammadith.org">www.muhammadith.org</a> and <a href="http://www.alwarag.com">www.alwarag.com</a>
Multilingual Corpus 2004	UMIST, UK	Written	11.5M words (Arabic 2.5M)	Translation	IT-specialized websites-computer system and online software help-one book
SOTETEL Corpus	SOTETEL-IT, Tunisia	Written	8M words	Lexicography	Literature, academic and journalistic material
Corpus of Contemporary Arabic (CCA) 2004	University of Leeds	Written and spoken	Around 1M words	TAFL and information retrieval	Websites and online magazines
<a href="#">DARPA Babylon Levantine Arabic Speech and Transcripts</a> (2005)	University of Pennsylvania LDC	Spoken	About 2000 telephone calls	Machine translation, speech recognition & spoken dialogue system	Fisher style telephone speech collection

Tableau 5.1 : Les corpus disponibles pour l'Arabe (source : page web de Latifa AL sulaiti)

Il existe aussi des moyens automatiques pour construire des corpus. Andrew Roberts de l'université de Leeds a réécrit en Java un moyen originellement baptisé BootCat par Marco Baroni et qu'il a baptisé JBootCat pour la génération automatique de corpus sur le web. Cependant la construction manuelle de corpus reste de loin la plus préférée. Elle est cependant coûteuse en temps et en moyens.

Le CCA possède les caractéristiques ci-après :

	<i>Classe</i>	<i>Nbre de textes</i>	<i>Nbre de mots</i>
1.	Short stories	31	45,460
2.	Education	10	25,574
3.	Religion	19	111,199
4.	Autobiography	73	153,459
5.	Sociology	30	85,688
6.	Tourist/travel	60	46,093
7.	Recipes	9	4,973
8.	Science	45	104,795
9.	Sports	4	8,290
10.	Economics	29	67,478
11.	Children's stories	27	21,958
12.	Health and medicine	32	40,480
13.	Interviews	23	58,408
14.	Politics	10	46,291
	<b>Total</b>	<b>402</b>	<b>820,146</b>

Tableau 5.2 : Caractéristiques de CCA

## 5-2 Processus d'expérimentations

Dans le processus d'expérimentations que nous avons menées, nous avons adopté la démarche détaillée dans la figure 5.1. [Djelailia K. et al., 2007]

Elle consiste en le tokenisation et l'indexation des textes bruts et leur normalisation puis nous les avons transformés en format translittéré.

Nous avons créé pour la comparaison des résultats une version radicalisée en utilisant un algorithme de stemming.

Dans tous les cas de figure nous avons opté pour l'élimination des mots outils (stop words). Nous avons par la suite monté nos expérimentations que nous détaillerons au chapitre 6.

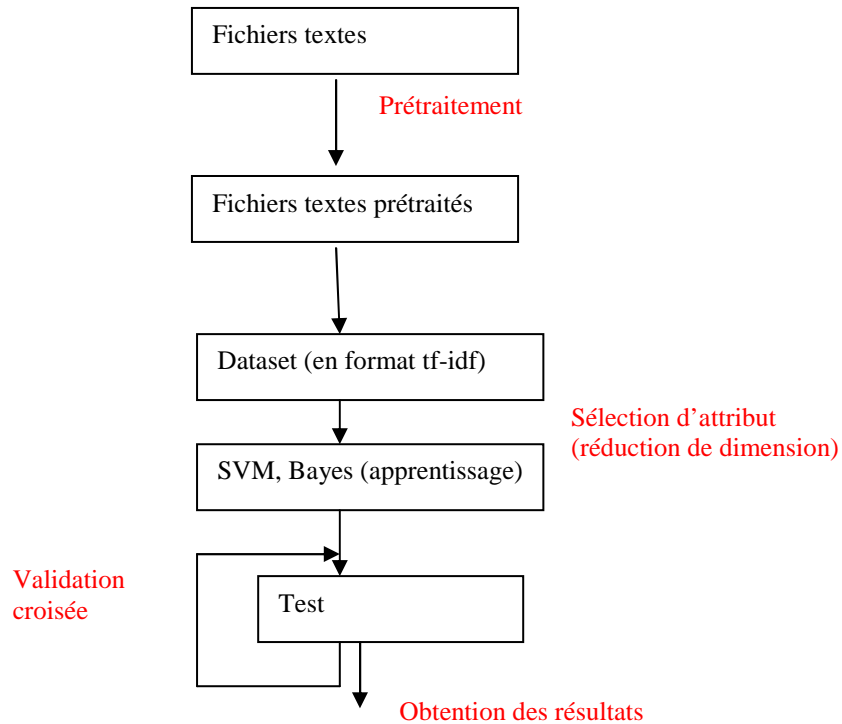


Fig 5.1 : Processus d'expérimentation

## 5-3 Le prétraitement

Pour la catégorisation de textes, un prétraitement du corpus utilisé est exigé en vue de les convertir sous la forme exigée pour un traitement statistique.

### 5-3.1 Obtention de textes bruts

Le corpus fourni est écrit en XML. Le premier traitement à effectuer étant de transformer les textes XML en textes brut, la cause en est simple ; les balises XML n'apportent que du bruit et ne font pas partie des textes.

### 5-3.2 Tokenisation et indexation

On cherche à transformer les textes de telle sorte qu'on puisse identifier aisément les termes qui le composent. Ceci est effectué en identifiant les délimiteurs de termes et de mettre chaque terme sur une ligne du fichier texte transformé tout en éliminant les doublants (indexation)

Exemple de tokenisation et indexation (cf. Tableau 5.3)

Texte non tokenisé	Texte tokenisé et indexé
الجزائر بلد شاسع، ذو تضاريس متنوعة و مناخ يغلب عليه الطابع القاري. تضاريس الجزائر تتكون من سهول وجبال وصحاري	الجزائر بلد شاسع ذو تضاريس متنوعة ومناخ يغلب عليه الطابع القاري تتكون من سهول وجبال وصحاري

Tableau 5.3 : exemple de tokénisation

Quand on procède à la tokenisation, il est souhaitable de faire une passe pour détecter les anomalies de la ponctuation. Il est par exemple ambigu de distinguer, la virgule qui fait parfois usage de séparateurs de décimales comme de séparateurs de milliers. Le point peut être utilisé comme zéro. On procède ensuite à l'élimination des signes diacritiques (les voyelles faibles) quoique la plupart des corpus n'en contient pas mais exceptionnellement ça peut exister ainsi que les caractères d'allongement qui sont souvent utilisés en langue Arabe et on considère le reste comme mot. Les chaînes vides sont écartées.

### 5-3.3 La normalisation

En vue de maîtriser la variation dans la façon de représenter les textes en Arabe, on procède à différentes sortes de normalisation [Larkey et al., 2002].

La normalisation consiste en :

- élimination de la ponctuation,
- éliminer tous les caractères qui ne sont pas des lettres,
- remplacer le ِ initial ou ُ par َ,
- remplacer َ par ِ,
- remplacer la séquence ىء par ى,
- remplacer ى final par ى,
- remplacer ة finale par ة.

### 5-3.4 Translittération

La translittération de l'Arabe est la représentation des caractères de l'Arabe standard moderne en utilisant seulement les caractères ASCII 7 bits (c'est-à-dire les caractères latins). Ceci est utilisé dans le cas où l'affichage et le traitement de l'Arabe sont impraticables ou strictement impossibles. C'est le cas notamment de certains langages de programmation. C'est pour cette cause que nous avons optée pour la translittération du corpus que nous avons utilisé. La

translittération utilisée est celle proposée par Tim buckwalter [Buckwalter] avec quelques modifications dictées par le souci de normalisation.

Les correspondances Arabes - translittération Buckwalter sont présentées dans le tableau 5.4

N°	Nom (Unicode name)	Buckwalter	Glyph
1.	hamza-sur-la-line (Lettre Arabe hamza)	'	ء
2.	Madda (Lettre Arabe alef avec madda dessus)		آ
3.	hamza-sur-'alif (Lettre Arabe aleph avec hamza au dessus)	>	أ
4.	hamza-sur-waaw (Lettre Arabe waw avec hamza au dessus)	&	ؤ
5.	hamza-sous -'alif (Lettre Arabe aleph avec hamza au dessous)	<	إ
6.	hamza-sur -yaa' (Lettre Arabe yeh avec hamza au dessus )	}	ئ
7.	bare 'alif (Lettre Arabe alef)	A	ا
8.	baa' (Lettre Arabe beh)	b	ب
9.	taa' marbuuTa (Lettre Arabe teh marbuta)	p	ة
10.	taa' (Lettre Arabe teh)	t	ت
11.	thaa' (Lettre Arabe theh)	v	ث
12.	Jiim (Lettre Arabe jeem)	j	ج
13.	Haa' (Lettre Arabe hah)	H	ح
14.	khaa' (Lettre Arabe khah)	x	خ
15.	Daal (Lettre Arabe dal)	d	د
16.	Dhaal (Lettre Arabe thal)	*	ذ
17.	raa' (Lettre Arabe reh)	r	ر
18.	Zaay (Lettre Arabe zain)	z	ز
19.	Siin (Lettre Arabe seen)	s	س
20.	Shiin (Lettre Arabe sheen)	\$	ش
21.	Saad (Lettre Arabe sad)	S	ص
22.	Daad (Lettre Arabe dad)	D	ض
23.	Taa' (Lettre Arabe tah)	T	ط
24.	Zaa' (DHaa') (Lettre Arabe zah)	Z	ظ
25.	Ayn (Lettre Arabe ain)	E	ع
26.	Ghain (Lettre Arabe ghain)	g	غ
27.	taTwiil (Lettre Arabe tatweel)	_	-
28.	faa' (Lettre Arabe feh)	f	ف
29.	Qaaf (Lettre Arabe qaf)	q	ق
30.	Kaaf (Lettre Arabe kaf)	k	ك
31.	Laam (Lettre Arabe lam)	l	ل
32.	Miim (Lettre Arabe meem)	m	م
33.	Nuun (Lettre Arabe noon)	n	ن
34.	haa' (Lettre Arabe heh)	h	ه
35.	Waaw (Lettre Arabe waw)	w	و
36.	'alif maqSuura (Lettre Arabe alef maksura)	Y	ى
37.	yaa' (Lettre Arabe yeh)	y	ي
38.	fatHatayn (Arabe fathatan)	F	َ
39.	Dammatayn (Arabe dammatan)	N	ِ
40.	Kasratayn (Arabe kasratan)	K	ِ
41.	fatHa (Arabe fatha)	a	َ

42.	Damma (Arabe damma)	u	◌ُ
43.	Kasra (Arabe kasra)	i	◌ِ
44.	Shaddah (Arabe shadda)	~	◌ّ
45.	Sukuun (Arabe sukun)	o	◌ْ
46.	dagger 'alif (Lettre Arabe superscript alef)	`	◌ْ◌ْ
47.	waSla-sur-alif (Lettre Arabe alef wasla)	{	◌ْ◌ْ

Tableau 5.4 : système de translittération proposé par Tim Buckwalter

Les différences entre la translittération utilisée [Darwish, 2002] et celle de Buckwalter [Buckwalter] sont :

- toutes les formes de Hamza sont remplacées par A (Lignes 1 à 7),
- la lettre ّ est remplacée par O (puisque le O n'est utilisée nulle part dans la translittération Buckwalter),
- la lettre ش est remplacée par P (puisque le P n'est utilisée nulle part dans la translittération Buckwalter).

