République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Amar Telidji - Laghouat

Faculté des sciences
Département d'informatique et Mathématiques
Thèse
présentée pour l'obtention du diplôme de Doctorat en Science
Spécialité : Informatique

Thème :

Noyaux rationnels pour la classification des données non structurées : Documents Web en arabe

présentée par : **Attia NEHAR**

Soutenue le 06 Mai 2017 devant le jury composé de :

M. B. Yagoubi	Professeur	Université de Laghouat	Président
A. Guessoum	Professeur	USTHB Alger	Examinateur
M. Abbas	Directeur de recherche	CRSTDLA Alger	Examinateur
Y. Ouinten	Maître de conférences	Université de Laghouat	Examinateur
D. Ziadi	Professeur	Université de Rouen	Directeur
H. Cherroun	Professeur	Université de Laghouat	Co-directeur

Résumé

La classification de documents a pour objectif d'assigner, d'une manière efficace, un document à une classe d'un ensemble prédéfini de classes. Pour la langue arabe, cette tâche présente des particularités liées à la langue. Des opérations telles que la racinisation et l'extraction de radicaux doivent se faire d'une manière efficace. La représentation des documents sous forme vectorielle permet d'appliquer des algorithmes conventionnels d'apprentissage. Cependant, elle engendre une perte d'information liées à l'ordre et la co-occurrence des mots et phrases. Une solution à ce problème consiste à utiliser des N-grammes (avec $N \geq 2$) à la place de termes simples isolés, ou modèle de sac-à-mots. Cette approche se fonde sur l'hypothèse qu'un ensemble de termes contigus peut capter la similarité entre documents mieux que des termes simples isolés. Notre thèse s'inscrit dans le cadre de la classification de documents en arabe.

L'objectif de notre travail a été d'une part, de proposer une technique efficace d'extraction de radicaux des mots. D'autre part, de proposer une plateforme unifiée pour analyser l'effet de l'extraction de radicaux et la taille des N-grammes sur la performance des systèmes de classification de documents en arabe.

Les résultats ont montré que l'utilisation des transducteurs pour l'extraction de radicaux constitue un choix naturel, vue leur capacité à modéliser la forme flexionnelle des mots en langue arabe. De plus, l'extraction de racines améliore légèrement la qualité des classificateurs en termes d'exactitude, rappel et F1, mais elle diminue légèrement la précision. Les classificateurs basés sur le noyau 3-grammes ont atteint les meilleurs résultats. Pour le niveau N-gramme terme, les résultats ont démontré que l'insertion des trous n'améliore pas les performances.

Mots clés : Classification de documents en arabe, N-grammes, Extraction de radicaux, Noyaux rationnels, Transducteurs.

Abstract

The goal of document classification is to efficiently assign a document to a class picked out from a predefined set of classes. For the Arabic language, this task has features peculiar to the language. Operations such as stemming and root extraction must be done in an efficient way. Vectorial representation of documents makes it possible to apply conventional machine learning algorithms. However, it gives rise to information loss related to the order and co-occurrence of words and phrases. One solution to this issue is to use N-grams (with $N \geq 2$) in place of simple isolated terms, or bag of words model. This approach is based on the assumption that a set of contiguous terms can capture the similarity between documents better than isolated terms. This thesis is part of the classification of documents in Arabic.

The aim of our work was, on the one hand, to propose an efficient technique for extracting radicals from words. On the other hand, to propose a unified platform to analyze the effect of root extraction and the size of N-grams on the performance of document classification systems in Arabic.

The results showed that the use of transducers for root extraction is a natural choice, given their ability to model the inflectional form of words in Arabic. In addition, root extraction slightly improves the quality of classifiers in terms of accuracy, recall and F1, but it slightly decreases precision. Classifiers based on the 3-grams kernel have achieved the best results. For the word N-grammes level, the results showed that gaps insertion did not improve performance.

Keywords : Arabic Text Classification, N-grams, Root Extraction, Rational Kernels, Transducers.

Remerciements

Je souhaite remercier en premier lieu Mme ***Hadda Cherroun***, Professeur des Universités, pour la confiance qu'elle m'a accordé en acceptant la direction de mes travaux, et pour m'avoir accepté au sein de son équipe. Je lui suis également reconnaissant pour le temps conséquent qu'elle m'a accordé, ses qualités pédagogiques et scientifiques, sa franchise et sa sympathie. J'ai beaucoup appris à ses côtés et je lui adresse ma gratitude pour tout cela.

J'adresse de chaleureux remerciements à mon co-encadrant de thèse, M. ***Ziadi Djelloul***, Professeur des Universités, pour son attention de tout instant sur mes travaux, pour ses conseils avisés et son écoute qui ont été prépondérants pour la bonne réussite de cette thèse. Son dynamisme et sa disponibilité ont été des éléments moteurs pour moi. J'ai pris un grand plaisir à travailler avec lui.

Je voudrais remercier aussi M. ***Ahmed Guessoum***, Professeur à l'université des sciences et technologies (Alger), M. ***Abbas Mourad***, Maitre de recherche au centre de recherche scientifique et technique pour le développement de la langue arabe (Alger), et M. ***Youcef Ouinten***, Maitre des conférences à l'université Amar Télidji (Laghouat), pour l'honneur qu'ils m'ont fait en acceptant d'être membres de mon jury de thèse. Je tiens à leur assurer de ma profonde reconnaissance pour l'intérêt qu'ils portent à ce travail.

J'associe à ces remerciements M. ***Mohamed Bachir Yagoubi***, Professeur à l'université Amar Télidji, pour avoir accepté de présider le jury et pour l'intérêt qu'il a porté à mon travail.

Enfin, je tiens à remercier tous les membres de l'équipe de modélisation et optimisation dans les systèmes informatiques, pour leur aide, leur bonne humeur et les bons moments que nous avons partagé.

Table des matières

Table des figures

Liste des tableaux

Introduction générale

Contexte

L'ÉVOLUTION exponentielle d'Internet et l'utilisation de grandes masses de données textuelles ont conduit à révéler la classification automatique des documents au grand jour, notamment par le biais des outils basés sur l'apprentissage automatique [Sebastiani and Ricerche, 2002],[Sebastiani, 2006]. L'importance grandissante de la langue arabe a suscité le développement d'outils et de techniques automatiques spécifiques afin de permettre son traitement automatique. Ce besoin n'est pas négligeable. En novembre 2015[1], la proportion d'utilisateurs d'Internet naviguant en langue arabe était de 5.0 %, venant à la quatrième position après l'anglais, le chinois et l'espagnol. La proportion de sites publiés en langue arabe était de 0.8 % selon la même source. Les sites qui doivent publier un contenu textuel se voient recevoir et traiter des centaines d'articles quotidiennement. Ce traitement, qui consiste essentiellement à classer ces articles en un nombre de catégories, ne peut être fait manuellement, d'où vient la nécessité des systèmes de classification de documents.

La classification de documents (CD) consiste à affecter, d'une manière automatique, un ensemble de documents à un ensemble prédéfini de classes ou catégories [Sebastiani and Ricerche, 2002]. Un système de classification de documents comporte les étapes suivantes :

1. **Le pré-traitement** : dans cette étape, le texte subit un certain nombre d'opérations de normalisation orthographiques, telles que la suppression des lettres de ponctuation, nombres, mots vides, caractères spéciaux et tout caractère non alphabétique. Certaines opérations liées à la langue peuvent être aussi effectuées à ce niveau, telles que la lemmatisation et l'extraction des racines.
2. **Représentation** : les documents sont transformés en une forme spécifique, souvent vectorielle, par l'extraction d'un certain ensemble de caractéristiques. Par exemple, la technique de sac à mots a été la première à être utilisée. Elle consiste à considérer tout mot apparaissant au moins une fois

1. http ://www.internetworldstats.com

dans l'ensemble des documents comme une caractéristique ou dimension dans le vecteur de représentation.

3. **Apprentissage** : le but de cette étape est de faire apprendre au système comment associer une catégorie à un document. Les algorithmes supervisés utilisés reposent sur des mesures de similarité entre les documents pour décider si deux documents sont similaires ou pas.

La classification de documents pour la langue arabe a suscité un grand intérêt dans les dernières quinze années à cause de l'augmentation du volume des documents sous forme numérique, et la nécessité qui en découle afin de les organiser. En effet, pour la langue arabe, l'augmentation en usage et en importance, a fait apparaitre la nécessité d'adapter les systèmes développés pour les autres langues, pour qu'ils soient utilisables efficacement [Hmeidi et al., 2014].

Les deux premières étapes d'un système de CD, pré-traitement et représentation, nécessitent la prise en compte des particularités de la langue arabe. En effet, le pré-traitement regroupe, en plus des opérations usuelles, la diacritisation, la racinisation et l'extraction des radicaux, qui représentent un challenge pour les systèmes de classification de documents en arabe.

La racinisation et l'extraction des radicaux sont des opérations importantes dans les systèmes de classification de documents. Elles sont appliquées pour réduire la dimension des vecteurs de représentation des documents. L'extraction de radicaux [2] permet de réduire chaque mot dans le document à ses radicaux. Par contre, la racinisation [3] permet de supprimer juste les préfixes et suffixes du mot [Aljlayl and Frieder, 2002].

Les techniques d'extraction de radicaux développées ont été proposées initialement dans le contexte de la recherche d'information [Larkey et al., 2002]. Elles peuvent être classées principalement en deux catégories : (i) techniques supervisées, basées sur des connaissances linguistiques ; (ii) techniques non supervisées, reposant sur des modèles statistiques de la langue ou sur l'apprentissage non supervisé. Dans les techniques supervisées, le système fait appel à des connaissances supplémentaires de la langue, telles que les dictionnaires des préfixes, suffixes, motifs ou les listes des racines et les règles linguistiques. Par contre, dans les techniques non supervisées on se passe des connaissances linguistiques, juste le texte brute est utilisé.

2. L'extraction de radicaux est connue sous le terme "root-based stemming" en anglais. 3. La racinisation est connue sous le terme "stem-based stemming" en anglais.

La représentation des documents joue aussi un rôle essentiel pour la qualité du système de classification. Elle permet d'une part de transformer les documents vers un format vectoriel, permettant l'utilisation d'algorithmes d'apprentissage conventionnels. D'autre part, elle permet de réduire la taille de ces vecteurs en sélectionnant un sous ensemble pertinent de caractéristiques.

Les N-grammes (niveau caractères) ont été exploitées pour représenter les documents en arabe pour la première fois par [Khreisat, 2009] en utilisant les mots surfaces. Les expérimentations ont été menées sur un corpus maison (en anglais : "in-house corpus") et l'effet de l'extraction des radicaux et de la racinisation n'ont pas été analysés. En effet, le fait de remonter aux racines des mots, permet de réduire plusieurs formes surfaces de mots à une même racine. Ceci peut améliorer la fonction de similarité entre documents et, par conséquence, la qualité du système de classification.

Pour illustration, prenons comme exemple deux documents (d_1, d_2) contenant une seule phrase chacun. Les deux phrases expriment la même idée de deux manières différentes. Les documents $d_1^{'}, d_2^{'}$ sont obtenus en appliquant les opérations de pré-traitement sur d_1, d_2 respectivement, et $d_1^{''}, d_2^{''}$ sont obtenus en appliquant l'extraction des radicaux sur les mots des documents $d_1^{'}, d_2^{'}$:

d_1	"اذا كمل العقل نقص الكلام"	d_2	"من كمل عقله نقص كلامه"
$d_1^{'}$	"كمل العقل نقص الكلام"	$d_2^{'}$	"كمل عقله نقص كلامه"
$d_1^{''}$	"كمل عقل نقص كلم"	$d_2^{''}$	"كمل عقل نقص كلم"

Le cosinus de deux vecteurs de représentation de documents d_i et d_j, utilisant les 3-grammes niveau caractères, est donnée comme suit :

$$\text{Cos}(V_{d_i}, V_{d_j}) = \frac{\text{nombre (3-grammes communs entre } d_i \text{ et } d_j)}{\sqrt{\text{nombre (3-grammes dans } d_i) \times \text{nombre (3-grammes dans } d_j)}} \quad (1)$$

En appliquant cette formule aux documents $d_1^{'}$ et $d_2^{'}$ on obtient :

$$\text{Cos}(V_{d_1^{'}}, V_{d_2^{'}}) = 0.46$$

En appliquant l'extraction des radicaux, cette similarité devient plus grande du fait qu'on réduit les mots à leurs radicaux :

$$\text{Cos}(V_{d_1^{''}}, V_{d_2^{''}}) = 1$$

De plus, nous pouvons remarquer que peu de travaux ont exploré les N-grammes (niveau termes) pour la classification de documents en arabe [Al-Thubaity et al., 2015]. Le niveau termes des N-grammes permet de considérer des suites

de mots, c'est-à-dire des phrases entières comme caractéristiques des documents. Ceci permet de caractériser les documents non pas par leurs mots individuels mais par des groupes de mots apparaissant en conjonction. Le tableau suivant donne des exemples de mots qui apparaissent souvent ensemble dans des documents de la même catégorie.

Catégorie	Mots en co-occurrence
Culture	محاضرة بعنوان , حرم شريف , حدود شرقية , مهرجان شعبي
Économie	مجلس ادارة , داو جونس , ارتفع مؤشر , صناعة تجارة
Générale	رئيس مجلس , مدير عام , تحت رعاية
Politique	شرق اوسط , رئيس وزراء , امم متحدة , دول عربية , وزير خارجية
Sociale	عيد فطر , توزيع جوائز , جمعية خيرية
Sport	رئيس الاتحاد , رعاية الشباب , الشباب الرياضة , كرة قدم

Exemples de mots qui apparaissent en conjonction dans les catégories.

Objectifs et contributions

Dans cette thèse, nous abordons le problème de la Classification de Documents en Arabe (CDA). Nous proposons une nouvelle approche, basée sur une représentation des documents par des machines à états finis : les transducteurs. Cette nouvelle représentation permettra de tirer partie des méthodes à noyaux, notamment les SVM et des noyaux rationnels, dans une plateforme unifiée. En effet, ces transducteurs sont aussi déployés pour proposer une nouvelle technique d'extraction de radicaux des mots en arabe.

Nous résumons notre apport dans cette thèse comme suit :

1. Notre première contribution est un travail de synthèse et de classification. En effet, nous commençons par présenter un état de l'art des différentes techniques de racinisation et d'extraction de radicaux proposées dans la littérature, ainsi qu'une analyse détaillée des systèmes de CDA proposés jusqu'à 2015. Nous établissons une taxonomie des systèmes de CDA selon plusieurs critères.
2. Concernant le problème de l'extraction des radicaux, nous proposons une nouvelle technique qui fait appel aux transducteurs pour modéliser les pré-

fixes, les motifs des mots et les suffixes [Nehar et al., 2012]. Cette technique consiste à extraire la racine d'un mot en deux phases : (i) elle commence par générer les racines potentielles du mot, c'est-à-dire toutes les racines qui répondent aux formes préfixe-modèle-suffixe modélisées ; (ii) puis elle choisit la racine la plus probable, en se basant sur une étude statistique des racines. Les lettres radicaux sont ensuite identifiées selon le modèle choisi.

Cette technique est analysée et comparée, dans [Nehar et al., 2016], avec d'autres techniques supervisées et non supervisés, en l'occurrence celles de [Khoja and Garside, 1999] et[Al-Serhan et al., 2003].

3. Pour le problème de classification de documents en arabe, nous proposons une plateforme unifiée [Nehar et al., 2013] qui se base sur l'utilisation des : (i) transducteurs pour la représentation des documents. Ceci permettra, pour le niveau des termes [Nehar et al., 2014], de capturer l'ordre et la co-occurrence des mots ; (ii) noyaux rationnels pour calculer la similarité entre ces documents.

   Dans cette plateforme, plusieurs configurations ont été explorées et testées pour répondre aux objectifs suivants :
   - Classification avec ou sans extraction des radicaux.
   - Effet de la technique d'extraction des radicaux adoptée.
   - La granularité de représentation utilisée (niveau caractères ou mots).
   - Choix de la taille du N-gramme (sous mots).
   - N-grammes avec ou sans trous (pour les N-grammes niveau termes, ou sous séquences de mots).

Organisation de la thèse

Ce manuscrit est organisé comme suit :
Les trois premiers chapitres sont consacrés au cadre théorique de notre travail. Dans le **premier chapitre**, nous introduisons les notions de base de l'apprentissage supervisé et plus particulièrement de la classification de documents.

Le **second chapitre** est consacré à la présentation des méthodes à noyaux et en particulier les noyaux rationnels. Dans un premier temps, nous donnons un aperçu général des méthodes à noyaux. Ensuite, nous introduisons la notion de noyau et son lien avec la mesure de similarité ou distance. Nous illustrons ces méthodes à noyaux par une des techniques les plus utilisées, à savoir : les Séparateurs à Vastes Marges (SVM). Dans un second temps, nous abordons la notion de noyaux *rationnels*, en expliquant comment ils sont matérialisés par les

transducteurs de mots. Nous exposons les propriétés principales ainsi que des exemples des noyaux rationnels.

Dans le **troisième chapitre** nous dressons un état de l'art des techniques de racinisation et d'extraction de radicaux, ainsi que la synthèse et taxonomie des différents systèmes de classification de documents en arabe. Nous soulignerons dans ce chapitre les difficultés de la CDA.

Les chapitres quatre et cinq sont dédiés à notre contribution dans le cadre de cette thèse. Dans le **quatrième chapitre**, nous décrirons notre approche pour l'extraction des radicaux. Cette approche est implémentée et analysée en vue d'être comparée avec d'autres techniques supervisées et non supervisées.

Dans le **cinquième chapitre** nous présentons notre plateforme de classification de documents en arabe. Nous expliquons comment représenter les documents sous forme de transducteurs pour pouvoir utiliser les noyaux rationnels dans l'apprentissage. Ce chapitre répond principalement à deux objectifs. Le premier objectif est d'analyser l'effet de l'extraction de radicaux sur les systèmes de CDA utilisant les N-grammes de niveau caractères. Le second objectif est de montrer, pour les N-grammes niveau termes, l'effet de considérer l'ordre et la co-occurrence des mots dans la représentation des documents.

nous rappelons, dans la **conclusion**, les différents travaux effectués et résultats obtenus. Nous proposons des perspectives et des pistes de recherche ouvertes à l'issue de ces travaux.

Chapitre 1

Généralités sur la classification de documents

Nous abordons, dans ce chapitre, le problème de classification de documents en général. Nous commençons par rappeler les bases théoriques de l'apprentissage et de la classification en particulier, et ce pour des données structurées présentant une uniformité dans leur structure et qui peuvent être stockées dans des tables. Puis, nous abordons le cas des données non structurées, en l'occurrence, les données textuelles. La classification de documents en langue Arabe possède des particularités intrinsèques à la langue, elle est traitée dans le chapitre 3.

1.1 Apprentissage automatique

L'apprentissage est une branche à l'intersection de plusieurs disciplines : intelligence artificielle, algorithmique et la recherche d'information. Elle concerne l'étude et la construction de systèmes qui ont la faculté d'apprendre à partir des données. Par exemple, un système peut être entrainé sur un ensemble de messages électroniques pour pouvoir distinguer les messages "Spams" des messages légitimes. Selon Arthur Samuel [Samuel, 2000], l'apprentissage (ou Machine learning en anglais) est le champs d'étude qui donne aux machines la faculté d'apprendre sans être explicitement programmées. En effet, pour certains types de problèmes, tel que le tri d'un tableau d'entiers, il est plus ou moins aisé de trouver un algorithme qui donne la solution, il s'agit juste de trouver le meilleur algorithme, nécessitant moins d'opérations ou d'espace mémoire, ou même les deux. Par contre, pour d'autres types de problèmes, par exemple le problème des spams, il n'est pas possible d'écrire un algorithme explicite. Tout ce que nous avons c'est un ensemble de messages (un ensemble de fichiers textes), et pour chaque message nous disposons de l'information "spam"/"non spam". Mais nous

ne savons pas comment transformer cette donnée d'entrée (message) en une sortie ("spam"/"non spam"). L'idée de l'apprentissage est de prendre de grandes masses de données (messages étiquetés par l'une des informations : "spam" ou "non spam" dans notre exemple) et d'apprendre les caractéristiques des messages. En d'autre termes, on demande à la machine d'extraire un algorithme pour ce type de tâche.

1.1.1 Vocabulaire et définitions

D'une manière plus formelle, Tom Mitchell [Mitchell, 1997] donne une formulation du "problème d'apprentissage bien posé" comme suit :

Définition 1.1. *Un programme est en mesure d'apprendre à partir d'une expérience E par rapport à une tâche T et une mesure de performance P, si sa performance à la tâche T, telle que mesurée par P, s'améliore avec l'expérience E.*

Par exemple, un programme qui apprend le jeu d'échec, peut améliorer sa qualité de jeu (mesurée par sa capacité à gagner des parties de jeux) à partir de l'expérience obtenue des parties de jeu contre des humains. Pour le système qui apprend à distinguer les messages légitimes des messages spam, il améliore sa qualité à les distinguer (tâche T), telle que mesurée par le pourcentage des décisions correctes (mesure P), en utilisant l'ensemble des messages pré-étiquetés (expérience E).

D'une manière générale, pour avoir un problème d'apprentissage bien posé, il faut bien identifier les trois composantes suivantes :

1. *Classe de tâche à apprendre* : elle correspond au type de la tâche à effectuer. Selon la connaissance qu'on veut faire apprendre au programme, il peut s'agir de l'un des types de problèmes suivants : apprentissage supervisé/non supervisé. On parle d'un *apprentissage supervisé* lorsque les données sont présentées avec leurs étiquettes pré-affectées par un superviseur, choisies à partir d'un ensemble d'étiquettes. Le but, alors, est de trouver une règle générale (une fonction f appelée hypothèse) qui fait associer aux nouvelles données des étiquettes. Dans le cas d'un *apprentissage non supervisé*, on ne dispose pas d'étiquettes pour les données. Le but, cette fois, est de découvrir des formes ou structures pour les données.
2. *Source d'expérience* : à partir de laquelle le système va apprendre la connaissance. Elle représente un choix important. Le type et la qualité des données

disponibles peuvent avoir un grand impact sur le succès ou l'échec du système d'apprentissage. Un des attributs importants de la source d'expérience est de savoir à quel point ces données sont représentatives de la distribution d'instances sur laquelle le système va être testé. Plus les deux ensembles (d'apprentissage et de test) suivent des distributions similaires, plus le système d'apprentissage est efficace [Abu-Mostafa et al., 2012].

3. *Mesure de performance à utiliser* : elle permet de mesurer l'amélioration de l'apprentissage de la tâche T. Selon l'application, elle est calculée en se basant sur la taille de l'échantillon d'apprentissage, le nombre de décisions correctes et le nombre de décisions incorrectes prises par le système.

1.1.2 Conception d'un système d'apprentissage

Avant d'aborder la conception d'un système d'apprentissage, nous fixons certaines idées à travers les définitions suivantes :

1. *Ensemble d'apprentissage* : ensemble noté $\mathcal{X}$ contenant des instances de données utilisées pour l'apprentissage.
2. *Attributs* : ensemble de caractéristiques, représentées généralement par un vecteur X, liées aux données d'apprentissage.
3. *Étiquettes* : ensemble, notée $\mathcal{Y}$, de valeurs ou catégories assignées aux données d'apprentissage. Selon le type de tâche à apprendre, les données peuvent avoir une caractéristique en plus, une valeur entière, par exemple pour le problème de classification, ou une valeur réelle dans le cas de régression.
4. *Ensemble de test* : ensemble de données, disjoint de l'ensemble d'apprentissage, utilisé pour évaluer la performance de l'algorithme après l'apprentissage.
5. *Ensemble d'hypothèses* : ensemble de fonctions associant l'espace des données d'apprentissage et de test à l'espace des étiquettes. Dans cet ensemble, l'algorithme d'apprentissage va chercher une fonction qui associe au mieux les deux espaces, d'apprentissage et d'étiquettes, mesurée sur les données de test.

Comme il est indiqué sur la figure 1.1, un système d'apprentissage comporte les composantes suivantes : un ensemble de données d'apprentissage, un ensemble d'hypothèses et un algorithme d'apprentissage [Abu-Mostafa et al., 2012].

L'objectif d'un système d'apprentissage est de trouver une "meilleure" estimation g, d'une dépendance "inconnue" f entre l'entrée $\mathcal{X}$ et la sortie $\mathcal{Y}$. Cette

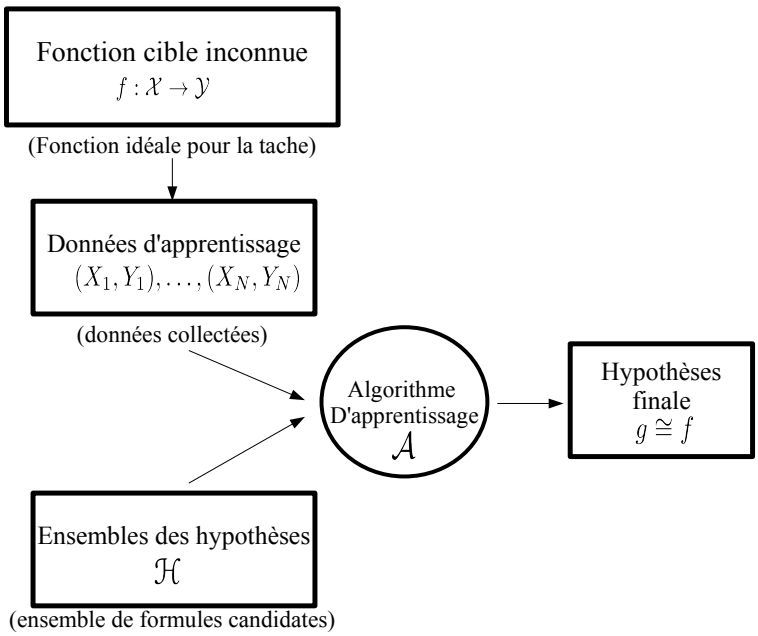


FIGURE 1.1: Composantes d'un système d'apprentissage.

dépendance, bien que inconnue, elle est perçue à travers les données d'apprentissage. La fonction d'estimation g, ou l'hypothèse finale qu'on cherche à trouver, est tirée par l'algorithme d'apprentissage à partir d'un espace $\mathcal{H}$ de fonctions d'hypothèses (ensemble d'hypothèses). Donc, f et g sont supposées avoir les mêmes ensembles d'entrée et sortie, et le choix de g est fait de telle sorte qu'elle aura un comportement très similaire à f sur les données d'apprentissage. Pour cela, la fonction d'estimation retenue est celle qui minimise le *risque d'erreur empirique*, i.e l'erreur cumulée commise par le choix de cette fonction, mesurée sur l'ensemble d'apprentissage. L'algorithme d'apprentissage est le module chargé de trouver cette fonction d'estimation. Il reçoit en entrée des données d'apprentissage et génère en sortie une fonction d'estimation, qu'il tire à partir d'un modèle prédéfini de formules (ensemble d'hypothèses $\mathcal{H}$).

Le choix d'un sous ensemble de $\mathcal{H}$ et d'un type d'algorithme d'apprentissage définissent le *modèle d'apprentissage* retenu par le système. En effet, le sous ensemble de fonctions d'hypothèses représente le type de fonction qu'on désire apprendre, et l'algorithme d'apprentissage indique comment l'apprendre.

Le processus d'apprentissage passe par trois étapes : Pré-traitement des données, apprentissage de g et évaluation de performances :

1. *Pré-traitement* : Souvent, les données d'apprentissage souffrent, dans leur forme brute, de plusieurs types d'anomalies. Elles risquent d'être incomplètes, erronées, ou même bruitées. Le but de cette étape est de traiter ces anomalies en procédant à des techniques telles que le nettoyage, traitement des données manquantes ou aberrantes, ou même faire des transformations sur ces données.
2. *Apprentissage* : Dans cette étape, et selon le type de la tâche, l'algorithme d'apprentissage peut choisir une hypothèse g de l'ensemble des hypothèses $\mathcal{H}$. La manière de choisir cette hypothèse définit l'algorithme d'apprentissage. L'hypothèse choisie doit refléter au mieux la structure des données d'apprentissage, i.e elle minimise le risque d'erreur empirique, tout en étant généralisable à des données autres que celles de l'apprentissage.
3. *Évaluation des performances* : Avant le déploiement du système d'apprentissage, il est très important d'évaluer le modèle généré. Le but est de mesurer sa qualité et son efficacité à prendre des futures décisions sur des données inconnues. Selon le type de la tâche considérée, il existe plusieurs techniques pour faire l'évaluation. L'ensemble des données est d'habitude divisé en deux parties : données d'apprentissage et données de test. Le scénario le plus simple est de choisir aléatoirement une partie des données pour l'utiliser dans l'apprentissage, et la partie qui reste sera utilisée pour le test. Par exemple, un découpage de $(80\% - 20\%)$ est très souvent adopté. Avec la technique de *validation croisée*, l'ensemble des données est segmenté en n parties, et l'apprentissage se fera n fois en excluant à chaque fois une partie, qui fera l'objet de test. Plusieurs mesures de performances ont été proposées dans la littérature [Sokolova and Lapalme, 2009].

1.1.3 Classification automatique

La classification est une tâche permettant de choisir une étiquette correcte pour une donnée d'entrée. L'ensemble d'apprentissage se présente avec des données correctement étiquetées d'avance par un expert (superviseur), cette tâche alors fait partie de l'apprentissage supervisé. A titre d'exemples, les tâches suivantes sont des problèmes de classification :

— Décider si un mail est spam ou pas.
— Décider si une demande de prêt auprès d'une banque est risquée ou pas.

— Déterminer si une transaction d'une carte de crédit est frauduleuse.
— Affecter un article à l'une des catégories d'un journal telles que : "Sports", "Sciences" ou "Politique".
— Décider si une tumeur est bénigne ou maligne.

On peut formaliser le problème de la classification comme suit :

Définition 1.2. *La Classification est une tâche consistant à trouver une application* $c : \mathcal{X} \longrightarrow \mathcal{Y}$ *permettant d'assigner une étiquette* $y \in \mathcal{Y}$ *à chaque instance* $x \in \mathcal{X}$.

En général, le nombre d'étiquettes ou classes est relativement petit, mais il peut atteindre une taille importante pour des tâches difficiles telles que la classification de documents ou la reconnaissance de formes. Dans le cas où les étiquettes sont représentées par un ensemble contenant juste deux éléments (par exemple $\mathcal{Y} = \{0, 1\}$ ou $\mathcal{Y} = \{-1, 1\}$) on parle alors d'une classification binaire. Le cas le plus général dans lequel $|\mathcal{Y}| > 2$ est connu sous l'appellation de multi-classification.