On remarque que la translittération utilisée n'utilise que des lettres minuscules ou majuscules et non des caractères spéciaux.

### 5-3.5 Radicalisation ou stemming

La radicalisation est l'un des moyens offerts dans le domaine de la recherche d'information qui a contribué à la normalisation dans tous processus de recherche d'information. On utilisera dans ce qui suit le vocable « stemmer » pour désigner tout moyen de radicalisation. Dans notre travail on utilise le terme stemming pour se référer à tout processus tendant à remplacer les différentes forme d'un mot par un représentant de sa classe.

Toutes les méthodes proposées dans ce domaine sont divisées en deux classes : Elimination des affixes et le stemming statistique.

**Les méthodes basées sur l'élimination des affixes :** Elles ont été testées sur plusieurs langues et on a constaté que dans un processus de recherche d'information, elles contribuent à augmenter le rappel mais la précision se dégrade un petit peu [L. Larkey et al. 2002 ]. Dans un système basé ranknig, les meilleurs documents peuvent ne pas être mis en début de liste.

**Le stemming statistique :** dans cette classe d'outils, on utilise en général des mesures de similarité entre chaînes de caractères. La méthode la plus utilisée est celle basée sur le classement d'un mot dans la classe d'un n-gramme si le mot contient le n-gramme spécifié ou une partie de ce n-gramme.

Les caractéristiques de la langue Arabe détaillées au chapitre 4 font que l'Arabe est une langue très difficile à radicaliser. Pour cette langue plusieurs méthodes ont été testées. Dans notre étude nous avons opté pour celle basée sur l'élimination des affixes. Dans une étude publiée par Leah larkey [Larkey et al. 2002], il a été prouvé que le light stemming (Une méthode de dessuffixation) dépasse en tout point de vue les autres méthodes.



### Principe:

1. Eliminer و si le reste est composé de 3 lettres ou plus. Néanmoins l'élimination de و reste problématique car plusieurs mots peuvent commencer par cette lettre. Il est à remarquer qu'on s'est restreint à l'éliminer seulement quand il est attaché à l'article défini ال. ,
2. Eliminer tous les articles définis si le mot restant est composé de plus de 2 caractères,
3. Balayer la liste des suffixes indiquée ci-dessous et éliminer les caractères de fin du mot s'ils se trouvent dans la liste à condition que la longueur de la chaîne restante dépasse deux caractères.

Le tableau 5.5 présente la liste des affixes pour Light 10 et Al-stem

Stemmer	Préfixes	Suffixes
Light10 de Leah Larkey	ال، وال، بال، كال، فال، لل، و	ها، ان، ات، ون، ين، يه، ية، ه، ة، ي
Al-stem de kareem Darwish	وال، فال، باء، بت، يت، لت، مت، وت، ست، نت، بم، لم، وم، كم، فم، ال، لل، وي، لب، فيب، وا، فا، لا، با	ات، وا، ون، وه، ان، تي، ته، تم، كم، هم، هن، ها، ية، تك، نا، ين، به، ة، ه، ي، ا

Tableau 5.5. : Les affixes considérés dans le stemming (Light 10 et Al-stem)

Nous avons opté pour al-stem de Kareem Darwish [Darwish, 2002] à cause du taux de compression qui dépasse 60% en moyenne.

### 5-3.6 Elimination des mots outils

Comme il a été dit précédemment ( cf. loi de Zipf), certains mots dont l'apport informationnel ou plutôt dont le pouvoir discriminant est faible sont éliminés. Il s'agit des mots outils (stop words), qu'on trouve dans toutes les langues. La langue Arabe en compte 168.

Nous avons utilisé une liste de 131 mots outils que nous avons éliminés dans la phase de génération des fichiers translittérés à partir des fichiers translittérés bruts du corpus.

## 5-4 Outil pour le prétraitement développé

Nous avons développé un outil en Visual Basic, qui prend en charge l'ensemble de ces tâches de prétraitement.

A l'entrée nous lui fournissons les textes en XML et un fichier de mots outils et à l'issue du traitement nous obtenons :

- un répertoire de textes bruts ; c'est-à-dire, sans les balises XML,
- un répertoire de textes translittérés bruts,
  - élimination des caractères latins, les symboles et les signes diacritiques,
  - remplacement des caractères Arabes par leurs correspondants latins (cf. tableau 5.3).

- un répertoire de textes translittérés déchargés des mots outils,
  - élimination des mots outils.
- un répertoire de textes translittérés et radicalisés (stemmés)
  - utilisation des règles de radicalisation vues plus haut.

Cet outil intègre Al-stem, écrit en PERL, fourni par Kareem Darwish auquel nous avons apporté des modifications en vue de procéder à un traitement batch de fichiers car l'outil original traite seulement de séquence de caractères entrée par console.

Après cette phase nous avons donc obtenu une base d'apprentissage avec et sans radicalisation que nous avons utilisée dans nos expérimentations.

# **Chapitre 6**

# **Expérimentations**

## Introduction

Dans ce présent travail nous avons mené une série d'expérimentations sous un environnement d'apprentissage en utilisant le corpus CCA.

### 6.1 Présentation de l'environnement utilisé

YALE (Yet another learning environment) devenu par la suite RapidMiner est l'environnement d'expérimentation que nous avons utilisé, plusieurs versions sont apparues la dernière est la version 4.0. Il est gratuitement distribué sur le WEB. Il a été conçu par une équipe de chercheurs de l'université de Dortmund. C'est un Java Open Source pour solutions Datamining. Quatre plugins (composants à rattacher) sont aussi offerts. Voilà la liste complète de ces plugins

Plugin	Objectif
Text	<b>Le plugin texte (formellement WVTool : Word Vector Tool plugin) peut être utilisé pour créer les vecteurs mots à partir de textes</b>
Value series	Le plugin Série de valeur offre des méthodes pour l'extraction automatique d'attributs à partir des séries de données
Data stream	Le plugin Flux de Données offre des opérateurs pour l'exploration des flux de données
CRF (Conditional Random Field plugin)	Le plugin champ conditionnel aléatoire offre quelques opérateurs de base pour la reconnaissance d'entités nommées.

Tableau 6.1 : les plugins disponibles pour RapidMiner

Offrant plus de 400 opérateurs regroupés en groupes chacun décrivant un ensemble de méthodes soit de traitement de données soit d'apprentissage. Les groupes d'opérateurs sont :

- les opérateurs d'entrée/sortie (Mise en forme des attributs, Générateurs d'exemples, Mise en forme des résultats,...),
- les opérateurs d'apprentissage (Règles d'association, Bayes, Fonctions noyaux, Arbres de décisions,...),
- les opérateurs de pré et post-traitement (Agrégations, filtres, Réduction de dimension,...),
- les opérateurs de validation (Evaluation des performances, validation simple et croisée, ...),
- les opérateurs de visualisation (Graphes, statistiques, ...).

Pour notre étude nous avons utilisé les opérateurs, d'entrée pour l'entrée des vecteurs TF-IDF, (ExempleSource) , De chargement des résultats ( PerformanceLoader), de pré-traitement pour la réduction de la dimension par sélection (RemoveCorrelatedFeatures), Pour la construction du classifieur (JMySVMLearner et ComplementNaiveBayes) et pour la validation des résultats du classifieur (XValidation).

## 6-1.1 Obtention de la représentation TF-IDF

Pour la transformation des données texte en vecteurs TF-IDF, nous avons utilisé le plugin Text pour la transformation des textes en entrée en vecteurs TF-IDF dont voici le schéma d'expérimentation pour les différents ensembles d'apprentissage :

Schéma graphique	Schéma XML
	<pre> &lt;operator name="Root" class="Experiment"&gt;   &lt;operator name="WVTool" class="WVTool"&gt;     &lt;list key="texts"&gt;       &lt;parameter key="+1" value="F:\EXP1\brutclasse1"/&gt;       &lt;parameter key="-1" value="F:\EXP1\brutclasse2"/&gt;     &lt;/list&gt;   &lt;/operator&gt; &lt;/operator&gt; </pre>

## 6-1.2 Construction du classifieur SVM

Pour la construction de classifieurs SVM nous avons utilisé le schéma d'expérimentation suivant :


Schéma graphique	Schéma XML
	<pre> &lt;operator name="Root" class="Experiment"&gt;   &lt;operator name="ExampleSource" class="ExampleSource"&gt;     &lt;parameter key="attributes" value="D:\YALE\yale-3.4\experimentation\exp1\exp1_brut.aml"/&gt;   &lt;/operator&gt;   &lt;operator name="XValid" class="XValidation"&gt;     &lt;parameter key="create_complete_model" value="true"/&gt;     &lt;parameter key="keep_example_set" value="true"/&gt;     &lt;parameter key="number_of_validations" value="5"/&gt;   &lt;/operator&gt;   &lt;operator name="JMySVMLeamer" class="JMySVMLeamer"&gt;   &lt;/operator&gt;   &lt;operator name="ApplierChain" class="OperatorChain"&gt;     &lt;operator name="Applier" class="ModelApplier"&gt;       &lt;list key="application_parameters"&gt;       &lt;/list&gt;     &lt;/operator&gt;   &lt;/operator&gt;   &lt;operator name="Evaluator" class="PerformanceEvaluator"&gt;     &lt;list key="additional_performance_criteria"&gt;     &lt;/list&gt;     &lt;parameter key="f_measure" value="true"/&gt;     &lt;parameter key="precision" value="true"/&gt;     &lt;parameter key="recall" value="true"/&gt;   &lt;/operator&gt; &lt;/operator&gt; &lt;/operator&gt; &lt;/operator&gt; &lt;/operator&gt; </pre>

### Validation croisée (leave one out)

Nous avons opté pour une validation croisée leave one out dont le principe repose sur le découpage de l'ensemble des exemples en  $n$  (Nous avons opté pour  $n=5$ ) mutuellement disjoints et où chaque classe (Classe : +1 et -1 ) est distribué avec la même fréquence sur chacun de ces ensembles. Soit A,B,C,D,E ces ensembles. On construit le classifieur sur  $A \cup B \cup C \cup D$  et on mesure le taux d'erreurs sur E, ensuite on construit le classifieur sur  $A \cup B \cup C \cup E$  et on prend les mesures (précision, rappel f-mesure) sur D et ainsi de suite. Les différentes mesures sont alors estimées par la moyenne des 5 mesures sur chacun des classifieurs.


### 6-1.3 Réduction de dimension

En vue de comparer les résultats avec une base réduite nous avons procédé à la réduction de dimension des différents ensembles d'apprentissage dont voici le schéma.