La classification ainsi définie permet d'attribuer une seule étiquette à la fois à une instance d'entrée. D'autres variantes de la classification existent. Par exemple, la classification multi-étiquettes permet d'affecter plus d'une classe au même exemple d'entrée.

La classification est une tâche composée de deux phases :

1. Construction du modèle : c'est la phase d'apprentissage. L'algorithme d'apprentissage construit un classificateur c en analysant l'ensemble d'apprentissage. Cette phase inclut plusieurs opérations telles que pré-traitement, la représentation et application de l'algorithme de construction du modèle, et l'évaluation du modèle.
2. Prédiction : dans cette phase, le classificateur issu de la phase précédente est utilisé pour prédire la classe d'une instance de données inconnue. Les opérations de pré-traitement et représentation, appliquées sur l'ensemble d'apprentissage dans la première phase, sont reprises dans cette phase.

Afin d'évaluer les performances du classificateur, un ensemble de test est utilisé pour mesurer l'exactitude du classificateur, i.e le pourcentage des instances correctement classifiées par ce classificateur. Cela suppose qu'on connait d'avance les étiquettes correctes des instances de l'ensemble de test.

1.2 Classification de textes

La classification de documents consiste à affecter, d'une manière automatique, un ensemble de documents à un ensemble prédéfini de classes. Bien que l'étude de cette tâche date depuis le début des années soixante, et qui portait sur l'indexation des revues scientifiques [Feldman and Sanger, 2006], [Maron, M. E., 1961], elle a connu un grand intérêt dans les dernières quinzaine années à cause de l'augmentation du volume des documents sous forme numérique, et la nécessité qui en découle pour les organiser [Sebastiani and Ricerche, 2002], [Feldman and Sanger, 2006]. Ce problème a été abordé suivant une approche basée sur les systèmes experts [Yang, 1994]. Cette approche, consistant à définir des classificateurs par des experts sur les domaines concernés, a atteint ses limites face au nombre croissant des documents sur le Web et par la diversité des catégories à traiter. La classification de textes s'apprête bien à l'apprentissage automatique, en l'occurrence un problème de classification supervisée. Elle est utilisée dans divers domaines d'application, tels que le filtrage des messages électroniques, l'indexation des articles, la recherche sur le Web, le peuplement automatique des catalogues, etc.

La classification de textes est définie formellement comme suit : [Sebastiani and Ricerche, 2002]

Définition 1.3. *La Classification de Document (CD) est une tâche consistant à assigner un booléen à chaque paire* $\langle d_j, c_i \rangle \in \mathcal{D} \times \mathcal{C}$*, tel que* $\mathcal{D}$ *est un espace de documents et* $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$ *est un ensemble prédéfini de classes. On définit alors une approximation de la fonction inconnue* $f : \mathcal{D} \times \mathcal{C} \rightarrow \{Vrai, Faux\}$ *(qui décrit comment les documents devraient être classés selon un expert) par une fonction* $g : \mathcal{D} \times \mathcal{C} \rightarrow \{Vrai, Faux\}$ *appelée classificateur. Si* $f(d_j, c_i) = Vrai$ *alors* d_j *est appelé membre ou exemple positif de la classe* c_i*, tandis que si* $f(d_j, c_i) = Faux$ *alors* d_j *est appelé exemple négatif de* c_i*.*

Bien que la fonction f est inconnue (i.e elle ne possède pas une forme analytique, d'ailleurs ce qui justifie le recours à l'apprentissage), elle est perçue à travers les instances de documents étiquetées par un expert du domaine. Le but alors de l'apprentissage est de trouver une fonction d'approximation g, qui se comporte d'une manière proche de f sur les données d'apprentissage et qui peut être généralisable pour des instances nouvelles de documents.

1.2.1 Systèmes de classification de textes

En général, et comme indiqué sur la figure 1.2, un système de classification de documents comporte les étapes suivantes :

1. **Pré-traitement** : dans cette première étape, le texte passe par certain nombre d'opérations de normalisation orthographiques, telles que la suppression des lettres de ponctuation, nombres, mots vides, caractères spéciaux et tout caractère non alphabétique. Certaines opérations liées à la langue peuvent être aussi effectuées à ce niveau, telles que la lemmatisation et l'extraction des racines.
2. **Représentation** : le texte tel qu'il est dans sa forme de base ne peut pas être manipulé par les algorithmes d'apprentissage. Pour cela, les documents seront transformés en une forme spécifique, souvent vectorielle, par l'extraction d'un certain ensemble de caractéristiques. La technique de sac à mots est la première à être utilisée. Elle consiste à considérer tout mot apparaissant au moins une fois dans un des documents comme une caractéristique ou dimension dans le vecteur de représentation. Vue le nombre important de caractéristiques qui peuvent être extraites, une réduction de dimension est souvent effectuée pour réduire la taille des vecteurs représentants. Cette réduction cause une perte d'information.
3. **Apprentissage** : le but de cette étape est de faire apprendre au système comment associer un document à une catégorie. Les algorithmes supervisés utilisés, tels que les machines à vecteurs de support ou l'algorithme des k plus proches voisins, reposent sur des *mesures de similarité* entre les documents pour décider si deux documents sont similaires ou pas (plus de détails sont donnés dans la section 1.2.5). Une fois le modèle de classification est déterminé, ses performances sont évaluées sur des données de test. Plusieurs mesures de performance existent dans la littérature, certaines s'intéressent plus aux erreurs de type *fausses acceptations* (précision), d'autres aux *faux rejets* (le rappel). Autres types de mesures reportent un compromis entre les deux types d'erreurs (exactitude ou F1). Selon le type d'application visée, l'utilisateur peut s'intéresser davantage à l'une ou l'autre des métriques.

Dans les sections suivantes on aborde, avec plus de détails, les trois étapes d'un système de classification de documents.

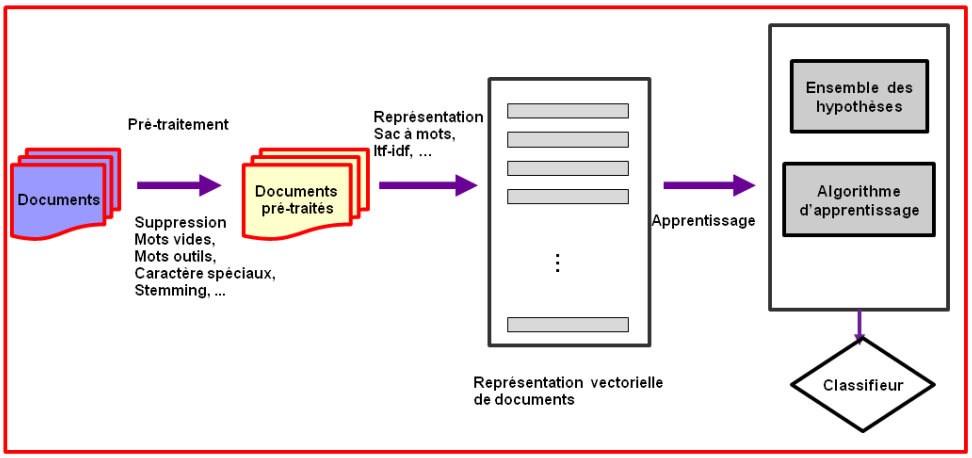


FIGURE 1.2: Système de classification de documents.

1.2.2 Pré-traitement

Avant d'appliquer toute méthode d'analyse de données textuelles, il est nécessaire de passer d'abord par une étape de pré-traitement. Cette étape vise à préparer les documents pour être compris et traités par la composante d'apprentissage. Elle inclut un certain nombre de tâches (obligatoires ou facultatives) telles que la conversion de formats, le nettoyage, la normalisation orthographique, la suppression des mots outils et la racinisation. On peut décomposer ces traitements en deux groupes selon qu'ils sont dépendants ou indépendants du langage.

1.2.2.1 Traitements indépendants du langage

Ces traitements peuvent être appliqués, si nécessaire, par tout système de classification d'une manière automatique. On compte parmi ces traitements les fonctions suivantes :

- *Conversion de format* : l'utilisation d'un encodage incorrect d'un fichier texte ne permet pas d'analyser correctement ce document. Pour cela il faut faire la conversion vers l'encodage adéquat dès le départ.
- *Suppression des caractères spéciaux* : jugés non importants pour la classification, ils sont supprimés et remplacés dans certains cas par des espaces. La liste des caractères spéciaux est la suivante : +, −, !, ?, ., ,, ;, :, {, }, =, #, &, %, $, [,], /, ", |, ainsi que les chiffres.
- *Découpage en unités lexicales* : ça permet de découper le texte en des lexèmes, des séquences de caractères non vides n'incluant pas des espaces

ou des caractères spéciaux. Cette opération s'effectue d'habitude suite à l'opération précédente.

— *Élagage ou Filtrage* : ceci permet de supprimer les unités lexicales peu ou trop fréquentes dans une catégorie. Ce type de termes est jugé non porteur d'information utile pour la caractérisation de la catégorie du document. Des métriques, telles que la fréquence dans le document ou dans la catégorie, sont utilisées pour définir quantitativement ce qui est un terme rare ou un terme fréquent. Ces paramètres sont déterminés empiriquement.

1.2.2.2 Traitements dépendants du langage

Contrairement aux traitements précédentes, ce type d'opérations est lié au langage des documents. Ces opérations peuvent être appliquées, si nécessaire, par tout système de classification pour une langue bien déterminée. On compte parmi ces traitements les opérations suivantes :

— *Conversion Majuscules/Miniscules* : pour les documents écrits avec l'alphabet latin, il est préférable de convertir les caractères majuscules en caractères minuscules. Ceci permet de réduire le nombre de termes différents dans le corpus.

— *Suppression des mots outils* : les mots tels que les articles, les conjonctions et les prépositions ne sont pas utiles pour la caractérisation de la catégorie d'un document. Ils sont supprimés en se basant sur des listes de mots outils spécifiques à la langue visée.

— *Traitement des synonymes* : les langages possèdent généralement des mots considérés comme synonyme l'un à l'autre. Ces synonymes sont déterminés et remplacés par un seul représentant.

— *Racinisation (anglais : Stemming)* : c'est une opération largement appliquée dans l'analyse des documents. Elle consiste à supprimer les préfixes et suffixes d'un mot pour le réduire à sa forme lexicale de base ou racine.

— *Lemmatisation (anglais : Lemmatization)* : elle consiste à réduire un mot en sa forme canonique ou son lemme. Ceci permet de regrouper les mots qui sont de la même famille.

1.2.3 Représentation des documents

La représentation des documents consiste à transformer ces documents vers un format compréhensible par la composante d'apprentissage. En effet, les algorithmes d'apprentissage conventionnels ont été conçus pour traiter des données

bien structurées, telles que les vecteurs ou tables des données. Au contraire à ce type de données structurées, les documents sont plats et présentent une hétérogénéité dans leur contenu et leur taille. Le *modèle vectoriel* est de loin la plus ancienne représentation utilisée. Dans ce modèle, un document est modélisé par un vecteur dans un espace de représentation, chaque dimension représente une caractéristique du document. Les documents peuvent être caractérisés par les mots ou les N-grammes.

1.2.3.1 Sac-à-mots

Le modèle de *sac-à-mots* utilise tous les mots présents dans la collection des documents pour indexer les vecteurs de représentation [Feldman and Sanger, 2006, Han et al., 2011].

Soit D l'ensemble des documents du corpus et T l'ensemble incluant tous les termes apparaissant dans le corpus. Dans le modèle du sac-à-mots, un document est représenté alors par un vecteur V dans l'espace vectoriel $\mathbb{R}^{|T|}$.

La manière de définir une valeur (ou un poids) pour une entrée du vecteur V définit ce qu'on appelle le *schéma de pondération*. Il existe plusieurs schémas de pondération. On définit la *fréquence d'un terme* dans un document comme le nombre de fois où le terme apparait dans ce document. Cette quantité, notée $freq(d_i, t_j)$ mesure l'association d'un terme $t_j \in T$ à un document donné $d_i \in D$, elle vaut zéro si le terme n'apparait pas dans ce document. On définit alors la *matrice des fréquences* comme une table regroupant tous les vecteurs représentant les documents du corpus. Si la taille du corpus vaut n et le nombre de termes vaut m on aura une matrice comme suit :

$$\mathrm{M} = \begin{bmatrix} freq(d_1, t_1) & \dots & freq(d_n, t_1) \\ \vdots & \ddots & \vdots \\ freq(d_1, t_m) & \dots & freq(d_n, t_m) \end{bmatrix}$$

Selon le schéma de pondération adopté, il existe plusieurs alternatives à la quantité $freq(d_i, t_j)$:

— **Fréquence booléenne** (anglais : Boolean frequency) : c'est une forme simplifiée dans laquelle $freq(d_i, t_j) = 1$ si le terme t_j apparait au moins une fois dans le document d_i, et $freq(d_i, t_j) = 0$ sinon.

— **Fréquence relative du terme** (anglais : Relative term frequency) : cette quantité mesure la fréquence du terme dans un document par rapport au

nombre total de termes présents dans ce même document :

$$freq_r(d_i, t_j) = \frac{\#t_j}{|d_i|}$$

où $\#t_j$ est le nombre d'occurrences du terme t_j dans le document d_i, et $|d_i|$ est la taille du document d_i.

— **Fréquence inversée** (anglais : Term Frequency - Inverse Document Frequency (TF-IDF)) : si la quantité précedente concerne la fréquence d'un terme au sein d'un seul document, la fréquence inversée (TF-IDF) concerne la fréquence du terme au sein du corpus entier :

$$\text{TF-IDF}(d_i, t_j) = freq(d_i, t_j) \times \log(\frac{N}{DocFreq(t_j)})$$

où N est le nombre de document dans le corpus, $DocFreq(t_j)$ est le nombre de documents contenant au moins une fois le terme t_j. On peut bien remarquer que plus le terme est présent dans plusieurs documents (i.e, $DocFreq(t_i)$ est proche de N), alors le log est moins important. La figure 1.3 illustre l'utilisation du modèle de sac-à-mots pour représenter un document.

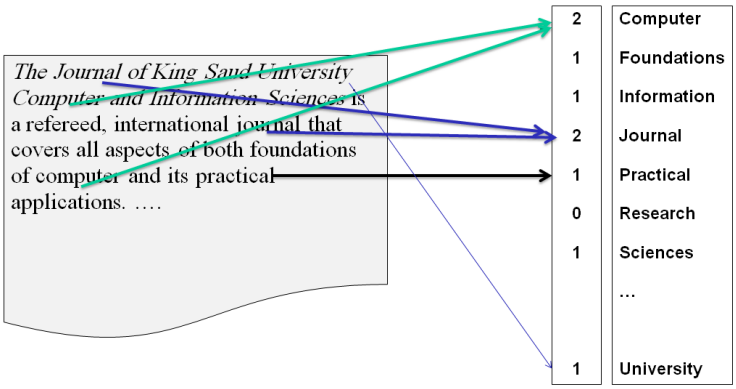


FIGURE 1.3: Structuration d'un texte avec le modèle sac-à-mots.

1.2.3.2 N-grammes

Les N-grammes sont largement utilisées dans l'analyse des données textuelles et les tâches de traitement du langage naturel [Cavnar and Trenkle, 1994, Huff-

man, 1995]. Pour une chaine de caractères, les N-grammes sont des tranches de N caractères contiguës de cette chaine. Bien que dans la littérature, le terme N-gramme peut inclure la notion de tout sous ensemble de caractères d'une chaîne (par exemple, un N-gramme constitué du premier et troisième caractère d'un mot), nous utilisons ici le terme pour désigner seulement les tranches contiguës de caractères.

Afin d'illustrer la notion des N-grammes, on construira les 2-grammes, 3-grammes et et 4-grammes du mot (TEXTE) qui est précédé et suivi d'une espace (l'espace est représentée par le caractère '_') :

$$\begin{aligned}
&\text{2-grammes} : _T, TE, EX, XT, TE, E_ \\
&\text{3-grammes} : _TE, TEX, EXT, XTE, TE_ \\
&\text{4-grammes} : _TEX, TEXT, EXTE, XTE_
\end{aligned}$$

Les N-grammes présents dans un corpus peuvent être utilisés pour indexer le vecteur représentant un document. Une entrée dans ce vecteur donne le nombre de N-grammes, correspondants à cette entrée, présents dans ce document.

Les N-grammes ont été utilisés pour représenter des textes dans plusieurs domaines : recherche d'information, indexation, analyse des séquences et classification de documents [Cavnar and Trenkle, 1994, Huffman, 1995]. Deux niveaux de granularité du gramme sont possibles : les caractères ou les mots. En effet, dans l'exemple précédent les N-grammes ainsi construits concernent le niveau caractère. L'exemple suivant illustre la notion des N-grammes au niveau mot. Les entités considérées sont les mots constituant la phrase "rien ne se perd tout se crée" :

1-grammes : *rien*, *ne*, *se*, *perd*, *tout*, crée

2-grammes : *rien ne*, *ne se*, *se perd*, *perd tout*, *tout se*, *se* crée

3-grammes : *rien ne se*, *ne se perd*, *se perd tout*, *perd tout se*, *tout se* crée

4-grammes : *rien ne se perd*, *ne se perd tout*, *se perd tout se*, *perd tout se* crée

1.2.4 Réduction de dimension

La nature du problème de classification de documents fait que, dans la plus part des cas, la dimension de la matrice des fréquences est très importante. En effet, si l'on considère tous les 3-grammes possibles, il s'agira d'un vecteur V de dimension $(|A| \times |A| \times |A|)$, telle que $|A|$ est la taille de l'alphabet utilisé. Si,

pour le niveau caractère, l'alphabet est limité[1], il n'en est pas de même pour le niveau mot, pour lequel l'alphabet peut atteindre des milliers de mots différents. Ceci donne lieu à des vecteurs creux, du fait que la plus part de ces 3-grammes ne figurent pas dans les documents. Malgré que certains algorithmes utilisés peuvent manipuler ce type de données, une réduction de dimension s'avère d'un grand intérêt surtout pour améliorer la complexité temporelle et spatiale.

Principalement, la réduction de dimension se fait de deux manières : la sélection des caractéristiques ou l'extraction des caractéristiques [Feldman and Sanger, 2006].

Sélection des caractéristiques

Plusieurs mots ne sont pas importants pour la classification de documents et peuvent être éliminés sans altérer les performances du système. Ceci peut se faire dans la partie pré-traitement, en supprimant par exemple les mots outils ou en filtrant les mots très fréquents. Des techniques relevant de la théorie de l'information ont été aussi utilisées pour mesurer l'utilité d'un terme pour la classification, telles que le gain d'information, l'information mutuelle ou la mesure Chi-2 [Manning et al., 2008].

Extraction des caractéristiques

L'extraction des caractéristiques consiste à créer un ensemble plus petit de nouvelles caractéristiques synthétisées à partir de l'ensemble d'origine. Des techniques de regroupement sémantique des termes ont été utilisées pour réduire la taille de l'ensemble des caractéristiques. Dans ce type de techniques, les mots qui présentent un fort degré de similarité sémantique sont regroupés et remplacés par un seul concept. D'autres techniques d'extraction de composantes sont aussi utilisées. On peut citer, à titre d'exemples, l'analyse en composantes principales ou l'indexation sémantique latente.

1.2.5 Techniques et algorithmes de classification de documents

Dans la partie pré-traitement, il a été question de préparer le corpus pour la tâche de classification de documents. Le résultat étant une représentation des documents au format (souvent vectoriel) compréhensible par les algorithmes conven-

1. Par exemple, $|A| = 26$ pour le français, ce qui donne un vecteur V de taille égale à 17576.

tionnels de classification. On rappelle aussi que le corpus est partitionné en deux parties. La première partie des documents est utilisée pour l'apprentissage du modèle de classification, et la deuxième pour la validation et l'évaluation de ce modèle.

A fin de produire un modèle de classification, la plupart des algorithmes se basent sur des métriques pour mesurer la distance ou la similarité entre deux documents. Nous commencerons par faire un survol de ces mesures avant d'aborder les catégories des algorithmes de classification de documents.

1.2.5.1 Mesures de distance

Mesurer la similarité entre des documents est une opération très importante dans la tâche de classification. Une mesure bien adaptée permet de capturer les formes des classes en question.

Les deux notions de distance et similarité sont duales. Plus deux objets sont similaires, mois ils sont distants, et vice-versa. Avant de voir des exemples des mesures de distance, utilisées dans le contexte de classification de documents, on rappelle la définition formelle de la notion de distance ainsi que ses caractéristiques :

Définition 1.4. *On appelle distance sur un ensemble E une application $d : E \times E \longrightarrow \mathbb{R}^+$, vérifiant les propriétés suivantes :*

Symétrie :	$\forall (u,v) \in E \times E,$	$d(u,v) = d(v,u)$
Séparation :	$\forall (u,v) \in E \times E,$	$d(u,v) = 0 \Leftrightarrow u = v$
inégalité triangulaire :	$\forall (u,v,t) \in E \times E \times E,$	$d(u,t) \leq d(u,v) + d(v,t)$

Pour un ensemble de documents, une mesure de similarité entre deux documents représentés par leurs vecteurs $v, u \in \mathbb{R}^m$ est définie comme une application $d : \mathbb{R}^m \times \mathbb{R}^m$ à valeurs dans $\mathbb{R}^+$. Par convention, les mesures de similarité renvoient des valeurs dans l'un des deux intervalles [-1,1] ou [0,1], où une similarité égale à 1 indique un maximum de ressemblance.

Soit $v = (v_1, \ldots, v_m)$ et $u = (u_1, \ldots, u_m)$ deux vecteurs de dimension m. Dans ce qui suit on donne des exemples de mesures de distance utilisées dans le contexte de la classification de documents :

— **Distance euclidienne** : En plus d'être simple à utiliser pour un modèle vectoriel, c'est la première distance utilisée dans la CD. La distance

euclidienne entre v, u se calcule comme suit :

$$d_{Euc} = \sqrt{(v_1 - u_1)^2 + \ldots + (v_n - u_n)^2}$$

— **Distance de cosinus** : Les vecteurs représentés dans un espace à partir de son origine, forment entre eux un angle. Le cosinus de cet angle donne une valeur proche de 1 pour un angle petit, et une valeur proche de zéro quant cet angle devient plus grand. Pour cela, la distance se calcule en soustrayant ce cosinus de 1 pour avoir des valeurs proches de 0 pour des vecteurs ayant des directions différentes, et inversement des valeurs proches de 1 pour des vecteurs ayant des même directions :

$$d_{Cos} = 1 - \frac{(v_1 u_1 + \ldots + v_n u_n)}{\sqrt{(v_1^2 + \ldots + v_n^2)} \times \sqrt{(u_1^2 + \ldots + u_n^2)}}$$

— **Distance de Manhattan** : La distance entre les deux vecteurs v et u vaut la somme des valeurs absolues des différences entre leurs coordonnées :

$$d_{Man} = |v_1 - u_1| + \ldots + |v_n - u_n|$$

— **Distance de Dice** :

$$d_{Dice} = \frac{2 \times \sum_1^m v_i u_i}{\sum_1^m (v_i)^2 + \sum_1^m (u_i)^2}$$

— **Distance de Jaccard** :

$$d_{Jacc} = 1 - \frac{\sum_1^m v_i u_i}{\sum_1^m (v_i)^2 + \sum_1^m (u_i)^2 - \sum_1^m v_i u_i}$$

1.2.5.2 Algorithmes de classification de documents

Les phases de pré-traitement et représentation des documents permettent de transformer le texte en un format compréhensible et utilisable par les algorithmes de classification. Dans ce qui suit on brosse les principales classes d'algorithmes d'apprentissage automatique, qui ont été appliquées au problème de la classification de documents.

Modèle probabiliste : Dans ce type d'algorithme, une probabilité $P(c_j \mid d)$ indiquant qu'un document $d = (t_1, t_2, \ldots)$ appartient à la classe c_j est calculée en appliquant le théorème de Bayes suivant :

$$P(c_j \mid d) = \frac{P(d \mid c_j) \times P(c_j)}{P(d)}$$

La quantité $P(d)$ étant constante pour toutes les catégories, elle est calculée une seule fois. Une estimation de la probabilité $P(c_j)$ peut être aussi calculée par la fraction de documents d'apprentissage assignés à la classe c_j. Par contre, pour calculer $P(d \mid c_j)$, le modèle suppose une indépendance entre les termes de d, d'où l'appellation de l'algorithme bayésien naïve. Cette hypothèse permet d'écrire :

$$P(d \mid c_j) = \prod_i P(t_i \mid c_j)$$

Lors de la classification d'un document d, on calcule toutes les probabilités $P(d \mid c_j)$, telle que $j \in \{1, \ldots, |C|\}$. La classe choisie est celle présentant la plus grande probabilité.

Malgré que l'hypothèse d'indépendance entre caractéristiques est souvent fausse, le classificateur naïve bayésien a fait preuve de son efficacité. Il est souvent utilisé pour comparer de nouvelles approches [Manning et al., 2008].

Modèle basé sur les arbres de décision Les arbres de décision [Quinlan, 1986] sont des graphes acycliques orientés, dans lesquels les nœuds internes contiennent des prédicats ou conditions sur les valeurs d'un attribut, et les feuilles représentent les catégories. Les arbres de décision permettent une décomposition hiérarchique de l'espace d'apprentissage étiqueté, à travers les tests effectués dans les nœuds.

Dans le contexte de la classification de documents, les nœuds sont typiquement des conditions sur la présence ou l'absence d'un mot dans le document [Aggarwal and Zhai, 2012]. La décomposition de l'ensemble d'apprentissage se fait d'une manière récursive dans l'arbre. Le but c'est d'avoir des feuilles contenant un nombre minimal de documents vérifiant certaines conditions de pureté de classe. Le principe de la classe majoritaire au niveau d'une feuille est utilisé pour la classification. A fin d'être classé, un nouveau document passe du sommet source vers une feuille de l'arbre.

Modèle basé sur les réseaux de neurones Un réseau de neurones [Mehrotra et al., 1997] (en anglais : Artificial Neural Network (ANN)) est un modèle de calcul dont la conception est très schématiquement inspirée du fonctionnement de vrai neurone. Il est composé de nœuds ou d'unités connectés par des liens. Un lien d'un nœud i vers un nœud j sert à propager une activation a_i de i à j. Chaque lien possède aussi un poids numérique w_{ij} qui détermine la force de ce lien. Chaque unité j calcule d'abord une somme pondérée de ses entrées :

$$in_j = \sum_{i=0}^{n} w_{ij} a_i$$

Elle applique en suite un fonction d'activation g à cette somme pour calculer la sortie :

$$a_j = g(in_j) = g(\sum_{i=0}^{n} w_{ij} a_i)$$

Un réseau de neurones est en général composé d'une succession de couches dont chacune prend ses entrées sur les sorties de la précédente. Chaque couche (l_i) est composée de N_i unités, prenant leurs entrées sur les N_{i-1} unités de la couche précédente. La couche d'entrée possède plusieurs unités et la couche de sortie peut avoir une ou plus unités.

Les réseaux de neurones ont été utilisés pour la classification de documents. Les unités d'entrées reçoivent les poids des termes du document et les unités de sortie produisent la valeur de la catégorie [Sebastiani and Ricerche, 2002]. L'apprentissage se fait par rétro-propagation, dans laquelle les documents sont présentés un par un à la couche d'entrée, et l'activation des nœuds se propage d'une couche à l'autre jusqu'à la couche de sortie. Si une mauvaise classification se produit, l'erreur se propage en arrière à travers les couches internes tout en modifiant les poids des liens pour minimiser l'erreur.

Modèles basés sur les exemples Ces modèles ne construisent pas une représentation explicite des catégories. Ils se basent uniquement sur les étiquettes des classes attachées aux documents d'apprentissage similaires au document à classer [Sebastiani and Ricerche, 2002, Aggarwal and Zhai, 2012]. En effet, dans ce type de systèmes, la phase d'apprentissage est minimale, juste quelques opérations de pré-traitement sont réalisées. Lorsqu'un exemple de test se présente, le classificateur décide de sa classe en se basant sur des calculs de similarité par rapport aux exemples d'apprentissage.

L'exemple typique de ces modèles est l'algorithme des K-Plus-Proches Voisins (k-PPV). Cet algorithme est basé sur l'apprentissage par analogie, en comparant l'instance à classer aux K instances d'apprentissage les plus proches. Le sens de proximité est matérialisé à travers l'une des distances ou mesures de similarité (distance euclidienne par exemple). La classe choisie pour le test est celle la plus fréquente parmi les classes des K instances. Le paramètre K peut être choisi empiriquement.

Machines à vecteurs de support Les machines à vecteurs de support (Support Vector Machines (SVMs)) ont été introduites pour résoudre des problèmes de classification [Cortes and Vapnik, 1995]. Conceptuellement, elles implémentent l'idée suivante : les vecteurs d'entrée sont reproduits, d'une manière non linéaire, dans un nouvel espace de très grande dimension, appelé *espace de reproduction*. Dans ce nouvel espace, une surface linéaire de décision est construite. Les propriétés de cette surface de décision assurent une bonne capacité de généralisation du classificateur.

Géométriquement, un classificateur SVM binaire peut être vu comme un hyperplan dans l'espace de reproduction, séparant les exemples positifs et les exemples négatifs de la classe. Cet hyperplan est choisi dans la phase d'apprentissage de telle sorte qu'il soit l'hyperplan séparant les deux groupes d'exemples avec une marge maximale. La marge représente la distance entre l'hyperplan et les exemples les plus proches des deux groupes [Feldman and Sanger, 2006].

Les SVMs ont été appliquées à la classification de documents initialement par Joachims [Joachims, 1998], puis par [Drucker et al., 1999],[Dumais and Chen, 2000], [Klinkenberg and Joachims, 2000], [Taira and Haruno, 1999] et par [Yang and Liu, 1999]. Une présentation détaillée des machines à vecteurs de support fera l'objet de la section 2.1.3 du Chapitre 2.

Modèles basés sur les ensembles de classificateurs L'idée d'utiliser plusieurs classificateurs est inspirée du fait qu'un ensemble d'experts peut produire de meilleures décisions qu'un seul expert, en combinant leurs connaissances. Dans ce type de modèles, un ensemble de classificateurs est construit. Un exemple de test est assigné une classe par vote entre les décisions individuelles de chaque classificateur [Dietterich, 2000]. Cet ensemble peut être composé de classificateurs de même type ou des classificateurs de nature différente. Le but de cette intégration est d'atteindre une meilleure performance du système. La construction de telles systèmes est conditionnée par les paramètres suivants : diversité ou pas du type des classificateurs utilisés, choix de l'ensemble d'apprentissage aléatoire ou guidé, et la manière de prendre la décision de classification.