Schéma graphique	Schéma XML
 <p>Le schéma graphique montre une hiérarchie de noeuds. Le noeud racine est 'Root' (classe 'Experiment'). Sous 'Root', il y a un noeud 'WVTTool' (classe 'WVTTool'). Sous 'WVTTool', il y a deux noeuds 'RemoveCorrelatedFeatures' (classe 'RemoveCorrelatedFeatures').</p>	<pre>&lt;operator name="Root" class="Experiment"&gt; &lt;operator name="WVTTool" class="WVTTool"&gt; &lt;list key="texts"&gt; &lt;parameter key="+1" value="F:\EXP1\brutclasse1"/&gt; &lt;parameter key="-1" value="F:\EXP1\brutclasse2"/&gt; &lt;/list&gt; &lt;/operator&gt; &lt;operator name="RemoveCorrelatedFeatures" class="RemoveCorrelatedFeatures"&gt; &lt;/operator&gt; &lt;/operator&gt;</pre>

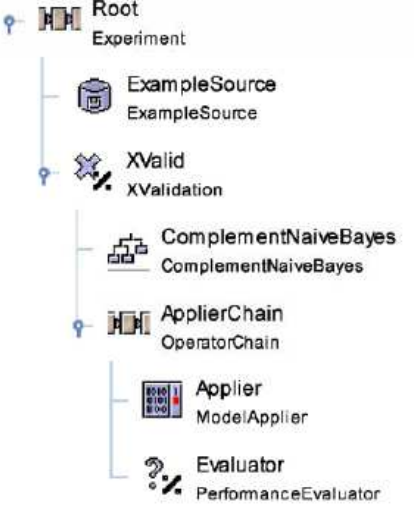
### 6-1.4 Chargement des résultats

Pour le chargement des résultats nous avons créé les expérimentations dont voici le schéma

Schéma graphique	Schéma XML
 <p>Le schéma graphique montre une hiérarchie de noeuds. Le noeud racine est 'Root' (classe 'Experiment'). Sous 'Root', il y a un noeud 'PerformanceLoader' (classe 'PerformanceLoader').</p>	<pre>&lt;operator name="Root" class="Experiment"&gt; &lt;operator name="PerformanceLoader" class="PerformanceLoader"&gt; &lt;parameter key="performance_file" value="D:\YALE\yale- 3.4\experimentation\exp1\exp1_brut.per"/&gt; &lt;/operator&gt; &lt;/operator&gt;</pre>

## 6-1.5 Construction du classifieur Naive Bayes

En vue de comparer les résultats obtenus avec les SVM nous avons jugé utiles de procéder aux mêmes expérimentations en faisant un apprentissage avec Naive Bayes dont voici le schéma

Schéma graphique	Schéma XML
	<pre data-bbox="654 492 1420 1321"> &lt;operator name="Root" class="Experiment"&gt;   &lt;operator name="ExampleSource" class="ExampleSource"&gt;     &lt;parameter key="attributes" value="D:\YALE\yale- 3.4\experimentation\exp1\exp1_brut.aml"/&gt;   &lt;/operator&gt;   &lt;operator name="XValid" class="XValidation"&gt;     &lt;parameter key="create_complete_model" value="true"/&gt;     &lt;parameter key="keep_example_set" value="true"/&gt;     &lt;parameter key="number_of_validations" value="5"/&gt;     &lt;operator name="ComplementNaiveBayes" class="ComplementNaiveBayes"&gt;   &lt;/operator&gt;     &lt;operator name="ApplierChain" class="OperatorChain"&gt;       &lt;operator name="Applier" class="ModelApplier"&gt;         &lt;list key="application_parameters"&gt;         &lt;/list&gt;       &lt;/operator&gt;       &lt;operator name="Evaluator" class="PerformanceEvaluator"&gt;         &lt;list key="additional_performance_criteria"&gt;         &lt;/list&gt;         &lt;parameter key="f_measure" value="true"/&gt;         &lt;parameter key="precision" value="true"/&gt;         &lt;parameter key="recall" value="true"/&gt;       &lt;/operator&gt;     &lt;/operator&gt;   &lt;/operator&gt; &lt;/operator&gt; </pre>

## 6.2 Expérimentations réalisées

Pour l'expérimentation nous avons construit 10 classifieurs bi-classe sur les 8 classes les plus peuplées du corpus c'est-à-dire contenant plus de 25 documents en plus de deux classes que nous avons montées sur les classes du corpus, en l'occurrence la classe « science et santé et médecine » et la classe « politique, sociologie et religion ».

Chaque classifieur a été bâti sous quatre aspects :

- 1- en données brutes,
- 2- en données radicalisées,
- 3- en données brutes avec réduction de dimension,
- 4- en données radicalisées avec réduction de dimension.

Nous avons étiqueté les documents appartenant à la classe des documents pertinents par « +1 » et nous avons pris autant de documents aléatoirement dans le restant du corpus pour construire la classe « -1 » de document non pertinents pour la classe considérée.

Voici un récapitulatif des classes utilisées :

<b>N°Expérimentation</b>	<b>Classe</b>	<b>Nb de documents</b>
1	Autobiography	144
2	Science	90
3	Tourism & travel	120
4	Sociology	60
5	Economics	58
6	Children's stories	54
7	Health and medicine	64
<b>8</b>	<b>Health and medicine and science</b>	<b>154</b>
<b>9</b>	<b>Politics, sociology and religion</b>	<b>136</b>
10	Short stories	62

Tableau 6.2 : Les classes utilisées dans les expérimentations

Nous avons donc mené 10 expérimentations différentes sous chacune des 4 conditions suscitées soit au total 40 expérimentations

Pour la comparaison nous avons mené les mêmes expérimentations mais en utilisant Naive Bayes à la place de SVM comme modèle d'apprentissage

Dans le chapitre qui va suivre nous allons donner les résultats de ces expérimentations



# **Chapitre 7**

## **Résultats et discussion**

## 7-4 Résultats

Avec ou sans réduction de dimension	N° expérimentation	Classe du corpus	Type de données	Cardinal de l'ensemble d'apprentissage	Nombre d'attributs	Taux de compression	VP	VN	FP	FN	Précision	Rappel	F-Mesure
Sans réduction de dimension	1	Autobiography	Brutes	144	49 428		70	10	3	63	95,45%	86,30%	90,65%
			Radicalisées		18 863	82%	70	9	3	64	95,52%	87,67%	91,43%
	2	Science	Brutes	90	37 651		44	4	1	41	98,00%	91,11%	94,09%
			Radicalisées		14 383	62%	37	3	8	42	86,31%	96,33%	89,16%
	3	Tourism & travel	Brutes	120	25 985		55	19	5	41	89,13%	68,33%	77,36%
			Radicalisées		10 442	60%	54	9	6	51	89,47%	85,00%	87,18%
	4	Sociology	Brutes	60	32 841		21	7	9	23	75,48%	76,67%	73,70%
			Radicalisées		13 029	60%	20	8	10	22	72,00%	73,33%	70,14%
	5	Economics	Brutes	58	25 805		29	8	0	21	100,00%	74,29%	80,56%
			Radicalisées		10 282	60%	28	7	1	22	96,00%	76,48%	82,64%
	6	Children's stories	Brutes	54	21 122		27	11	0	16	100,00%	58,67%	73,43%
			Radicalisées		8 932	58%	26	8	1	19	96,00%	70,67%	79,43%
	7	Health and medicine	Brutes	64	25 245		32	15	0	17	100,00%	53,33%	68,22%
			radicalisées		10 275	59%	30	8	2	24	93,81%	75,24%	81,79%
	8	Health and medicine and science	Brutes	154	48 002		76	5	1	72	98,75%	93,58%	96,04%
			Radicalisées		17 973	63%	75	8	2	69	97,42%	89,58%	93,17%
	9	Politics, sociology and religion	Brutes	136	53 250		45	6	23	62	73,32%	91,21%	81,16%
			Radicalisées		19 714	63%	51	12	17	56	76,94%	82,42%	79,40%
	10	Short stories	Brutes	62	23 850		31	7	0	24	100,00%	77,42%	87,27%
			Radicalisées		10 289	57%	28	4	3	27	90,00%	87,10%	88,52%
<b>Moyenne</b>		Brutes	94	34 318		430,0	92,0	42,0	380,0	93,01%	77,09%	82,25%	
		Radicalisées		13 418	61%	419,0	76,0	53,0	396,0	89,35%	82,38%	84,29%	

Avec ou sans réduction de dimension	N° expérimentation	Classe du corpus	Type de données	Cardinal de l'ensemble d'apprentissage	Nombre d'attributs	Taux de compression	VP	VN	FP	FN	Précision	Rappel	F-Mesure
Avec réduction de dimension	11	Autobiography	Brutes	144	16 038	68%	71	9	2	64	96,97%	87,67%	92,09%
			Radicalisées		8 015	58%	73	10	0	63	100,00%	86,30%	92,65%
	12	Science	Brutes	90	9 758	74%	42	3	3	42	94,00%	93,33%	93,17%
			Radicalisées		5 284	63%	39	4	6	41	87,78%	91,11%	89,23%
	13	Tourism & travel	Brutes	120	7 156	72%	49	2	11	58	84,06%	96,67%	89,92%
			Radicalisées		3 930	62%	50	2	10	58	85,29%	96,67%	90,62%
	14	Sociology	Brutes	60	6 543	80%	28	12	2	18	92,67%	60,00%	70,55%
			Radicalisées		4 017	69%	28	14	2	16	93,33%	53,33%	66,00%
	15	Economics	Brutes	58	5 415	79%	26	2	3	27	90,29%	93,33%	91,21%
			Radicalisées		3 203	69%	22	3	7	26	82,10%	90,00%	83,11%
	16	Children's stories	Brutes	54	2 941	86%	27	11	0	16	100,00%	58,67%	73,43%
			Radicalisées		1 992	78%	26	8	1	19	96,00%	70,67%	79,43%
	17	Health and medicine	Brutes	64	4 725	81%	26	0	6	32	86,39%	100,00%	92,17%
			Radicalisées		2 921	72%	27	1	5	31	88,33%	97,14%	91,60%
	18	Health and medicine and science	Brutes	154	15 065	69%	73	1	4	76	95,22%	98,75%	96,93%
			Radicalisées		7 579	58%	75	5	2	72	97,42%	93,50%	95,39%
	19	Politics, sociology and religion	Brutes	136	15 024	72%	56	23	12	45	79,51%	66,26%	71,64%
			Radicalisées		7 854	60%	51	12	17	56	76,94%	82,42%	79,40%
	20	Short stories	Brutes	62	4 503	81%	26	3	5	28	84,85%	90,32%	87,50%
			Radicalisées		3 011	71%	29	4	2	27	93,10%	87,10%	90,00%
<b>Moyenne</b>		Brutes	94	8 717	75%	424,0	66,0	48,0	406,0	90,40%	84,50%	85,86%	
		Radicalisées		4 781	64%	420,0	63,0	52,0	409,0	90,03%	84,82%	85,74%	

Tableau 7.1 : Tableau récapitulatif des différentes expérimentations

## 7-5 Différentes mesures

Nous avons opté pour le calcul des 3 mesures les plus utilisées dans la catégorisation de textes à savoir, la précision, le rappel et la F-mesure.