1.2.6 Évaluation des performances

Le but général du développement d'un système de classification est de réduire l'erreur mesurée sur un ensemble de test ; c'est-à-dire de produire un classificateur qui présente une bonne capacité de généralisation. On entend par erreur de

classification la proportion de décisions incorrectes commises par le classificateur. Cette quantité peut être aussi réduite en maximisant l'exactitude (anglais : accuracy), sachant que l'exactitude est la proportion de décisions correctes prises par le classificateur.

L'exactitude n'est pas appropriée pour des classes à faible pourcentage de documents dans le corpus, par exemple moins de 10%. En effet, si l'on considère un corpus à plusieurs classes ayant tous des pourcentages de documents inférieurs à 10%, un classificateur ayant comme stratégie de répondre par non pour toute question donnera une exactitude de 90% pour ce corpus. Pour cela, on donnera dans cette section d'autres mesures à considérer dans diverses configurations, à savoir : la précision, le rappel et F_1.

On utilisera le terme générique *efficacité* pour les mesures qui permettent d'évaluer la qualité du classificateur, tels que l'exactitude, le rappel, la précision ou F_1. Le terme *performance* se réfère à la qualité avec laquelle le système de classification fait l'apprentissage. Il peut s'agir, par exemple, de la complexité temporelle ou espace du système.

1.2.6.1 Mesures d'efficacité

Les mesures d'efficacité utilisées, héritées du domaine de recherche d'information, incluent l'exactitude, la précision, le rappel et F_1. L'exactitude correspond au pourcentage des décisions correctes prises par le classificateur. Le rappel, défini pour une classe déterminée, correspond au pourcentage des documents bien classés parmi les documents appartenant à cette classe. La précision, définie aussi pour une classe déterminée, correspond au pourcentage des documents correctement assignés par le classificateur à cette classe parmi tous les documents assignés à cette classe par ce même classificateur. La mesure F_1 constitue un compromis entre le rappel et la précision.

Pour une classe donnée, on donne la table de contingence (Tableau 1.1). Par exemple, un vrai positif (VP) correspond à un document correctement classifié dans la catégorie. Par contre, un faux positif correspond à un document assigné par le classificateur à la catégorie alors qu'il ne fait pas partie de cette catégorie.

Les mesures sont définies par les formules reportées sur le tableau 1.2. On note que F_1 est une moyenne harmonique de la précision et le rappel, donnée par :

$$F_1 = \frac{2 \times \text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}}$$

		Étiquette affectée par l'expert	
		Oui	Non
Décision du classificateur	Oui	VP	FP
	Non	FN	VN

Tableau 1.1: Table de contingence.

Exactitude	Précision	Rappel	F_1
$\frac{VP+VN}{VP+VN+FP+FN}$	$\frac{VP}{VP+FP}$	$\frac{VP}{VP+FN}$	$\frac{2\times VP}{2\times VP+FP+FN}$

Tableau 1.2: Mesures d'efficacité d'un système de classification pour une classe.

Pour un corpus comportant plusieurs classes, on désire avoir, pour chaque mesure, une moyenne pour l'ensemble des classes. Il existe deux manières pour calculer ces moyennes : macro-moyenne ou micro-moyenne. La macro-moyenne est une simple moyenne arithmétique des mesures de chaque classe est calculée. Par contre, la micro-moyenne considère dans le calcul tous les documents du corpus sans tenir compte de la classe (Voir tableau 1.3).

La différence entre les deux méthodes est importante. Dans la macro-moyenne, les différentes classes ont des poids égaux, ce qui donne la même importance à une petite classe et une grande classe. Par contre, dans le micro-moyenne tout document est assigné le même poids que le reste des documents. Pour la mesure F_1, qui ne tient pas compte des VN, cela a comme effet la domination des grandes classes.

Le recours à l'une de ces mesures ou l'autre est guidé par l'application visée. Dans certaines applications, telle que la détection d'intrusion, il est plus dangereux d'accepter des intrus (FP) que d'avoir un taux de refus incorrects élevé (FN). Pour ce type d'application on s'intéresse à la précision, par laquelle on veut garantir un bon nombre de VP. Dans d'autres types d'applications, telle que le filtrage des Spams, on est sensible plus au mauvais refus (marquer un mail légitime comme spam), on s'intéresse alors au rappel par lequel on veut assurer beaucoup de VP.

Dans d'autre types d'application, on cherche à avoir un compromis entre rap-

	Micro-moyenne	Macro-moyenne
Exactitude	$\frac{2\times\sum_1^{\lvert\mathcal{C}\rvert}\mathrm{VP}_i}{2\times\sum_1^{\lvert\mathcal{C}\rvert}VP_i+\sum_1^{\lvert\mathcal{C}\rvert}FP_i+FN_i}$	$\frac{\sum_1^{\lvert\mathcal{C}\rvert}\frac{2\times VP_i}{2\times VP_i+FP_i+FN_i}}{\lvert\mathcal{C}\rvert}$
Précision	$\frac{\sum_1^{\lvert\mathcal{C}\rvert}VP_i}{\sum_1^{\lvert\mathcal{C}\rvert}VP_i+FP_i}$	$\frac{\sum_1^{\lvert\mathcal{C}\rvert}\frac{VP_i}{VP_i+FP_i}}{\lvert\mathcal{C}\rvert}$
Rappel	$\frac{\sum_1^{\lvert\mathcal{C}\rvert}VP_i}{\sum_1^{\lvert\mathcal{C}\rvert}VP_i+FN_i}$	$\frac{\sum_1^{\lvert\mathcal{C}\rvert}\frac{VP_i}{VP_i+FN_i}}{\lvert\mathcal{C}\rvert}$
F_1	$\frac{2\times\sum_1^{\lvert\mathcal{C}\rvert}\mathrm{VP}_i}{2\times\sum_1^{\lvert\mathcal{C}\rvert}VP_i+\sum_1^{\lvert\mathcal{C}\rvert}FP_i+FN_i}$	$\frac{\sum_1^{\lvert\mathcal{C}\rvert}\frac{2\times VP_i}{2\times VP_i+FP_i+FN_i}}{\lvert\mathcal{C}\rvert}$

Tableau 1.3: Mesures d'efficacité d'un système de classification pour un ensemble de classes.

pel et précision. La mesure F_1 décourage un système qui sacrifie une mesure par rapport à l'autre [Han et al., 2011].

1.2.6.2 Techniques de validation

A fin d'évaluer et comparer différentes techniques de classification de documents, on estime l'efficacité d'un système en utilisant les mesures d'efficacité données dans la section précédente. Cette efficacité est estimée sur un ensemble de test indépendant de l'ensemble d'apprentissage. Plusieurs techniques de choix de l'ensemble de test existent dans la littérature. On s'intéresse plus particulièrement à deux techniques,a savoir, l' échantillonnage aléatoire et la validation croisée :

— *Échantillonnage aléatoire* : Dans cette méthode, l'ensemble des documents est subdivisé aléatoirement en deux sous ensembles indépendants. Le choix de la proportion des documents dans les deux sous ensembles n'est pas unique. On peut choisir par exemple les valeurs suivantes : (2/3, 1/3), (3/4, 1/4) ou (4/5, 1/5). Le premier ensemble est alors utilisé pour construire le classificateur et le deuxième ensemble pour l'évaluer. Cette procédure est répétée plusieurs fois (en gardant la même proportion des documents à chaque fois) et l'efficacité est donnée en calculant la moyenne des efficacités de toutes les itérations.

— *Validation croisée* : Dans cette méthode, l'ensemble des documents $\mathcal{D}$ est subdivisé en k sous ensembles indépendants $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$, de tailles approximativement égales. L'apprentissage et l'évaluation se font k fois en

sélectionnant à chaque fois un sous ensemble différent pour jouer le rôle d'ensemble de test et les $k-1$ sous ensembles restants comme ensemble d'apprentissage. Chaque sous ensemble participe une seule fois dans l'évaluation. Pour cela, l'efficacité est évaluée en calculant la somme des bonnes et mauvaises décisions à chaque itération.
Une stratification de la validation croisée est possible en sélectionnant à chaque fois un sous ensemble reflétant approximativement la distribution des différentes classes en termes de taille de chaque classe et proportion des exemples positifs et négatifs au sein des classes. En général, une validation croisée stratifiée avec $k = 5$ ou $k = 10$ est utilisée dans les systèmes de classification de documents.

1.3 Conclusion

Dans ce chapitre, nous avons commencé par exposer la notion d'apprentissage en général et expliquer dans quels contextes il est utile de l'appliquer. On a montré que les situations qui s'apprêtent à l'apprentissage sont des problèmes pour lesquels on ne peut pas trouver directement un algorithme de résolution ou une forme analytique. On dispose uniquement de données collectées sur lesquelles on veut inférer une solution. La classification est l'exemple typique de l'apprentissage supervisé. Le but était d'apprendre un classificateur à partir des données étiquetées disponibles. On s'est intéressé, par la suite, plus particulièrement au cas des données textuelles, pour lequel on a expliqué les différentes phases et traitements dans un système de classifications des documents. Un aperçu plus détaillé a été donné sur les méthodes d'évaluation de ce type de systèmes et des mesures d'efficacité qui leurs sont spécifiques.

Chapitre 2

Méthodes à noyaux et noyaux rationnels

Dans ce chapitre, nous exposons deux domaines importants en grand lien avec l'apprentissage, à savoir les méthodes à noyaux et les noyaux rationnels. Nous commençons, dans la section 2.1, par donner un aperçu général des méthodes à noyaux, puis nous expliquons la notion de noyau et son lien avec la mesure de similarité ou distance. Nous illustrons ensuite ces méthodes à noyaux par une des techniques les plus utilisées : les Séparateurs à Vastes Marges (SVM). Dans la section 2.2, nous abordons les *noyaux rationnels*, basés sur les transducteurs de mots. Ce type de noyaux est dit "unificateur", i.e. que tout autre noyau sur les mots n'est qu'un type particulier des noyaux rationnels. En fin, nous exposons les propriétés principales ainsi que des exemples des noyaux rationnels.

2.1 Méthodes à noyaux

Les méthodes à noyaux sont des techniques largement utilisées dans l'apprentissage automatique. Leur flexibilité permet de les utiliser pour étendre des algorithmes conventionnels, tels que les SVMs ou l'algorithme des k-PPVs, pour définir des surfaces non linéaires de décision [Shawe-Taylor and Cristianini, 2004], [Gartner, 2009]. L'idée principale de ces méthodes réside dans l'utilisation des *noyaux*, appelés aussi *fonctions noyaux*, qui définissent un produit interne dans un espace de grande dimension. Cet espace est appelé *espace de reproduction*. Pour un algorithme n'utilisant que des produits scalaires entre des vecteurs dans l'espace d'entrée, le remplacement de ces produits par des *noyaux définis positifs* permet d'étendre cet algorithme pour effectuer une séparation linéaire dans l'espace de reproduction. D'une manière équivalente, ceci permet aussi d'effectuer une séparation non linéaire dans l'espace d'entrée. Dans les sections suivantes,

nous expliquons la notion de noyau ou fonction noyau ainsi que les principes des méthodes à noyaux.

Soit $\mathcal{X}$ un ensemble d'éléments. Un objet est un élément $x \in \mathcal{X}$. Un ensemble de données à traiter est souvent un ensemble fini d'objets : $\mathcal{S} = \{x_1, x_2, \ldots, x_n\}$. Par exemple :

— $\mathcal{X}$ est l'ensemble de tous les documents, de taille finie, composés des mots de la langue arabe.

— Un objet $x \in \mathcal{X}$ est un document en langue arabe de taille finie.

— $\mathcal{S}$ est un corpus composé de n documents.

Soit un algorithme $A : \mathcal{F}^n \longrightarrow \mathbb{R}$ capable d'opérer sur des données d'un espace $\mathcal{F}$, c-à-d il reçoit en entrée n éléments de $\mathcal{F}$ et renvoi une sortie réelle.

Pour pouvoir analyser le corpus $\mathcal{S}$, il faut soit avoir l'ensemble $\mathcal{X}$ identique à $\mathcal{F}$, soit définir une application $\phi : \mathcal{X} \longrightarrow \mathcal{F}$ et opérer sur l'ensemble :

$$\phi(\mathcal{S}) = \{\phi(x_1), \phi(x_2), \ldots, \phi(x_n)\} \in \mathcal{F}^n$$

Plutôt que de représenter chaque objet $x \in \mathcal{X}$ explicitement par $\phi(x) \in \mathcal{F}$, et donc $\mathcal{S} \in \mathcal{X}^N$ par $\phi(\mathcal{S}) \in \mathcal{F}^N$, on définit une fonction de similarité :

$$K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$$

On peut alors représenter $\mathcal{S}$ par la matrice de similarité, de taille $n \times n$, définie par :

$$[K]_{ij} := K(x_i, x_j)$$

On utilisera alors des algorithmes capables de traiter des matrices carrées. En fait, ces algorithmes utilisent des produits internes entre les éléments de $\phi(\mathcal{S})$. On montrera dans la suite que ces produits internes peuvent être évalués par la fonction K :

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

La représentation par similarité permet d'avoir une matrice réelle carrée quelque soit le type des objets à représenter (vecteurs, documents textuelles, images, graphes, etc...). Le même algorithme alors peut traiter tout type de données. Ceci permet d'avoir une grande souplesse dans le choix de la fonction de similarité, d'une part, et dans le choix de l'algorithme qui sera appliqué à la matrice de similarité, d'autre part. La taille de la matrice est toujours $n \times n$ quelles que soient la nature et la complexité des objets.

2.1.1 Notion de noyau

Définition 2.1. *Toute fonction de similarité $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ est appelée un noyau sur $\mathcal{X}$.*

L'idée derrière est de définir une fonction K telle que pour toute paire d'objets $(x, x') \in \mathcal{X}^2$, $K(x, x')$ soit égale à un produit interne des vecteurs $\phi(x)$ et $\phi(x')$:

$$K(x, x') = \quad \langle \phi(x), \phi(x') \rangle$$

Nous nous restreindrons à une classe particulière de noyaux, dite noyaux définis positifs.

Définition 2.2. *Un noyau défini positif (n.d.p.) sur l'ensemble $\mathcal{X}$ est une fonction $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ symétrique : $\forall (x, x') \in \mathcal{X}^2, K(x, x') = K(x', x)$ et qui satisfait, pour tout $N \in \mathbb{N}, (x_1, x_2, \ldots, x_N) \in \mathcal{X}^N$ et $(a_1, a_2, \ldots, a_N) \in \mathbb{R}^N$:*

$$\sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j K(x_i, x_j) \geq 0$$

De manière équivalente, pour tout ensemble d'objets $\mathcal{S} \in \mathcal{X}^N$, la matrice de similarité $[K]_{ij} := K(x_i, x_j)$ est *symétrique semi-définie positive*. Généralement, on se réfère à un noyau défini positif par le terme "noyau" tout simplement [Scholkopf and Smola, 2001]. Les méthodes à noyaux opèrent sur des matrices symétriques semi-définies positives.

Exemples de noyaux :

- ▷ **Noyau linéaire**
  Soit $\mathcal{X} = \mathbb{R}^N$ et soit la fonction $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ définie par :
  $\forall (x, x') \in \mathcal{X}^2, \quad K(x, x') = \langle x, x' \rangle$
  On peut vérifier que K est un noyau défini positif :
  — $\langle x, x' \rangle = \langle x', x \rangle$
  — $\sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j \langle x_i, x_j \rangle \geq 0$
- ▷ **Généralisation du noyau linéaire**
  Soit ϕ une fonction quelconque, alors

  $$K(x, x') = \langle \phi(x), \phi(x') \rangle$$

  est un noyau :

— $\forall (x, x') \in \mathcal{X}^2 : \quad K(x, x') = \langle \phi(x), \phi(x') \rangle = \langle \phi(x'), \phi(x) \rangle = K(x', x)$
— $\forall (a_1, \dots a_n) \in \mathcal{R}^N, (x_1, \dots, x_N) \in \mathcal{X}^N :$
$$\sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j K(x_i, x_j) = \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j \langle \phi(x_i), \phi(x_j) \rangle \geq 0$$

▷ **Noyau polynomial**
Soit $\mathcal{X} = \mathbb{R}^N$. La fonction $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ définie par : $\forall x, x' \in \mathcal{X}, K(x, x') = \langle x, x' \rangle^d$ est un noyau.
Pour $N = 2$ et $d = 2$,

$$\begin{aligned} K(x, x') &= (x_1 x'_1 + x_2 x'_2)^2 \\ &= (x_1^2 x'^2_1 + x_2^2 x'^2_2 + 2 x_1 x'_1 x_2 x'_2) \\ &= \langle (x_1^2, x_2^2, \sqrt{2} x_1 x_2), (x'^2_1, x'^2_2, \sqrt{2} x'_1 x'_2) \rangle \\ &= \langle \phi(x), \phi(x') \rangle \end{aligned}$$

(Pour $x = (x_1, x_2)$, $\phi((x_1, x_2)) = (x_1^2, x_2^2, \sqrt{2} x_1 x_2) \in R^3$)

Réciproquement, si $K(x, x')$ est un noyau sur $\mathcal{X}$, alors il existe un espace de Hilbert $\mathcal{H}$ muni d'un produit scalaire $\langle ., . \rangle_{\mathcal{H}}$ et une application $\phi : \mathcal{X} \longrightarrow \mathcal{H}$ tel que :

$$\forall (x, x') \in \mathcal{X}^2, K(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$$

Ceci veut dire que tout noyau peut être vu comme un produit scalaire dans un espace de Hilbert.

2.1.2 Principes des méthodes à noyaux

Généralement, une méthode à noyaux comprend deux composantes : un module qui fait la transformation vers un espace de reproduction, et un algorithme d'apprentissage pour découvrir les formes [1] linéaires de décision dans cet espace [Shawe-Taylor and Cristianini, 2004]. La stratégie adoptée alors est de représenter les données d'entrée dans un nouvel espace, dans lequel des relations linéaires peuvent être facilement trouvées. La transformation faite dans la première composante est définie implicitement par la fonction noyaux. Elle dépendra du type spécifique des données et de la connaissance du domaine. La deuxième composante ne dépend ni des données, ni du domaine d'application, il s'agit d'un algorithme robuste et d'usage général (SVMs ou k-PPV par exemple). Cet

1. Selon le type de la tâche, il peut s'agir d'un hyperplan, de fonction linéaire de régression ou toute autre forme linéaire.

algorithme opère sur les données transformées, c'est-à-dire dans l'espace de reproduction, mais le calcul des produits scalaires se fait en utilisant la fonction noyau (astuce du noyau) dans l'espace d'entrée. Les aspects clés d'une méthode à noyaux peuvent être résumés comme suit :

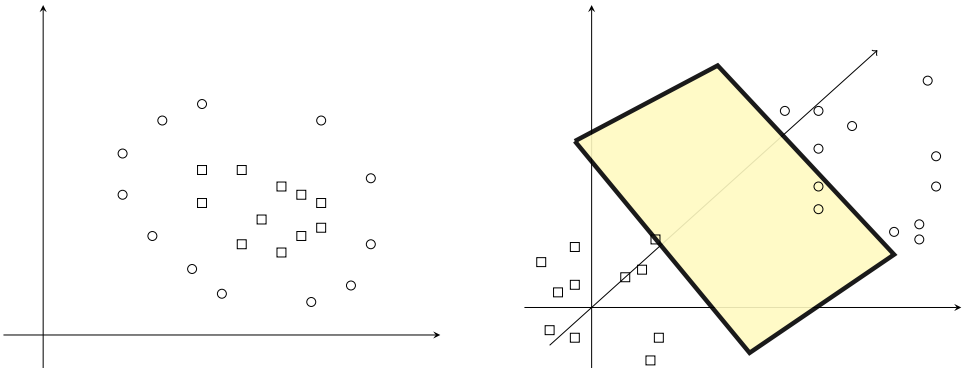
(i) Les données sont représentées dans un nouvel espace vectoriel de reproduction. La dimension de cet espace est d'habitude très grande, voir infinie.

(ii) Les relations linéaires sont identifiées entre les images des données dans l'espace de reproduction.

(iii) Les algorithmes d'apprentissage sont implémentés de telle sorte qu'on aura pas besoin de calculer explicitement les coordonnées des images, seules leurs produits internes sont considérés.

(iv) Les produits internes entre ces images peuvent être calculés de manière efficace directement dans l'espace d'entrée, en utilisant les fonction noyaux ou l'astuce du noyaux.

La figure 2.1 illustre ce principe à travers une transformation ϕ appliquée sur un jeu de données non linéairement séparable (figure 2.1(a)), pour lequel on peut facilement trouver un plan de séparation après reproduction sur le nouvel espace (figure 2.1(b)).



(a) Les données avant la transformation.

(b) Les données après la transformation.

FIGURE 2.1: Transformation non linéaire des données.

Modularité

L'aspect modulaire des méthodes à noyaux se montre dans la réutilisation de l'algorithme d'apprentissage [Shawe-Taylor and Cristianini, 2004]. Il est clair que

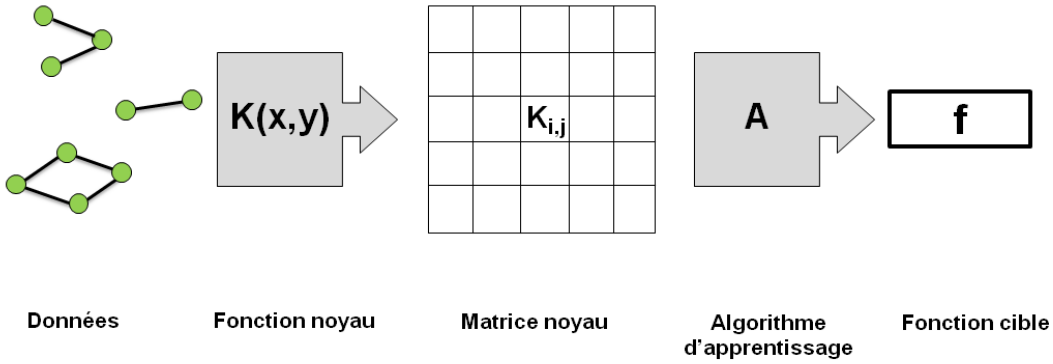


FIGURE 2.2: Différentes étapes d'une méthode à noyaux.

le même algorithme peut être utilisé avec n'importe quel noyau. La figure 2.2 montre les différentes étapes d'implémentation d'une méthode à noyaux [Shawe-Taylor and Cristianini, 2004]. Les données, dans leur forme de base, sont traitées et transformées en utilisant un noyau pour créer une matrice carrée dite *matrice noyau*. Cette matrice enregistre les distances entre les données telles que mesurées par la fonction noyau. Ensuite, cette matrice est utilisée par l'algorithme d'apprentissage pour produire la fonction cible f, qui sera utilisée dans la phase de production du système (faire des prédictions par exemple).

Comme nous venons de le constater, l'algorithme d'apprentissage dans une méthode à noyau admet en entrée une matrice noyau. Dans ce qui suit on expliquera comment construire cette matrice et quelles sont ses propriétés.

2.1.3 Exemple de méthodes à noyaux : Séparateurs à Vastes Marges (SVMs)

Les SVMs (pour Séparateurs à Vastes Marges ou Support Vector Machines en anglais) constituent une méthode de classification binaire par apprentissage supervisé, introduits par Vapnik en 1995 [Cortes and Vapnik, 1995]. Cette méthode représente une alternative récente pour la classification. Elle consiste à séparer deux ensembles d'exemples (appelés exemples positifs et exemples négatifs) par un hyperplan dans un espace approprié, donc elle repose essentiellement sur l'existence de cet hyperplan, et fait appel à un jeu de données d'apprentissage pour apprendre le modèle de séparation. Elle utilise également une fonction noyau qui permet une séparation optimale des données. En considérant les données d'apprentissage, nous distinguons deux cas de figures : données linéairement

séparables et données non linéairement séparables.

Dans ce qui suit, nous expliquerons le principe de fonctionnement des SVMs ainsi que le formalisme qui lui est dédié, puis nous aborderons l'aspect mathématique des SVM.

Étant donné un ensemble d'exemples (ou échantillon) étiqueté avec deux labels, le but des SVMs est de trouver un classificateur qui sépare ces deux classes de données, tout en maximisant la distance entre ces deux classes. Avec les SVM, ce classificateur est un modèle linéaire appelé *hyperplan*. La figure 2.3 montre un hyperplan qui sépare les deux ensembles de points.

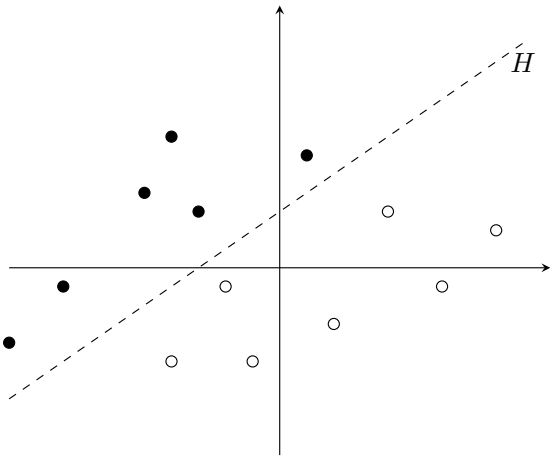


FIGURE 2.3: Un hyperplan H séparant deux ensembles de points.

Les points les plus proches, qui seuls seront utilisées pour déterminer l'hyperplan, sont appelées *les vecteurs de support* (Figure 2.4).

Il est clair qu'il existe une multitude d'hyperplans valides, permettant une bonne séparation des points, mais la propriété remarquable des SVM c'est qu'elles cherchent, parmi ces hyperplans, celui qui permet une séparation optimale. Elles vont donc chercher, parmi les hyperplans valides, celui qui passe au milieu des points des deux classes, donc l'hyperplan le plus sure. En effet, supposons qu'un exemple n'ait pas été décrit parfaitement, une petite variation n'affecte pas sa classification si sa distance à l'hyperplan est grande. Plus formellement, cela revient à trouver un hyperplan dont la distance minimale aux exemples d'apprentissage est maximale. Cette distance est appelée *marge* entre l'hyperplan et les exemples.

L'hyperplan séparateur optimal est celui qui maximise la marge (Figure 2.5). Comme on cherche à maximiser cette marge, on parlera de séparateurs à vaste

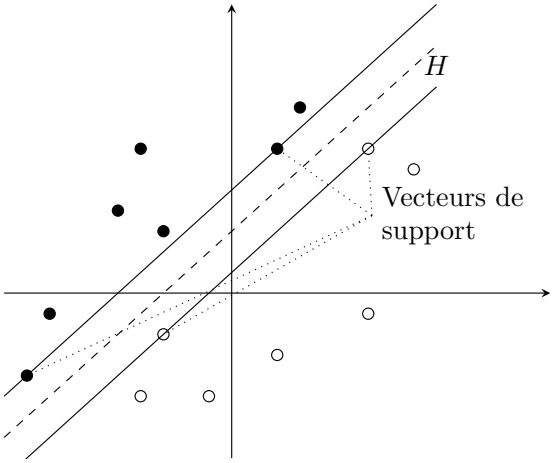


FIGURE 2.4: Les vecteurs de support.

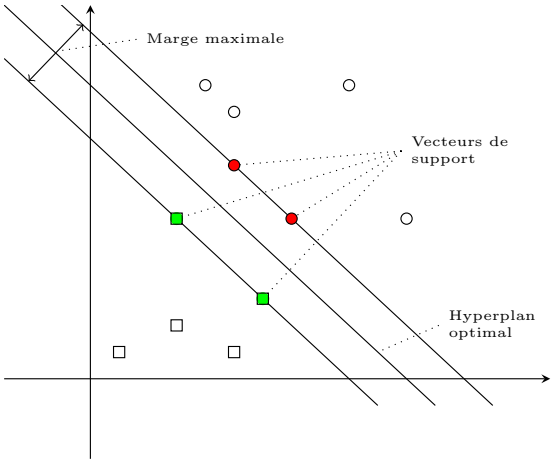
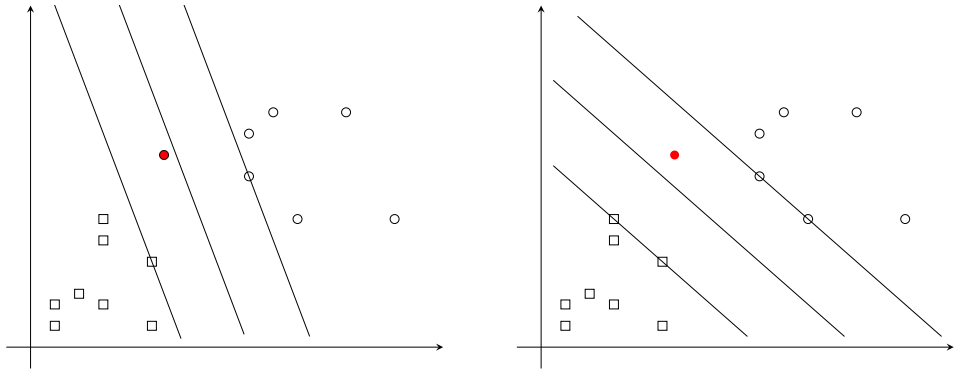


FIGURE 2.5: Hyperplan optimal et marge maximale.

marge.

Pourquoi maximiser la marge ?

Il est clair que le fait d'avoir une marge plus large procure plus de confiance lorsque l'on classe un nouvel exemple. De plus, si l'on trouve le classificateur qui se comporte le mieux vis-à-vis des données d'apprentissage, il est clair aussi qu'il sera celui qui permettra au mieux de classer les nouveaux exemples. Dans la figure 2.6, la partie droite nous montre qu'avec un hyperplan optimal, un nouvel exemple reste bien classé même s'il tombe dans la marge. On constate sur la partie gauche qu'avec une plus petite marge, l'exemple se voit mal classé.



(a) Hyperplan avec une faible marge

(b) Hyperplan avec une meilleure marge

FIGURE 2.6: Hyperplans à faible et meilleure marges.

Un nouvel exemple inconnu sera classé suivant sa position par rapport à l'hyperplan optimal. Dans la figure suivante (Figure 2.7), l'élément en bleu sera classé dans la catégorie des carrés.

Données linéairement séparables et non linéairement séparables

Suivant la distribution du jeu de données, on distingue deux cas de modèles SVM, cas linéairement séparable et cas non linéairement séparable (Figure 2.8). Les premiers sont les plus simple car il est facile de trouver un classificateur linéaire. Naturellement, un grand nombre de jeux de données sont non linéairement séparables, le classificateur de marge maximale ne peut pas être utilisé car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables.