### 7-5.1 Précision

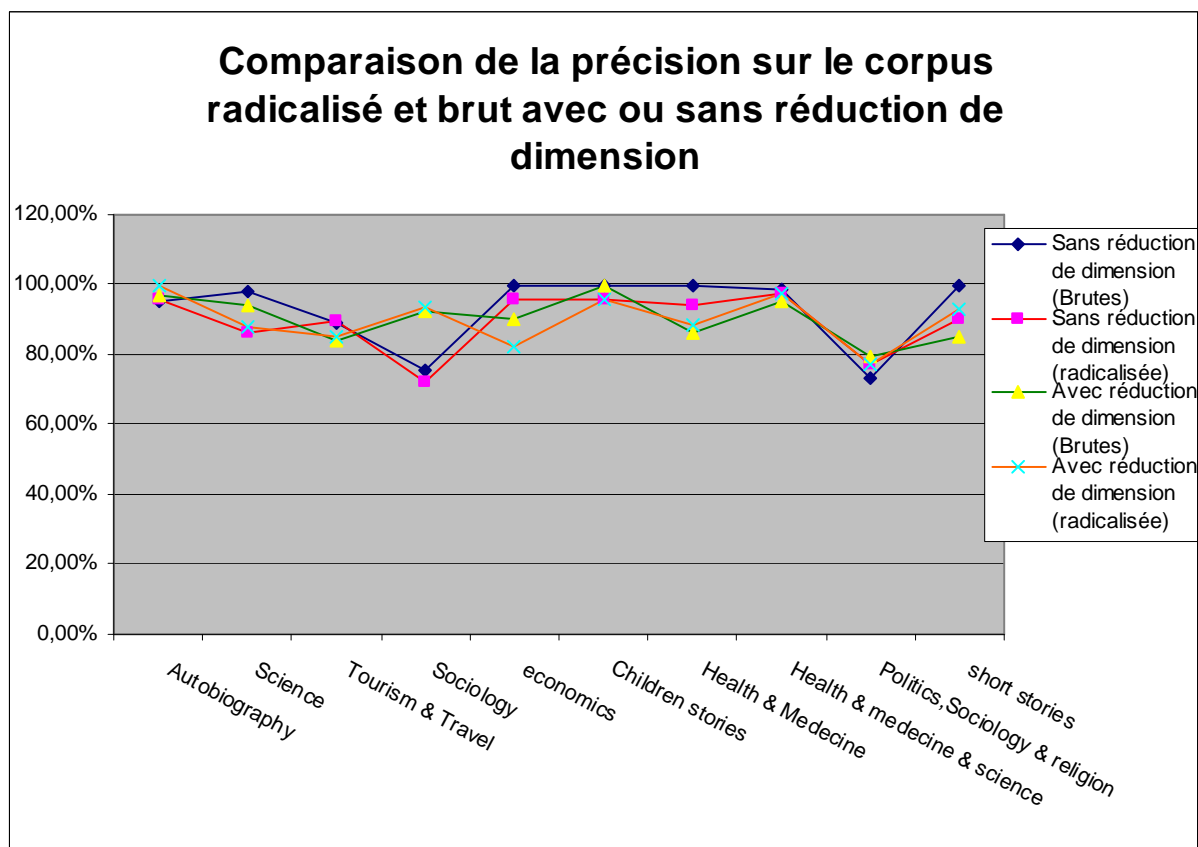


Fig 7.1 : Comparaison de la précision sur le corpus radicalisé et brut avec ou sans réduction de dimension

#### Moyennes :

Nature des données d'apprentissage	Micro Précision
Sans réduction de dimension et sans stemming	93,01%
Sans réduction de dimension et avec stemming	89,35%
Avec réduction de dimension et sans stemming	90,40%
Avec réduction de dimension et avec stemming	90,03%

#### Discussion

On remarque que la précision se dégrade avec le stemming

Pour les données sans réduction de dimension cette perte est de : 3,66 %

Pour les données avec réduction de dimension il y a un petit peu de perte de précision qui est de l'ordre de 0,37 % entre les données radicalisées et les données non radicalisées.

On observe tout de même une perte de précision globale de 0,97% entre les données sans réduction de dimension et les données avec réduction de dimension

Si le stemming nuit à la précision, la réduction de dimension en fait plus.

## 7-5.2 Rappel

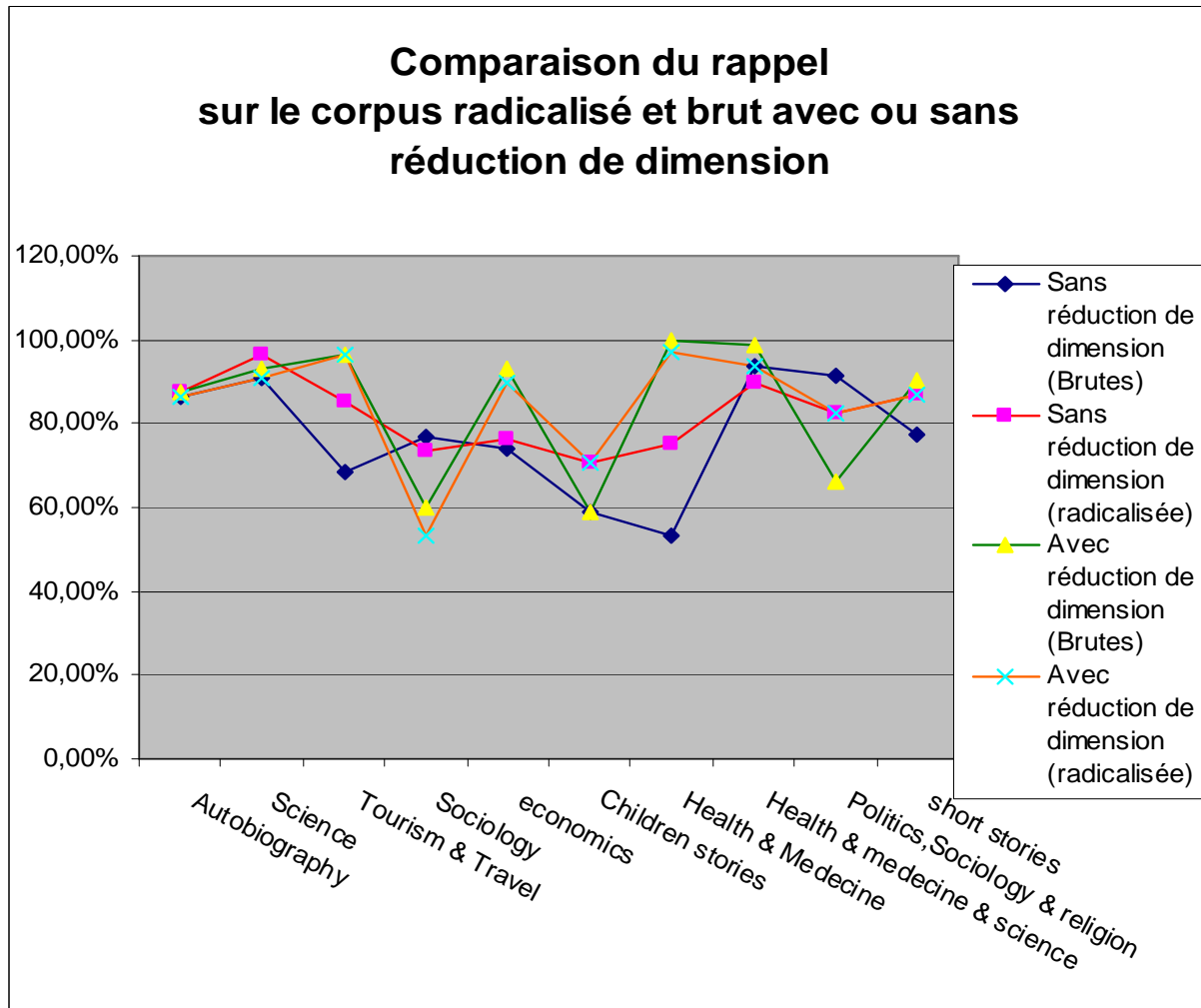


Fig 7.2 : Comparaison du rappel sur le corpus radicalisé et brut avec ou sans réduction de dimension

### Moyennes :

Nature des données d'apprentissage	Micro-rappel	Macro-rappel
Sans réduction de dimension et sans stemming	77,09%	82,38%
Sans réduction de dimension et avec stemming	82,38%	84,65%
Avec réduction de dimension et sans stemming	84,50%	86,53%
Avec réduction de dimension et avec stemming	84,82%	86,96%

### Discussion

On remarque que le rappel augmente avec le stemming ce qui confirme l'intuition. Il est amélioré de 5,29% sur les données sans réduction de dimension et de 0,32 % sur les données avec réduction de dimension

Aussi, il est à noter que le rappel augmente significativement avec la réduction de dimension. Cette augmentation est de 4,93 %