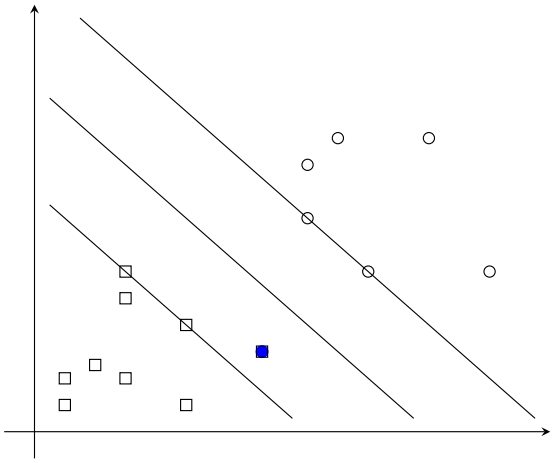
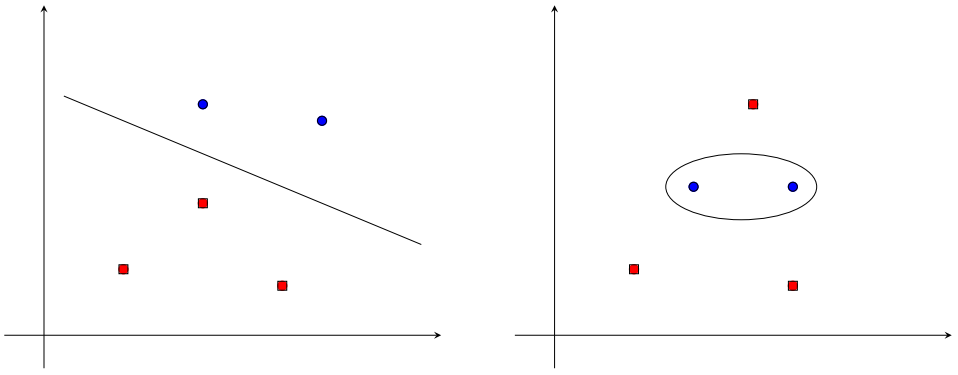


FIGURE 2.7: Nouvel élément à classer.



(a) Cas de données linéairement séparables.

(b) Cas de données non linéairement séparables.

FIGURE 2.8: Linéarité et non linéarité des données.

L'idée retenue par SVM, pour surmonter le problème des données non linéairement séparables, va dans un autre sens : on va tenter de trouver un transformation (ou mapping, en anglais) de l'espace d'entrée vers un autre espace (de re-description) dans lequel les données deviennent linéairement séparables. La figure 2.1 donne une représentation imagée de ce genre de transformation. L'espace de re-description aura une dimension plus grande que l'espace d'entrée, il est donc plus probable de trouver un hyperplan séparateur des exemples de l'échantillon dans cet nouvel espace.

Une transformation est donc faite d'un problème de séparation non linéaire, dans l'espace d'entrée, à un problème de séparation linéaire, dans un espace de re-description, de plus grande dimension. Cette transformation est réalisée via la fonction noyau (voir section 2.1.1).

Fonction Noyau

l'algorithme de construction de l'hyperplan proposé par *Vapnik*, en 1963, est un classificateur linéaire. Par contre, en 1992, *Bernard Boser, Isabelle Guyon* et *Vapnik* ont proposé une autre méthode pour construire un classificateur non linéaire en appliquant la fonction *noyau* (originellement proposée par Aizerman) pour maximiser la marge des hyperplans.

On tente de trouver une transformation (mapping) de l'espace d'entrée vers un autre espace, l'space de sortie, dans lequel les données seront linéairement séparables. Nous pouvons alors appliquer une méthode à marge maximale dans le nouvel espace. La dimension du nouvel espace est généralement très élevée. Cela ne pose pas de problème pour le classificateur à marge maximale vu que sa formulation duale fixe le nombre de variables à déterminer en fonction de la taille de l'échantillon. Nous noterons le nouvel espace par F, et la transformation vers cet espace par :

$$\phi : X \longrightarrow F$$

Mesure de similarité

De manière générale, il peut être utile de savoir à quel point un exemple est similaire à un autre. Pour faire cela, on utilise souvent en mathématique le produit scalaire qui moyennant une normalisation, correspond au cosinus de l'angle entre deux vecteurs.

En utilisant la transformation ϕ, on peut définir une mesure de similarité dans le nouvel espace :

$$K(x, z) = \langle \phi(x), \phi(z) \rangle$$

La fonction K est dite noyau (voir section 2.1.1).

Pour calculer l'hyperplan optimal dans le nouvel espace, il suffit de remplacer toutes les occurrences du produit scalaire par le noyau. Plus généralement, tout algorithme d'apprentissage accédant exclusivement aux exemples à travers le produit scalaire (ou d'une grandeur qui en dérive) est dit *kernelisable*, i.e, on peut substituer les produits scalaires par une fonction noyau. Le produit scalaire lui même peut être vu comme un noyau dont la transformation ϕ est l'identité.

Le produit scalaire calculé dans le nouvel espace peut être très couteux en temps de calcul étant donné que sa complexité est linéaire en la dimension de F et que cette dernière peut être très élevée. Cet inconvénient peut rendre le calcul de l'hyperplan optimal fastidieux, voire impossible. Remarquons que, si l'on peut déterminer une autre forme plus économique pour la fonction $K(x, z)$, on peut se passer de l'utilisation explicite de ϕ. En effet, le résultat du produit scalaire étant un réel, une autre fonction à image dans $\mathbb{R}$ peut être utilisée.

2.2 Noyaux rationnels

Dans le contexte de la classification, plusieurs algorithmes ont été originalement conçu pour traiter des vecteurs de taille fixée. Cependant, les applications récentes, telles que la classification de documents, l'analyse des données biologiques ou de la parole, nécessitent une analyse de séquences de longueurs variables et plus généralement des automates pondérées [Cortes et al., 2004]. En effet, le résultat d'un système complexe d'extraction d'information, combinant plusieurs sources de connaissances, pour répondre à une requête, est typiquement un automate pondéré qui représente d'une manière compacte un ensemble de réponses alternatives. Les poids affectés par le système aux différentes réponses peuvent être utilisés pour ordonner ces réponses. Aussi, dans un système de reconnaissance vocale, la sortie dans de tels systèmes est un automate pondéré comportant plusieurs séquences.

Par ailleurs, les méthodes à noyaux sont bien adaptées à ce type de tâches d'apprentissage, vue leur efficacité de calcul et leur comportement dans des espaces de représentation à grandes dimensions. Dans cette section, on présente les noyaux rationnels, qui sont une généralisation des noyaux, basée sur les transducteurs pondérés. Ce type de noyaux permettra d'étendre les méthodes à noyaux

pour l'analyse des données de tailles variables, ou de manière plus générale, aux automates pondérées.

2.2.1 Transducteurs de mots

Les *transducteurs de mots* sont une forme généralisée des automates finis. En effet, chaque transition dans l'automate fini est augmentée par une étiquette de sortie en plus de son étiquette d'entrée. Les étiquettes en sortie sont assemblées le long d'un chemin de l'état initial vers un état final, pour former une chaine de sortie. Les transducteurs pondérés portent sur leurs transitions, en plus des deux étiquettes d'entrée et de sortie, une valeur numérique appelée poids de la transition. Le poids d'une paire de chaines d'entrée et de sortie (x, y) est calculé en additionnant les poids des chemins vers des états finaux, étiquetées par (x, y). Avant de définir formellement les transducteurs pondérés, on présente les notations et définitions algébriques nécessaires.

Définition 2.3. *Un système* $(\mathbb{K}, \odot, e)$ *est dit monoïde s'il vérifie les propriétés suivantes :*

(i) $\forall\ a, b \in \mathbb{K} : a \odot b \in \mathbb{K}$ *(Stabilité).*

(ii) $\forall\ a, b, c \in \mathbb{K} : (a \odot b) \odot c = a \odot (b \odot c)$ *(Associativité).*

(iii) $\exists\ e \in \mathbb{K}, \forall\ a \in \mathbb{K} : a \odot e = e \odot a = a$ *(Existence d'un élément neutre).*

Si, de plus, $\odot$ *est commutative;* $\forall\ a, b \in \mathbb{K} : a \odot b = b \odot a$*, alors* $(\mathbb{K}, \odot, e)$ *est dit monoïde commutatif.*

Définition 2.4. *Un système* $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ *est dit semi-anneau si :*

i) $(\mathbb{K}, \oplus, \bar{0})$ *est un monoïde commutatif avec élément neutre* $\bar{0}$*.*

ii) $(\mathbb{K}, \otimes, \bar{1})$ *est un monoïde avec élément neutre* $\bar{1}$*.*

iii) $\otimes$ *est distributive sur* $\oplus$*.*

iv) $\bar{0}$ *est un élément absorbant pour l'opération* $\otimes$*;* $\forall\ a \in \mathbb{K}, a \otimes \bar{0} = \bar{0} \otimes a = \bar{0}$*.*

Le tableau 2.1 liste des exemples usuels de semi-anneaux utilisés dans la pratique. Pour le semi-anneau logarithmique, l'opération $\oplus_{\log}$ est définie comme suit :

$$x \oplus_{\log} y = -\log(e^{-x} + e^{-y})$$

Les transducteurs pondérés [Berstel, 1979][Cortes et al., 2007] sont définis comme suit :

Semi-anneau	$\mathbb{K}$	$\oplus$	$\otimes$	$\bar{0}$	$\bar{1}$
Booléen	$\{0, 1\}$	$\vee$	$\wedge$	0	1
Probabilité	$\mathbb{R}_+$	$+$	$\times$	0	1
Logarithmique	$\mathbb{R} \cup \{-\infty, +\infty\}$	$\oplus_{\log}$	$+$	$+\infty$	0
Tropical	$\mathbb{R} \cup \{-\infty, +\infty\}$	min	$+$	$+\infty$	0

Tableau 2.1: Exemples de semi-anneaux usuels.

Définition 2.5. *Un transducteur pondéré T sur le semi-anneau $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ est défini par :*
$T = (\Sigma, \Delta, Q, I, F, E, \lambda, \rho)$ où Σ est l'alphabet d'entrée, Δ est l'alphabet de sortie, Q est un ensemble fini d'états, $I \subseteq Q$ l'ensemble des états initiaux, $F \subseteq Q$ l'ensemble des états finaux, $E \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times (\Delta \cup \{\varepsilon\}) \times \mathbb{K} \times Q$ un ensemble fini de transitions, $\lambda : I \to \mathbb{K}$ fonction de pondération des états initiaux, et $\rho : F \to \mathbb{K}$ la fonction de pondération des états finaux.

Pour un chemin π dans le transducteur, $p[\pi]$ et $n[\pi]$ dénotent, respectivement, l'état d'origine et l'état destination de ce chemin. $w[\pi]$ donne la somme des poids des transitions de ce chemin. On dénote par $P(I, x, y, F)$, l'ensemble des chemins d'un état initial de I vers un état final de F étiquetés par la chaine d'entrés x et la chaine de sortie y. Un transducteur T est dit *régularisé* si le poids de sortie affecté par T à toute paire de chaine (x, y) donné par :

$$[\![T]\!](x, y) \quad = \bigoplus_{\pi \in P(I,x,y,F)} \lambda(p[\pi]) \otimes w[\pi] \otimes \rho[n[\pi]] \tag{2.1}$$

est bien défini dans $\mathbb{K}$. Si $P(I, x, y, F) = \emptyset$ alors $[\![T]\!](x, y) = \bar{0}$. La figure 2.9 montre un exemple d'un transducteur simple. $[\![T]\!](abb, baa)$ est la somme des poids de tous les chemins, vers un état final, étiquetés par $x = abb$ et $y = baa$. Donc, $[\![T]\!](abb, baa) = 0.1 \times 0.2 \times 0.3 \times 0.1 + 0.5 \times 0.3 \times 0.6 \times 0.1$

Propriétés des transducteurs de mots

Les transducteurs régularisés sont clos sous les opérations suivantes, appelées opérations rationnelles :

— La *somme* (ou *union*) de deux transducteurs pondérés T_1 et T_2 est définie par :

$$\forall (x, y) \in \Sigma^* \times \Sigma^*, [\![T_1 \oplus T_2]\!](x, y) \quad = [\![T_1]\!](x, y) \oplus [\![T_2]\!](x, y) \tag{2.2}$$

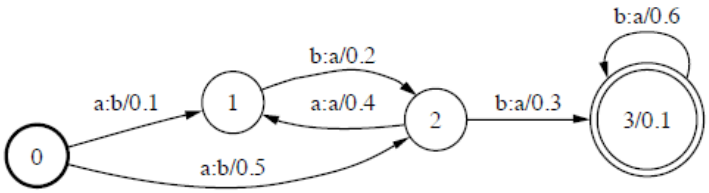


FIGURE 2.9: Exemple d'un transducteur pondéré.

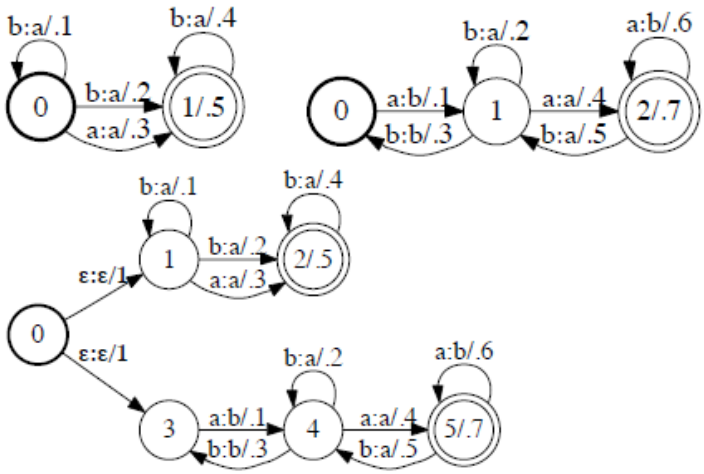


FIGURE 2.10: Somme de deux transducteurs.

— Le *produit* (ou *la concaténation*) de deux transducteurs pondérés T_1 et T_2 est définie par :

$$\forall (x,y) \in \Sigma^* \times \Sigma^*, [\![T_1 \otimes T_2]\!](x,y) = \bigoplus_{x=x_1x_2, y=y_1y_2} [\![T_1]\!](x_1,y_1) \otimes [\![T_2]\!](x_2,y_2) \tag{2.3}$$

— La *composition* de deux transducteurs pondérés T_1 et T_2 en mettant en correspondance l'alphabet d'entrée et de sortie Σ, est un transducteur pondéré dénoté par $T_1 \circ T_2$ lorsque la somme :

$$[\![T_1 \circ T_2]\!](x,y) = \bigoplus_{z \in \Sigma^*} [\![T_1]\!](x,z) \otimes [\![T_2]\!](z,y) \tag{2.4}$$

est bien définie sur $\mathbb{K}$ pour toute paire $x, y \in \Sigma^*$

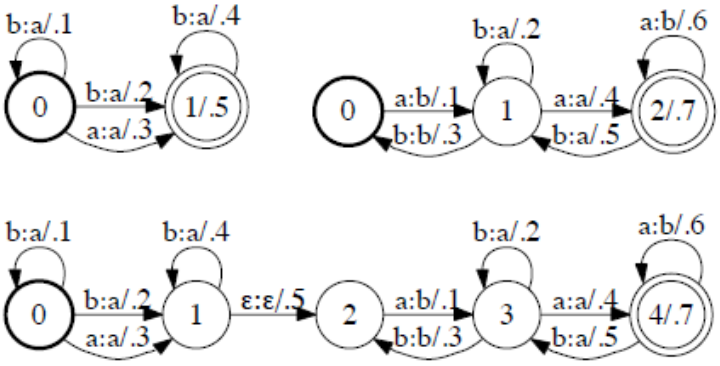


FIGURE 2.11: Produit de deux transducteurs.