La réduction de dimension et le stemming ont un effet positif sur le rappel

### 7-5.3 F-mesure

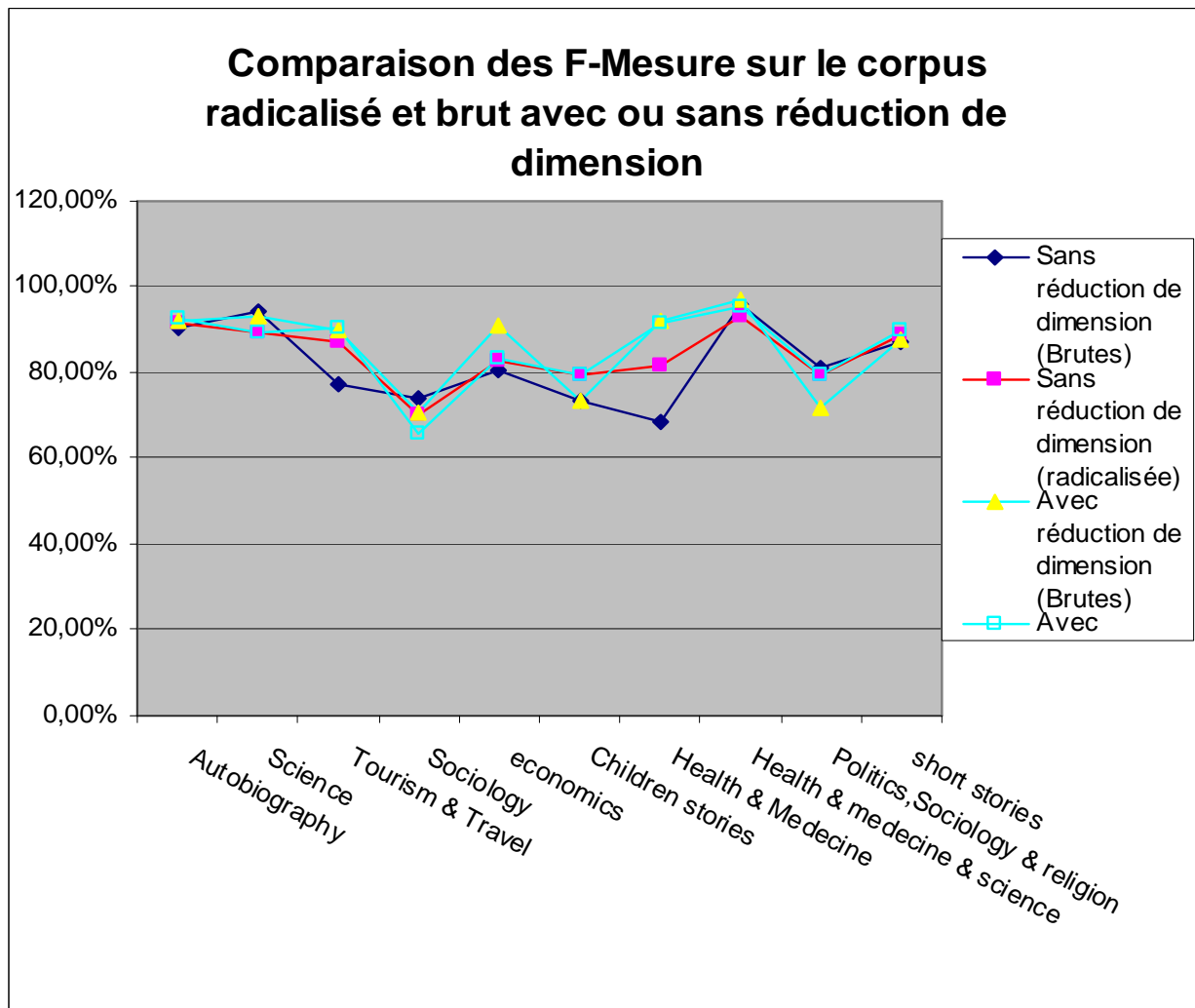


Fig 7.3 : Comparaison de la F-mesure sur le corpus radicalisé et brut avec ou sans réduction de dimension

#### Moyennes :

Nature des données d'apprentissage	micro f-mesure
Sans réduction de dimension et sans stemming	82,25%
Sans réduction de dimension et avec stemming	84,29%
Avec réduction de dimension et sans stemming	85,86%
Avec réduction de dimension et avec stemming	85,74%

#### Discussion

On remarque que la f-mesure est améliorée avec le stemming

Pour les données sans réduction de dimension ce gain est de : 2,04 %

Pour les données avec réduction de dimension ce gain est de :

\* 3,61 % pour les données brutes

\* 1,45 % pour les données radicalisées

On remarque tout de même qu'il n'y a aucun gain dans le cas de réduction de dimension entre les données brutes ou radicalisées

Conclusion : Si on ne procède pas à une réduction de dimension, Le stemming améliore la f-mesure

Dans le cas où on a fait une réduction de dimension le stemming n'améliore pas forcément la f-mesure. Mais il est globalement plus intéressant que celui obtenu sur des données non radicalisées

#### 7-5.4 Quelques remarques sur le stemming et la réduction de dimension

Le stemming ou radicalisation que nous avons opéré sur les textes bruts a permis la compression à plus de 60% sur les textes bruts sans réduction de dimension et à près de 45 % sur les textes après réduction de dimension. Avec le stemmer de Porter appliqué au corpus Reuters-21578, le taux de compression est de 61% ( 15247 racines sur une base d'index de 39289) ce qui laisse à penser que al-stem est un outil aussi efficace pour la langue Arabe que l'est celui de Porter sur l'Anglais.

Pour la réduction de dimension, on note que près de 4 attributs sont corrélés sur les textes bruts et près de 3 attributs sur les textes radicalisés. La réduction de dimension permet d'obtenir des dimensions plus réduites à près de 3/4 sur les textes bruts et près de 2/3 sur les textes radicalisés. Sur de grandes bases d'apprentissage cette compression est très significative car elle contribue à une amélioration significative du temps de construction des classifieurs ainsi que les ressources nécessaires telles que la mémoire et les processeurs. Ce gain est très important pour des applications temps réel.

#### 7-5.5 Comparaison entre Bayes et SVM

Nous avons mené les mêmes expérimentations en entraînant des classifieurs avec la méthode Naive Bayes sur les mêmes classes utilisées avec les SVM. Les résultats obtenus sont présentés dans les tableaux 7.2 et 7.3

Méthode	Nature de données	Précision	Rappel	F-mesure
Bayes	Données brutes	94,73%	79,21%	84,85%
	Données radicalisées	93,33%	78,48%	83,83%
SVM	Données brutes	93,01%	77,09%	82,25%
	Données radicalisées	89,35%	82,38%	84,29%

Tableau 7.2 : Comparaison des résultats obtenus entre les SVM et Naive Bayes sans réduction de dimension

Méthode	Nature de données	Précision	Rappel	F-mesure
Bayes	Données brutes	93,59%	83,33%	87,07%
	Données radicalisée	92,93%	83,55%	86,78%
SVM	Données brutes	90,40%	84,50%	85,86%
	Données radicalisées	90,03%	84,82%	85,74%

Tableau 7.3 : Comparaison des résultats obtenus entre les SVM et Naive Bayes avec réduction de dimension

Nous remarquons en général que la précision avec Naive Bayes est meilleure qu'avec SVM et que le rappel est dégradé et qu'en général la f-mesure est meilleure avec Bayes.

Nous restons tout de même septiques sur cette comparaison car d'autres travaux ont montré que SVM est meilleur pour les deux mesures [Dumais, 1998]. Nous estimons que cela est dû au nombre de documents des ensembles d'apprentissage. Dumais dans [Dumais, 1998] a traité la question de la taille des classes en apprentissage. Elle a estimé que plus les classes sont grandes plus les résultats deviennent stables et par là une comparaison entre les classifieurs devient possible. Dans ses expérimentations, elle a observé que SVM est meilleur que Bayes quand cette taille est de l'ordre de quelques centaines de documents.

Sur le corpus Reuter 21578, connu pour sa taille considérable (21578 documents répartis sur 118 catégories), les résultats avec SVM sont meilleurs qu'avec d'autres méthodes d'apprentissage. Les SVM performant entre 84 et 92% de F-mesure sous différentes conditions (tailles de l'ensemble d'apprentissage, taille de l'ensemble de test, nombre de classe d'expérimentation...)

On remarque tout de même que pour la langue Arabe sur le corpus CCA avec les conditions d'utilisation discutées au niveau du chapitre 6 les résultats coïncident avec ceux obtenus pour le corpus Reuters 21578.

**Ceci nous mène à conclure que l'Arabe ne présente pas de particularité dans la phase d'apprentissage. Ses seules particularités résident dans la préparation des données (prétraitement).**

## 7-6 Discussion générale

A l'issue des expérimentations que nous avons menées sur le corpus CCA avec les machines à vecteurs supports et en utilisant la technique de stemming pour la radicalisation des termes, nous pouvons conclure ce qui suit :

Pour la langue Arabe, le stemming est une technique de représentation qui apporte une amélioration lors de la classification des documents comme dans les autres langues déjà testées. Cependant pour des applications favorisant la précision, cette technique contribue au bruitage du rendu par des documents non pertinents.

La réduction de dimension, si elle contribue à l'amélioration du temps de construction de classifieurs, elle possède les mêmes effets que le stemming au niveau de la précision. Quoiqu'en général, la f-mesure, compromis entre le rappel et la précision se trouve améliorée dans les deux cas.



# **Conclusion et perspectives**

## Conclusion et perspectives

Dans ce travail, nous avons abordé la problématique de la classification de documents textuels en langue Arabe.

A travers notre investigation, nous avons remarqué que les travaux sur la classification de documents en langue Arabe sont rares.

Nous avons remarqué que notre langue se singularise par des particularités qui la rendent difficile à appréhender en vue de l'extraction d'informations statistiques, car notre objectif était de tester des méthodes numériques qui ont démontré leur robustesse dans les autres langues. Les problèmes d'extraction d'attributs s'ils s'avèrent difficiles pour les langues telles que l'Anglais ou le Français, elles le sont encore plus pour la langue Arabe.

Nous avons étudié la panoplie des méthodes de classification. Les machines à vecteurs supports nous ont le plus séduits malgré leur jeune âge (1995). Cependant, leur compréhension et le réglage des paramètres utilisés sont d'une grande difficulté.

Nous avons démarré de l'à priori que, vu les particularités de la langue Arabe, les méthodes de classification basées sur la représentation en sacs de mots seraient peut-être inefficaces et donneraient des résultats médiocres et on serait alors amené à penser que seules les méthodes se basant sur une analyse morphosyntaxique sont prometteuses. Ce constat s'est avéré faux car à l'issue des résultats que nous avons obtenus nous pouvons conclure que les méthodes d'extraction d'attributs testées sur les autres langues sont adaptables à la langue Arabe et que seuls les prétraitements pour cette langue sont d'une complexité avérée. Les résultats des traitements, quant à eux, sont comparables à ceux obtenus sur les autres langues. Nous continuons à penser malgré tout que notre travail serait d'un intérêt avéré si nous le déroulerons sur un corpus plus important.

De ce fait, l'une de nos perspectives futures seraient d'élaborer un moyen de construction automatique de corpus en langue Arabe afin de le mettre à la disposition des équipes de recherche en « Recherche d'information ».

En ce qui concerne la classification automatique de textes Arabes, plusieurs issues sont encore ouvertes à savoir :

- **la représentation de documents** : Il n'a pas encore été prouvé que la représentation par les mots est la meilleure pour la langue Arabe. Il serait alors intéressant de tenter d'expérimenter d'autres représentations telles que les phrases ou les concepts guidés par une ontologie.
- **l'hybridation de méthodes** : Il serait peut être intéressant de penser à utiliser de façon complémentaire les informations tirées à partir d'une analyse morphosyntaxique des documents et ceux obtenu à travers les méthodes de représentation numérique en vue de construire des classfieurs et évaluer l'apport de cette hybridation.

# **Références et bibliographie**

**[A. Lehman et P. Bouvet, 2001]** A. Lehman et P. Bouvet (2001). Evaluation, rectification et pertinence du résumé automatique de texte pour une utilisation en réseau. Colloque du Chapitre Français de l'ISKO (International Society of Knowledge Organization) 5-6 juillet 2001 à l'Université de Paris X "Filtrage et résumé automatique de l'information sur les réseaux", PP. 111-125, Paris

**[Aas et Eikvil, 1999]** K. Aas et L. Eikvil (1999). Text categorization: a survey. Technical report, Norwegian Computing Center.

**[Abdelali, et al, 2004]** A. Abdelali, J. Cowie et S.H. Soliman (2004). Arabic Information Retrieval Perspectives. Proceedings of JEP-TALN 2004 Arabic Language Processing, Fez 19-22. April 2004.

**[Al-Sulaiti, L. et Atwell, E., 2004]** L. Al-Sulaiti et E. Atwell (2004). The Design of a Corpus of Contemporary Arabic. International Journal of Corpus Linguistics, vol. 11, forthcoming, 2006.

**[Aljlal et Frieder, 2002]** M. Aljlal et O. Frieder (2002). On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach, In 11<sup>th</sup> International Conference on Information and Knowledge Management (CIKM), November 2002, Virginia (USA), PP.340-347.

**[Androutopoulos et al., 2000]** I. Androutopoulos, J. Koutsias, K.V. Chandrinou et Spyropoulos, C. D. (2000). An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. In Belkin, N. J., Ingwersen, P., et Leong, M.-K., editors, Proceedings of SIGIR-00, 23<sup>rd</sup> ACM International Conference on Research and Development in Information Retrieval, PP. 160–167, Athens, GR. ACM Press, New York, US.

**[Apté et al., 1994]** C. Apté, F.J. Damerau, et S.M. Weiss (1994). Automated learning of decision rules for text categorization. ACM Transactions on Information Systems, 12 (3) PP. 233–251.

**[Attia, 2000]** M. Attia (2000). A large-scale computational processor of the Arabic morphology, A Master's Thesis, Cairo University, (Egypt).

**[Baloul et al., 2002]** S. Baloul, M. Alissali, M. Baudry et P. Boula de Mareuil (2002). Interface syntaxe-prosodie dans un système de synthèse de la parole à partir du texte en Arabe, 24<sup>e</sup> Journées d'Étude sur la Parole, 24-27 juin 2002 Nancy, PP. 329-332.

**[Benzecri, 1976]** J.P. Benzecri (1976). L'Analyse des Données, volume 2. Dunod, Paris.

**[Bladi et al, 2003]** P. Baldi, Frasconi et P. Smyth (2003). Modeling the Internet and the Web Probabilistic Methods and Algorithms P. ISBN: 0-470-84906-1

**[Borko et Bernick, 1964]** H. Borko et M. Bernick (1964). Automatic document classification. part II : additional experiments. Journal of the Association for Computing Machinery, 11(2) PP. 138–151

**[Buckley et Salton, 1995]** C. Buckley et G. Salton, (1995). Optimization of relevance feedback weights. In Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, PP. 351–357, New York.

**[Buckwalter]** Buckwalter, T. Qamus: Arabic lexicography

**[Carreras et Márquez, 2001]** X. Carreras et L. Márquez, (2001). Boosting trees for anti-spam email filtering. In Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing, Tzigov Chark, BG.

**[Caropreso et al., 2001]** M.F. Caropreso, S. Matwin et F. Sebastiani (2001). A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In Chin, A. G., editor, Text Databases and Document Management : Theory and Practice, PP. 78–102. Idea Group Publishing, Hershey, US.

**[Cavnar et Trenkle, 1994]** W.B. Cavnar et J.M. Trenkle (1994). N-gram-based text categorization. In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, PP. 161–175, Las Vegas, US.

**[Chai et al., 2002]** K.M. Chai, H.T. Ng, et H.L. Chieu (2002). Bayesian online classifiers for text classification and filtering. In Beaulieu, M., Baeza-Yates, R., Myaeng, S. H., et Järvelin, K., editors, Proceedings of SIGIR-02, 25th ACM International Conference on Research and Development in Information Retrieval, PP. 97–104, Tampere, FI. ACM Press, New York, US.

**[Chen et Gey, 2002]** A. Chen et F. Gey (2002) : Building an Arabic Stemmer for Information Retrieval. Proceedings of the Eleventh Text Retrieval Conference (TREC 2002). National Institute of Standards and Technology, Nov 18-22, 2002, PP. 631-640.

**[Cohen, 1996]** W.W. Cohen (1996). Learning rules that classify e-mail. In The 1996 AAAI Spring Symposium on Machine Learning in Information Access, PP. 18–25

**[Cohen et Singer, 1999]** W.W. Cohen et Y. Singer (1999). Contextsensitive learning methods for text categorization. ACM Transactions on Information Systems, 17(2) : PP.141–173.

**[Darwish, 2002]** K. Darwish(2002). Building a Shallow Arabic Morphological Analyzer in One Day. Proceedings of the workshop on Computational Approaches to Semitic Languages in the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02), Philadelphia, PA, USA. PP. 47-54.

**[Darwish, 2003]** K. Darwish (2003). Probabilistic Methods for Searching OCR-Degraded Arabic Text, Doctoral dissertation, University of Maryland.

**[De Loupy, 2001]** C. de Loupy (2001). L’apport de connaissances linguistiques en recherche documentaire. In TALN’01.

**[Débili et al., 2002]** F. Débili , H. Achour et E. Souici (2002). La langue Arabe et l'ordinateur: de l'étiquetage grammatical à la voyellation automatique, Correspondances de l'IRMC, N° 71, juillet-août 2002, PP. 10-28.

**[Deerwester et al., 1990]** S. Deerwester, S. Dumais, T. Landauer, G. Furnas et R. Harshman (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information science*, 41(6) PP. 391–407.

**[Djelailia K. et al., 2007]** K. Djelailia, H.F. Merouani, Y. Tlili, Les Machines à vecteur support dans la catégorisation de textes arabes *Proceeding COSI 2007 (Oran 11-13 Juin 2007)*, JED2007 (Annaba 27-28 Mai 2007)

**[Dumais et al., 1998]** S. Dumais, J. Platt, D. Heckerman et M. Sahami (1998). Inductive learning algorithms and representations for text categorization. In Gardarin, G., French, J. C., Pissinou, N., Makki, K., et Bouganim, L., editors, *Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management*, PP. 148–155, Bethesda, US. ACM Press, New York, US.

**[El-Halles, 2006]** A. M. El-Halles (2006). 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).

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

**[Escofier, 1965]** B. Escofier (1965). Analyse Factorielle des Correspondances. PhD thesis, Université de Rennes. Publiée dans les Cahiers du B.U.R.O., numéro 13, 1969.

**[Escudero et al., 2000]** G. Escudero, L. Màrquez et G. Rigau (2000). Boosting applied to word sense disambiguation. In de Mántaras, R. L. et Plaza, E., editors, *Proceedings of ECML-00, 11th European Conference on Machine Learning*, PP. 129–141, Barcelona, ES. Springer Verlag, Heidelberg, DE. Published in the “Lecture Notes in Computer Science” series, number 1810.

**[Eyheramendy, 2003]** S. Eyheramendy, D. D. Lewis et D. Madigan (2003). On the Naïve Bayes Model for Text Categorization In proceedings of the 9th International Workshop on Artificial Intelligence and Statistics

**[Forsyth, 1999]** R.S. Forsyth (1999). New directions in text categorization. In Gammerman, A., editor, *Causal models and intelligent data management*, PP 151–185. Springer Verlag, Heidelberg.

**[Fuhr et Buckley, 1991]** N. Fuhr et C. Buckley (1991). A probabilistic learning approach for document indexing. In *ACM Transactions on Information Systems*, volume 9, PP. 223–248.

**[Gilli, 1988]** Y. Gilli (1988). Texte et fréquence. Number 360. Université de Besançon, Paris.

**[He et al., 2000]** J. He, A.H. Tan, et C.L. Tan (2000). A comparative study on chinese text categorization methods. In *PRICAI Workshop on Text and Web Mining*, PP. 24–35.

**[Hofmann, 1999]** T. Hofmann (1999). Probabilistic latent semantic indexing. In *Proceedings of SIGIR'99, 1999*.

**[Hull, 1994]** D.A. Hull (1994). Improving text retrieval for the routing problem using latent semantic indexing. In Croft, W. B. et van Rijsbergen, C. J., editors, Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval, PP. 282–289, Dublin, IE. Springer Verlag, Heidelberg.

**[Iyer et al., 2000]** Iyer, R. D., Lewis, D. D., Schapire, R. E., Singer, Y., et Singhal, A. (2000). Boosting for document routing. In Agah, A., Callan, J., et Rundensteiner, E., editors, Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management, PP 70–77, McLean, US. ACM Press, New York, US.

**[Joachims, 1997]** T. Joachims (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In Fisher, D. H., editor, Proceedings of ICML-97, 14th International Conference on Machine Learning, PP. 143–151, Nashville, US. Morgan Kaufmann Publishers, San Francisco, US.

**[Joachims, 1998]** T. Joachims (1998). Text categorization with support vector machines: learning with many relevant features. In Nédellec, C. et Rouveirol, C., editors, Proceedings of ECML-98, 10th European Conference on Machine Learning, PP 137–142, Chemnitz, Springer Verlag, Heidelberg. Published in the “Lecture Notes in Computer Science” series, number 1398.

**[Joachims, 1999]** T. Joachims (1999). Transductive inference for text classification using support vector machines. In Bratko, I. et Dzeroski, S., editors, Proceedings of ICML-99, 16th International Conference on Machine Learning, PP. 200–209, Bled, SL. Morgan Kaufmann Publishers, San Francisco, US.

**[Joachims, 2000]** T. Joachims (2000). Estimating the generalization performance of a SVM efficiently. In Langley, P., editor, Proceedings of ICML-00, 17th International Conference on Machine Learning, PP. 431–438, Stanford, US. Morgan Kaufmann Publishers, San Francisco, US.

**[Kadri et Benyamina, 1992]** Y. Kadri, A. Benyamina (1992), Système d’analyse syntaxico-sémantique du langage Arabe, mémoire d’ingénieur, université d’Oran Es-sénia.

**[Khreisat, 2006]** L. Khreisat (2006). Arabic Text Classification Using N-Gram Frequency Statistics A Comparative Study.

**[Kim et al., 2000]** Y.H. Kim, S.Y. Hahn et B.T. Zhang (2000). Text filtering by boosting naive Bayes classifiers. In Belkin, N. J., Ingwersen, P., et Leong, M.-K., editors, Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval, PP. 168–75, Athens, GR. ACM Press, New York, US

**[Kiraz, 1996]** G. A. Kiraz (1996). Analysis of the Arabic Broken Plural and Diminutive, In Proceedings of the 5th International Conference and Exhibition on Multi-Lingual Computing (ICEMCO96), Cambridge, UK.

**[Koller et Sahami, 1997]** D. Koller et M. Sahami (1997). Hierarchically classifying documents using very few words. In Fisher, D. H., editor, Proceedings of ICML-97, 14th International Conference on Machine Learning, PP. 170–178, Nashville, US. Morgan Kaufmann Publishers, San Francisco, US.

**[Larkey et al., 2002]** L.S. Larkey, L. Ballesteros et M. Connell (2002). Improving Stemming for Arabic Information Retrieval: Light Stemming and Cooccurrence Analysis, In Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2002), Tampere, Finland, August 2002, PP. 275-282.

**[Lebart et Salem, 1994]** L. Lebart, et A. Salem (1994). Statistique textuelle. Dunod, Paris.

**[Lewis, 1992a]** D.D. Lewis (1992). An evaluation of phrasal and clustered representations on a text categorization task. In Belkin, N. J., Ingwersen, P., et Pejtersen, A. M., editors, Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval, PP. 37–50, Kobenhavn, DK. ACM Press, New York, US.

**[Lewis, 1992b]** D.D. Lewis (1992). Representation and learning in information retrieval. PhD thesis, Department of Computer Science, University of Massachusetts, Amherst, US.

**[Lewis, 1998]** D.D. Lewis (1998). Naive (Bayes) at forty : The independence assumption in information retrieval. In Nédellec, C. et Rouveirol, C., editors, Proceedings of ECML-98, 10th European Conference on Machine Learning, PP. 4–15, Chemnitz, DE. Springer Verlag, Heidelberg. Published in the “Lecture Notes in Computer Science” series, number 1398.

**[Lewis et Ringuette, 1994]** D.D. Lewis et M. Ringuette (1994). A comparison of two learning algorithms for text categorization. In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, PP. 81–93, Las Vegas, US.

**[Lewis et al., 1996]** D.D. Lewis, R.E. Schapire, J.P. Callan, et R. Papka (1996). Training algorithms for linear text classifiers. In Frei, H.-P., Harman, D., Schäuble, P., and Wilkinson, R., editors, Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval, PP. 298–306, Zürich, CH. ACM Press, New York, US.

**[Li et Jain, 1998]** Y.H. Li et A.K. Jain (1998). Classification of text documents. The Computer Journal, 41(8): PP. 537–546.

**[Liu et al., 2002]** Y. Liu, Y. Yang et J. Carbonell (2002). Boosting to correct the inductive bias for text classification. In Proceedings of CIKM-02, 11th ACM International Conference on Information and Knowledge Management, McLean, US. ACM Press, New York, US.

**[Liddy et al., 1994]** E.D. Liddy, W. Paik et E.S. Yu (1994). Text categorization for multiple users based on semantic features from a machine-readable dictionary. ACM Transactions on Information Systems, 12(3) PP. 278–295.

**[Mitchell, 1997]** T.M. Mitchell (1997). Machine Learning. Computer Science. McGraw-Hill, New York.

**[Mladenić, 1998]** D. Mladenić (1998). Feature subset selection in text learning. In Nédellec, C. et Rouveirol, C., editors, Proceedings of ECML-98, 10th European Conference on Machine Learning, PP 95–100, Chemnitz, DE. Springer Verlag, Heidelberg. Published in the “Lecture Notes in Computer Science” series, number 1398.



**[Mladenić et Grobelnik, 1998]** D. Mladenić et M. Grobelnik (1998). Word sequences as features in text-learning. In Proceedings of ERK-98, the Seventh Electrotechnical and Computer Science Conference, PP. 145–148, Ljubljana, SL.

**[Morin, 2002]** A. Morin (2002). Factorial correspondence analysis : a dual approach for semantics and indexing. In Proceedings of the Conference Compstat.

**[Moulinier, 1996]** I. Moulinier (1996). Une approche de la catégorisation de textes par l'apprentissage symbolique. PhD thesis, Université Paris 6, Paris.

**[Moulinier, 1997]** I. Moulinier (1997). Feature selection: a useful preprocessing step. In Furner, J. et Harper, D., editors, Proceedings of BCSIRSG-97, the 19th Annual Colloquium of the British Computer Society Information Retrieval Specialist Group, Electronic Workshops in Computing, Aberdeen, UK. Springer Verlag, Heidelberg.

**[Ng et al., 1997]** H.T. Ng, W.B. Goh et K.L. Low (1997). Feature selection, perceptron learning, and a usability case study for text categorization. In Belkin, N. J., Narasimhalu, A. D., et Willett, P., editors, Proceedings of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval, PP. 67–73, Philadelphia, US. ACM Press, New York, US.

**[Porter, 1980]** M.F. Porter (1980). An algorithm for suffix stripping. Program, 14(3): PP. 130–137.

**[Rocchio, 1971]** J. Rocchio (1971). Relevance feedback in information retrieval. In Salton, G., editor, The SMART Retrieval System Experiments in Automatic Document Processing, PP. 313–323. Prentice-Hall.

**[Sable et Hatzivassiloglou, 2000]** C.L. Sable et V. Hatzivassiloglou (2000). Text-based approaches for non-topical image categorization. International Journal of Digital Libraries, 3(3): PP.261–275.

**[Sahami, 1999]** M. Sahami (1999). Using Machine Learning to Improve Information Access. PhD thesis, Computer Science Department, Stanford University.

**[Salton, 1988]** G. Salton et C. Buckley (1988). Term-weighting Approaches in Automatic Text Retrieval. Information Processing and Management, 24(5), PP ; 513-523.

**[Salton et McGill, 1983]** G. Salton et M. Mc Gill (1983). Introduction to Modern Information Retrieval. McGraw-Hill, New York.

**[Saul, L., et Pereira, F.,1997]** L.Saul et F. Pareira (1997). Aggregate and mixedorder Markov models for statistical language processing. In Cardie, C., et Weischedel, R. (Eds.), Proceedings of the Second Conference on Empirical Methods in Natural Language Processing PP. 81–89. Somerset, NJ: ACL Press.

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

**[Schapire et al., 1998]** R.E. Schapire, Y. Singer, et A. Singhal (1998). Boosting and Rocchio applied to text filtering. In Croft, W. B., Moffat, A., van Rijsbergen, C. J., Wilkinson, R., et Zobel, J., editors, Proceedings of SIGIR-98, 21<sup>st</sup> ACM International Conference on Research and Development in Information Retrieval, PP. 215–223, Melbourne, AU. ACM Press, New York, US.

**[Schapire et Singer, 2000]** R.E. Schapire et Y. Singer (2000). BOOSTEXTER : a boosting-based system for text categorization. Machine Learning, 39(2/3): PP. 135–168.

**[Schütze et al., 1995]** H. Schütze, D.A. Hull et J.O. Pedersen (1995). A comparison of classifiers and document representations for the routing problem. In Fox, E. A., Ingwersen, P., et Fidel, R., editors, Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval, PP. 229–237, Seattle, US. ACM Press, New York, US.

**[Sebastiani, 2002]** F. Sebastiani (2002). Machine learning in automated text categorization. ACM Computing Surveys, 34(1): PP. 1–47.

**[Siam et al., 2006]** M.M. Siam, Z.T. Fayed et M. B. Habib (2006). An intelligent system for Arabic text categorization. IJICIS, Vol.6, No. 1, JANUARY 2006

**[Singhal et al., 1997]** A. Singhal, M. Mitra et C. Buckley (1997). Learning routing queries in a query zone. In Proceedings of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval, PP. 25–32, Philadelphia, US.

**[Stricker, 2000]** M. Stricker (2000). Réseaux de neurones pour le traitement automatique du langage : conception et réalisation de filtres d'information. PhD thesis, Université Pierre et Marie Curie - Paris VI, Paris.

**[Teytaud et Jalam, 2001]** O. Teytaud et R. Jalam (2001). Kernel based text categorization. In Proceeding of IJCNN-01, 12th International Joint Conference on Neural Networks, Washington, US. IEEE Computer Society Press, Los Alamitos, US.

**[Tzeras et Hartmann, 1993]** K. Tzeras et S. Hartmann (1993). Automatic indexing based on Bayesian inference networks. In Korfhage, R., Rasmussen, E., et Willett, P., editors, Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval, PP. 22–34, Pittsburgh, US. ACM Press, New York, US.

**[Van Rijsbergen, 1979]** C.J. Van Rijsbergen (1979). Information Retrieval, 2nd edition. Dept. of Computer Science, University of Glasgow.

**[Vapnik et Cortes, 1995]** V. Vapnik et C. Cortes (1995). Support vector networks. Machine Learning, 20: PP. 273–297.

**[Vinot et Yvon, 2002]** R. Vinot et F. Yvon (2002). Quand simplicité rime avec efficacité: analyse d'un catégoriseur de textes. In Colloque International sur la Fouille de Texte (CIFT'02), Hammamet, Tunisie.

**[Wiener et al., 1995]** E.D. Wiener, J.O. Pedersen et A.S. Weigend (1995). A neural network approach to topic spotting. In Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval, PP. 317–332, Las Vegas, US.

**[Xu et al., 2002]** J. Xu, A. Fraser et R. Weischedel (2002). Empirical Studies in Strategies for Arabic Retrieval, Proceedings of the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2002), August 11-15, 2002, PP. 269-274.

**[Yang, 1997]** Y. Yang (1999). An evaluation of statistical approaches to text categorization. Information Retrieval, 1(1/2): PP. 69–90.

**[Yang et Chute, 1994]** Y. Yang et C.G. Chute (1994). An example-based mapping method for text categorization and retrieval. ACM Transactions on Information Systems, 12(3): PP. 252–277.

**[Yang et Liu, 1999]** Y. Yang et X. Liu (1999). A re-examination of text categorization methods. In Hearst, M. A., Gey, F., et Tong, R., editors, Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval, PP. 42–49, Berkeley, US. ACM Press, New York, US.

**[Yang et Pedersen, 1997]** Y. Yang et J.O. Pedersen (1997). A comparative study on feature selection in text categorization. In Fisher, D. H., editor, Proceedings of ICML-97, 14th International Conference on Machine Learning, PP. 412–420, Nashville, US. Morgan Kaufmann Publishers, San Francisco, US

# **Annexe**

## Un extrait d'un texte du corpus

إدوارد سعيد شغل المثقفين في العالم منذ ربع قرن!  
د. محمد الأحمرى

compiled by  
Latifa Al-Sulaiti

مؤسسة الوقف الإسلامي Saudi Arabia,  
Saudi Arabia  
2003

created in machine-readable form in <http://www.lahaonline.com>  
Texts collected for use in the Corpus of Contemporary Arabic project, June, 2003  
Whole text of 1330 words copied from the site  
October 2, 2003  
Riyadh, Saudi Arabia

Arabic  
print; written

إدوارد سعيد شغل المثقفين في العالم منذ ربع قرن!  
ذات يوم في ممرات الجامعة لقيت شابا يحمل لوحا خشبيا للدعاية على صدره يقابله آخر على ظهره، مكتوب على الجهتين إعلان عن محاضرة لـ "إدوارد سعيد"، فتأكدت منه عن حقيقة قدوم المحاضر فقال إن اللقاء هو فقط عرض لفيلم مسجل عن محاضرة له سابقة، وأنهم يتوقعون جمهورا كبيرا لها!!  
ثم التقيت بمبشر قضى أكثر روح من حياته في بلاد الشام يدأب لإنشاء الكنائس وتوجيه الشباب العربي للنصرانية، من لبنان إلى القدس، لا يفتؤ يحاول.. عسى ولعل، ولكنه كان يسبح ضد التيار، وقد ربطت السياسة بين موقفينا، فهو يناصر بحكم المعاشية واختلاف الموقف قضايا العرب في أمريكا، وهو كاثوليكي، ومن أول من سمعت منه وصف المتعصبين البروتستانت لإسرائيل بـ "الصهاينة"، وكان يأنف من مجرد ذكرهم، ويраهم طائفة تسخر بلاده لمصلحة إسرائيل، ويهز أريحيته للحديث ذكر رجليين فقط في تاريخ العرب الثقافي الحديث، هما: إدوارد سعيد ومحمد حسين فضل الله؛ لأنه يراهما "متطرفين جدا في الذكاء!"

ثم يتحدث بمزاج شاعري عن تلك الليلة العامرة التي شهد فيها في مؤتمر المستشرقين في شيكاغو مواجهة تاريخية بين إدوارد سعيد وبرنارد لويس، بعد ضجة كتاب الاستشراق لإدوارد سعيد، وقد أدار المواجهة مؤرخ شهير هو "ويليام مكنيل"، مؤلف كتب مهمة في التاريخ، منها: "صعود الغرب".

وكان مكنيل وقتها رئيس رابطة مؤرخي أمريكا - كما فهمت - من شاهد الحادثة. ثم يصف كيف قدم مكنيل الرجلين الخصمين، فكانت له صرامته، إذ حدد عشرين دقيقة فقط لكل منهما أن يقدم رأيه وحجته، ثم وقت للرد، ثم يجيبان عن أسئلة قليلة من القاعة، والمناقشة في وقت محدد.. يضاء قرب نهاية كل سياق نور أصفر موح بنهاية الوقت، ثم ضوء أحمر يوقف المتحدث. قال: كان اللقاء متعة ووجبة ثقافية رائعة، تساوي أن يسافر لمشاهدتها المهتم، وقد استطاع إدوارد - وهو الأديب المتمكن - أن يجعل الحاضرين يسخرون من لويس، ويأسفون لضعفه أمام العربي الذي استولى على المجال دون مناوئ قادر على المواجهة. قال: لقد كانت ليلة جميلة أن يستمع نخبة المثقفين للرجلين اليساريين الذين شقا معسكر الثقافة المتعلقة بالعالم العربي والإسلامي، فأما لويس الذي أصبح صهيونيا، فقد كان يساريا ثم ارتد - كما يقولون - ليمثل دور متطرفي الصهاينة، ويسخر حياته الباقية الطويلة في الإفساد الثقافي إلى اليوم.. إفساد الرؤية الغربية للعالم الإسلامي، وإفساد الحكم على ثقافتنا، وتشويهها قدر طاقته، والتهوين منها، ونشر الرعب في قلوب الغربيين منا.

وبين شق آخر هو بقية اليسار، من المتمردين على ثقافة اليمين المتطرف الصهيوني، وقد قاد الموقف اليساري في هذه القضايا إدوارد سعيد، المثقف البقظ، وحف به قوم لا يقل بعضهم شهرة عنه، رأت فيه معلما لنهجها، ومدافعا عن المظلومين. ولد في القدس عام 1935م، وتلقى بداية تعليمه في القاهرة، ثم أكمل في أمريكا في أكثر من جامعة، وانتهى به الأمر أستاذًا للأدب الإنجليزي في جامعة كولومبيا في مدينة نيويورك. لقيته مرة واحدة في قاعة "ويستمنستر" في لندن يقدم مقابلة أو محاورة عن المنطقة، وعن جوانب من حياته، وكانت امتلأت القاعة قبل بدء اللقاء، وأثناء المحاضرة.. كان بجانيبي أستاذ في أحد الجامعات اللندنية، بيده مسجل، أما أنا فقد كتبت معظم اللقاء، وانتهت الأوراق التي بيدي، فطلبت منه مزيدا من الورق، إذ كان يحمل رزمة من الأوراق ربما ليكتب لو لم يعمل المسجل، ولما انتهى اللقاء، قلت: كانت المحاضرة "جيدة"، فغضب

جاري وقال :ماذا تقول؟! إنها "عظيمة" وليست مجرد جيدة ..أدركت من استنكار جاري أنني هضمت مستوى اللقاء , وشعرت أنني قصرت في حق المحاضر.

وقد كانت محاضراته حقا كما وصفها جاري , إذ يمتلك أسلوبا حواريا مقنعا .ثم نزلت من الطابق الأعلى لأجد الناس يصطفون لالتقاط الصور معه ,ولما خف الزحام صافحته وشكرته وذهبت ,وبدأ حديثه مع كاتب اسكتلندي يتحدثان في كتاب ألفه الأخير عن الموسيقى ,ولأن إدوارد حجة في الموسيقى كما يكتبون عنه ,فعندما يقيمون عمله وإنتاجه يضعون إلى جانب مكانته في السياسة والأدب الإشادة بمعرفته في هذا الميدان وتقدمه بين الناقدن لهذا الفن أيضا .واحتفل أخيرا بمرور ربع قرن على صدور كتابه المهم "الاستشراق" وهو الكتاب الذي أخرج مؤلفه للناس من مجرد ناقد أدبي أكاديمي ,ومن ناطق سياسي ذي صوت عال للمسألة الفلسطينية إلى مفكر عالمي مضاد للاستعمار وثقافته ,شق الكتاب للناس دربا جديدا في التعرف على ظاهرة الاستشراق ,وما يؤسس لها ويلحق بها من ثقافة ,وكان لتلامذته دور مهم في تقديمه وتأليف الكتب عنه ,وله تلميذة من الهند أصبحت تدرّس في الجامعة نفسها ,كتبت عنه وجمعت له كتابا مهما ,وقد اهتمت بموضوع لغة الاستعمار ,وكتبت رسالة علمية لها أهميتها في هذا الجانب ,مما أغرى المثقفين الإسرائيليين بدعوتها وتكريمها ,تحفيقا من حملتها عليهم ,أو حرفا لها عن طريقها ,وإغاضة لأستاذها ,وهكذا يقتنصون المثقفين الجدد!

اشتهر إدوارد سعيد ناقدا قبل أن يفاجئ الناس بـ "الاستشراق" ,هل كان مبدعا في كتابه؟ يقول خصومه :لا ,فلم يزد على أن جدد أقوال الناقدن ,ونصوص الأدباء والرحالة ,ومواقف السياسيين ; ليصوغ منها حملة شرسة على الاستعمار وأربابه . وربما رآها آخرون حملة ناقمة ,عمياء ,دل عليها كتابان تاليان هما كتاب "تغطية الإسلام" و"ثقافة الإمبريالية" ,لقد جار عليه خصومه ,وذلك ردهم على لذعائه ولوذعيته .كتاب الاستشراق فيه تطبيق لنظريات ميشيل فوكوه ,ولا يضره ذلك ففلسفة فوكوه وجدت تطبيقا ميدانيا لها ,في مسائل المعرفة وعلاقتها بالسلطة ,وإعجاب إدوارد بفوكوه كبير ,حتى إنه اهتم بحضور محاضراته ,وربما حضر درسه الافتتاحي .وكان كتاب إدوارد سعيد قوة لكتابات فوكوه ,وتطبيقا للنظرية تجاوز بالتطبيق والتفريع فكرة "صاحب نظرية المعرفة سلطة" أو "المعرفة تستتبع السلطة" وأسلوبه العالي نفخ الحياة في جفاف التنظير.

وقد غزاه السرطان وأرهقه ,وفي مقدمة هيكل لكتاب إدوارد عن أوصلو ومحادثات السلام ,سلاه هيكل عن السرطان بأن الأمراض تختار أجسادها .كان شجاعا ,تميز عن مثقفي الشرق بكشف زيف الغرب واستغلاله للمعرفة ,وسيلة للاستعمار , وكان شجاعا بتجاوز عقدة خنوع المثقف وحرصه وتبعيته وتهافته ,وخالف نهج مثقفي العرب في المهجر الذين يلونون بالصمت خوفا من نفوذ اليهود ,وحرصا على مواقعهم الوظيفية ,انضم لمنظمة التحرير بقناعة.

قامت الدنيا عليه في جامعة ينفذ فيها يهود نيويورك ,ولم يبال بعرائضهم المطالبة بطرده ,يقول :لم أملك إلا أن أفق على الحدود اللبنانية وأرسي المحتلين بحجر ,والتقطت الصورة له وهو يرحمهم.

وبدأت حملة جديدة طالب فيها أساتذة جامعتة بطرده .ومن قبل ذلك لاحقه خصومه اليهود ينكرون كونه فلسطينيا مولودا في القدس ,وذهب وفد يستقصي تاريخه ,ونشرت مجلة "كومنتري" الأدبية اليهودية ملفا بذلك ,فزادت هذه المطاردة من ذبوع قضيته .وخالف عرفات وصلحه بشجاعة ,وخالف آراء كثير من مثقفي العرب في الموقف من مذابح هتلر لليهود ,فهو يصدق حدوثها ,ويستنكر قول المنكرين ,ويتعاطف مع ضحاياها ,وله في المسألة الفلسطينية رأي جريء حيث يطالب بدولة ديموقراطية واحدة في فلسطين للجميع ,تحكمها الأغلبية وترعى حق الأقلية ,ويخالف من يقول بدولتين.

موقفه من قضايا المسلمين موقف منصف غالبا ,وتغيظ مواقفه التيار الوصولي في الثقافة العربية المعاصرة; لأنه كان شديد القسوة على من يسميه بالمثقف الخائن ,ويكثر من تكرار استخدام أحد الكتب الفرنسية المثيرة التي كتبت مطولا عن خيانة المثقفين وتبعيتهم .وهنا نلاحظ ذلك الجانب المكروه للوصوليين ,وسوطه المرفوع الذي يجلد به ظهورهم ,لم يكن يملك ما يخاف عليه ,ولم يزد مرضه إلا تخففا وشجاعة ,وقد سأل أحد المعلقين عن سر مضاعفة جهده ,فأكد خطر معاناته لسرطان الدم ,والمصاب بالسرطان لا وقت لديه ,كان إذا حل على برنامج "تشارلي روز" تهاتفنا بالخبر; لأننا سنقضي ساعة من المتعة والفكرة ,وبراعة المواجهة ,فشجعان العقول قليل ..وهو من القلة التي تستطيع أن تكشف حدود الحرية الفكرية في أمريكا ,وما أصعب أن ترى حدود الحرية.

خسر المسلمون والعرب مدافعا فصيحا عن قضاياهم ,ومهتما بارزا بقضية فلسطين .لقد كان رجلا واحدا ,ولكنه كان جهازا إعلاميا ثقافيا مؤثرا ,أكثر مما أثرت الدول العربية في التوعية بالمسألة الفلسطينية في الغرب ,وكان مطلعا ومتابعا للأحداث ومعلقا فطنا ,ومتحدثا أسرا ,يفوق أسلوب حديثه أسلوب كتابته .كتبه القديمة والحديثة دائما معروضة في طبعات جديدة ,لا ينتهي حولها الجدل ,كان صيادا وعارضا للفكرة ,مجيدا ومبدعا في اعتراضه ومؤثرا .عمقه في أدب الإنجليزية لا يبارى , وأجاد الفرنسية ,ثم عاد لبيروت وتمكن من العربية.

ولكم وددت أن يجد القارئ العربي كتاب "الاستشراق" بترجمة عربية جيدة ,فإن مترجمه أعجمه ,وأضر كمال أبو ديب بكتاباتة ,ولو قارنت هذه الترجمات مع ترجمات كتب أخرى مثل كتاب "صور المثقف" أو كتاب المقابلة الطويلة معه التي أجراها بارسيمان; لرأيت فرق الطريقتين .لقد كان لسان العرب الحر ,محاضرا ومحاورا ,أما بعد حسم معركته مع لويس فقد قل من فكر في مواجهته.

ودّع الناس مفكرا ومناضلا ثقافيا لا بديل له ,ولا مقارب ,وبقيت آثاره مدرسة في النزاهة ومكافحة الظلم ,كان يعتقد الشك - كما وصف نفسه ..ولو كان مسلما لترحمنا عليه