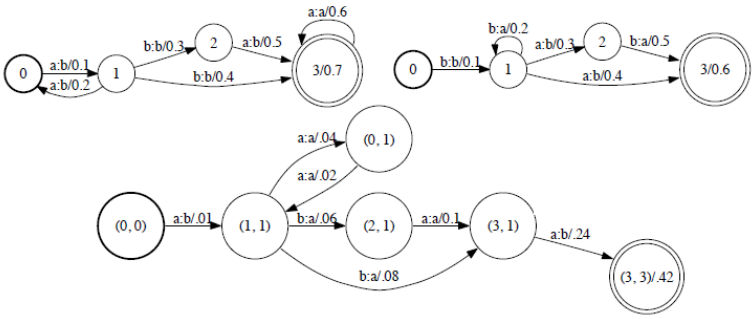


FIGURE 2.12: Composition de deux transducteurs.

— L'*étoile de Kleen* d'un transducteur T, notée T^* est un transducteur défini par :

$$[\![T^*]\!](x, y) = \Sigma_{n=0}^{+\infty}[\![T^n]\!](x, y) \tag{2.5}$$

Ces opérations sont utilisées pour créer des transducteurs pondérés complexes à partir de transducteurs plus simples. Les algorithmes utilisés pour réaliser ces opérations sont simples et efficaces.

2.2.2 Transducteurs et noyaux

Les noyaux rationnels représentent une famille généralisée des noyaux, en se basant sur les transducteurs pondérés. Ils étendent les méthodes à noyaux à l'analyse des séquences de tailles variables et plus généralement aux automates pondérées. Soient X et Y deux ensembles non vides. Une fonction $K : X \times Y \to \mathbb{R}$

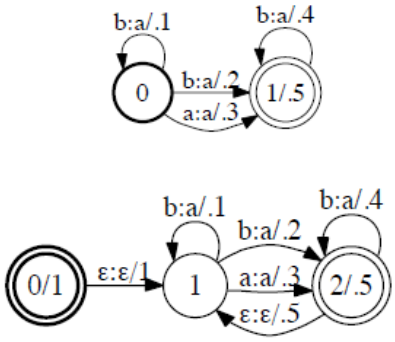


FIGURE 2.13: Étoile de Kleen d'un transducteur.

est qualifiée comme un noyau sur $X \times Y$. Cortes *et, al.* [Cortes et al., 2004] donnent une définition formelle pour les noyaux rationnels :

Définition 2.6. *Un noyau K sur $\Sigma^* \times \Delta^*$ est dit rationnel s'il existe un transducteur pondéré $T = (\Sigma, \Delta, Q, I, F, E, \lambda, \rho)$ défini sur un semi anneau $\mathbb{K}$ et une fonction $\varphi : \mathbb{K} \to \mathbb{R}$ tel que pour tout $x \in \Sigma^*$ et $y \in \Delta^*$:*

$$K(x, y) = \varphi([\![T]\!](x, y)) \tag{2.6}$$

K est alors défini par la paire (φ, T).

Le noyau $K(x, y)$ se calcule en utilisant le transducteur T comme suit :

$$K(x, y) = \varphi(A_x \circ T \circ A_y) \tag{2.7}$$

où A_x (resp. A_y) est l'automate pondéré acceptant uniquement x (resp. y).

En général, le calcul en utilisant l'opération de composition et un algorithme de plus courte distance, permet d'avoir une complexité de l'ordre de $O(|x||y|)$. Pour des cas plus spécifiques, l'utilisation d'une composition plus efficace peut engendrer une complexité meilleure [Allauzen, Cyril and Mohri, Mehryar, 2008].

Les noyaux rationnels peuvent être étendus aux automates pondérés. Soit A (resp. B) un automate pondéré défini sur le semi-anneau $\mathbb{K}$ et l'alphabet Σ (resp. l'alphabet Δ). $K(A, B)$ mesure la similarité entre les deux automates pondérés A et B, à travers le noyau K. Elle est défini par :

$$K(A, B) = \varphi(\bigoplus_{(x,y) \in \Sigma^* \times \Delta^*} [\![A]\!](x) \otimes [\![T]\!](x, y) \otimes [\![B]\!](y)) \tag{2.8}$$

tel que la somme : $\bigoplus_{(x,y)\in\Sigma^*\times\Delta^*}[\![A]\!](x)\otimes[\![T]\!](x,y)\otimes[\![B]\!](y)$ est bien définie et à valeurs dans $\mathbb{K}$. En effet, cette somme est toujours définie lorsque A et B sont des automates pondérés acycliques. L'équation (2.8) peut s'écrire comme suit :

$$K(A,B) = \varphi(\bigoplus_{(x,y)\in\Sigma^*\times\Delta^*}[\![A\circ T\circ B]\!](x,y)) \tag{2.9}$$

Parmi les noyaux rationnels existants, nous nous intéressons à ceux qui vérifient la condition de Mercer, ou en d'autres mots, ceux qui sont symétriques définis positifs. Ce type de noyaux rationnels assurent la convergence des algorithmes d'apprentissage vers une solution optimale unique [Cortes et al., 2004]. Le théorème suivant caractérise les noyaux rationnels symétriques définis positifs, en se basant sur les transducteurs inverses.

Définition 2.7. *Soit T un transducteur. Le transducteur inverse de T, noté T^{-1} est obtenu à partir de T en inversant les étiquettes d'entrée et sortie sur les transitions.*

Théorème 2.1. *Pour tout transducteur T, la fonction $K = [\![T\circ T^{-1}]\!]$ est un noyau rationnel symétrique défini positif.*

Ceci signifie que toute fonction pouvant s'écrire sous forme d'une composition d'un transducteur avec son inverse, est considérée comme noyau rationnel symétrique défini positif.

2.2.3 Exemple de noyaux rationnels

Les noyaux rationnels peuvent être utilisés dans plusieurs types d'applications, utilisant divers types de données, telles que les chaînes de caractères, les images, les graphes ou les automates pondérées. L'exemple suivant concerne un mode de représentation des données et connaissances très utilisé dans la pratique, à savoir : les données textuelles. On expliquera comment mesurer la similarité entre deux instances de documents en utilisant les noyaux rationnels.

Noyaux rationnels pour les données textuelles

On considère deux chaines de caractères $s = abac$ et $t = abab$, construites sur l'alphabet $\Sigma = \{a, b, c\}$. Sachant que plus ces deux chaines possèdent des sous-chaines communes, plus elles sont similaires, alors on peut choisir le noyau des N-grammes comme mesure de similarité. Cette situation est très courante dans des champs d'applications tels que la bio-informatique, la recherche d'information ou la classification de documents. Le noyau des bi-grammes est modélisé à travers le transducteur T_{bigram} (figure 2.14). Ce transducteur donne tous les bi-grammes d'une chaine en entrée. Pour

calculer le nombre de bi-grammes en commun entre les deux chaines s et t, il suffit de faire :

$$\varphi((A_s \circ T_{bigram}) \circ (T_{bigram}^{-1} \circ A_t))$$

sachant que A_s (resp. A_t) est un automate acceptant uniquement la chaine s (resp. la chaine t) (figure 2.15).

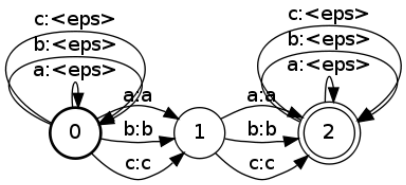


FIGURE 2.14: Transducteur compteur des bi-grammes avec $\Sigma = \{a, b, c\}$.

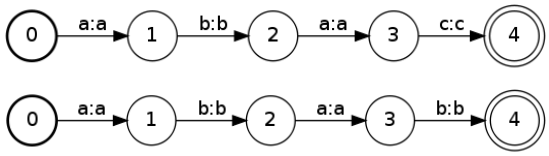


FIGURE 2.15: Automates linéaires acceptant les chaine s et t.

En effet, la première partie de la composition : $A_s \circ T_{bigram}$ donne un transducteur dont les chemins de l'état initial vers l'état final représentent tous les bi-grammes de la chaine s (figure 2.16). Pour extraire ces bi-grammes il suffit de parcourir les chemins en gardant les symboles en sortie. Par exemple, le premier chemin (en haut de la figure 2.16) dans ce transducteur donne en sortie le bi-gramme ac, le deuxième chemin (celui au milieu) donne le bi-gramme ba, tandis que le dernier (en bas) donne le bi-gramme ab. La deuxième partie de la composition : $T_{bigram}^{-1} \circ A_t$, donne un transducteur dont les chemins représentent tous les bi-grammes de la chaine t (figure 2.17). La composition des deux parties donne les bi-grammes communs entre les deux chaines (figure 2.18).

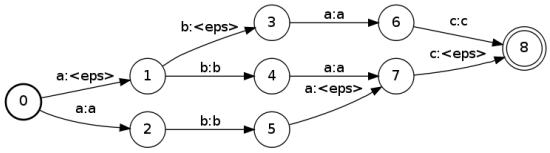


FIGURE 2.16: Transducteur donnant les bi-grammes de la chaine s.

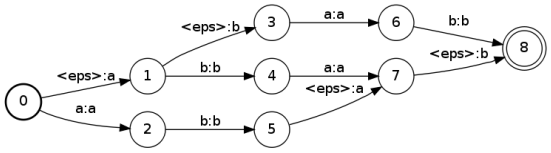


FIGURE 2.17: Transducteur donnant les bi-grammes de la chaine t.

Pour certains types d'applications, il est question de comparer non pas des instances de données séparées, mais des ensembles d'instances entre eux. Un exemple typique est celui des langages rationnels. Les ensembles d'instances sont alors représentés par des automates pondérés. Le noyau des N-grammes peut être utilisé pour ce type de données en remplaçant simplement les automates A_t et A_s par deux automates acceptant des langages.

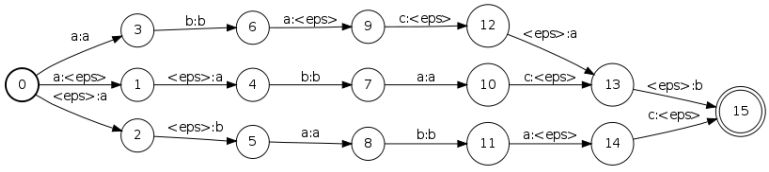


FIGURE 2.18: Transducteur donnant les bi-grammes communs entre les chaines s et t.

2.3 Conclusion

Dans ce chapitre, nous avons exposé deux outils théoriques importants pour l'apprentissage automatique, à savoir : les méthodes à noyaux et les noyaux rationnels.

Les méthodes à noyaux ont prouvé leur efficacité dans le domaine de l'apprentissage automatique, notamment pour les tâches de classification. L'avantage majeur de ces méthodes c'est qu'elles peuvent être étendues pour des données non structurées, telles que les documents à taille variable, les graphes ou les automates. Il suffit pour cela de choisir, ou définir, un noyau adéquat pour la tâche et le type des données en question.

Les noyaux rationnels représentent une forme généralisée des noyaux, dans laquelle les transducteurs ont été utilisés. Ils présentent l'avantage d'être facilement utilisés pour définir de nouveaux noyaux moyennant des opérations rationnelles.

Chapitre 3

Classification de documents en langue arabe

La classification de documents en langue arabe a commencé à attirer l'attention des chercheurs depuis le début des années 2000. Les systèmes proposés suivent le même schéma général que pour les langues latines. Cependant, la nature et les spécificités de la langue arabe font que la première partie du système de classification contient, en plus des tâches indépendantes de la langue, des traitements liés à la langue arabe. La nature flexionnelle et la richesse de la langue arabe, constituent aussi un challenge supplémentaire. Ceci a induit plusieurs travaux à réaliser pour l'analyse morphologique des mots arabes.

Dans ce chapitre, on commence par exposer les difficultés de la classification de documents en langue arabe. Les différentes techniques de racinisation et d'extraction de radicaux proposées dans la littérature feront l'objet de la section 3.2. Dans la section 3.3, un état de l'art des systèmes de CDA proposés dans la littérature est présenté selon un ordre chronologique.

3.1 Difficultés de la classification de documents en arabe

Le traitement automatique de la langue arabe, en général, et la classification de documents en arabe, en particulier, font face à de nombreux défis :

— Le premier défi est lié à l'analyse morphologique de la langue arabe, qui constitue un outil crucial pour les systèmes de classification de documents en arabe. En effet, le processus de la CDA dépend du contenu des documents, un nombre important de caractéristiques peut conduire à une mauvaise performance en termes de précision et de temps de traitement.

L'arabe est une langue possédant un lexique très riche, un nombre important de mots peut être généré à partir d'un lemme ou une racine. En traitant tous les mots des documents, le système de classification va se retrouver avec un très grand nombre de caractéristiques. Une solution consiste à faire des traitements

au niveau morphologique, telles que la racinisation et l'extraction des radicaux. Ce traitement morphologique permet de réduire le nombre de caractéristiques, ce qui améliore les performances du système. Cependant, il peut induire des erreurs de racinisation ou d'extraction de racines pouvant causer une dégradation des performances.

— Le deuxième type de défis concerne le niveau sémantique de la langue arabe. La classification de documents est sensible à la signification des expressions. La richesse morphologique et l'ambiguïté orthographique, en raison de la nature diacritique optionnelle de la langue arabe, peuvent conduire à un grand nombre d'homographes et homonymes [Habash, 2010]. Les synonymes sont également répandus en langue arabe, ils viennent compliquer la tâche des systèmes de classification du fait que le même concept peut être exprimé par plusieurs termes différents.

— Le troisième défi est lié à l'insuffisance des corpus arabes libres pour évaluer les systèmes de CDA. Plusieurs travaux ont été réalisés sur des ensembles de données non standards et recueillis via des sites Web.

Dans ce qui suit, nous allons donner plus de détails concernant ces défis.

L'analyse morphologique est l'étude de la structure interne du mot. Morphologiquement, la langue arabe est la plus compliquée et la plus riche des langues [Habash, 2010]. Plusieurs mots peuvent être formés en utilisant la même racine, quelques modèles, et quelques affixes. Un des défis pour l'extraction de la racine est que les mots, en arabe moderne, sont exempts de signes diacritiques, ce qui les rend plus ambigus. Par exemple, les deux mots (كَتَبَ, il a écrit) et (كُتُبٌ, des livres) sont extraits de la même racine, mais ont des significations différentes lorsque ils sont vocalisés. De plus, l'absence de vocalisation peut engendrer des cas d'ambiguïté plus importants. Prenons par exemple les deux mots suivants : (يَنَعَتْ, mûrit) et (يَنْعِتُ, Qualifier). Ces mots sont extraits de deux racines différentes (ينع et نعت respectivement) pourtant ils sont identiques orthographiquement.

Les affixes et les clitiques multiples peuvent apparaître dans un mot, en raison de la nature agglutinante de l'arabe, donnant des formes de mots qui sont traduits, par exemple en français, à toute une phrase. Par exemple, le mot arabe (أَنُلْزِمُكُمُوهَا) est traduit en "Allons-nous vous contraindre à accepter".

Les racines avec des radicaux géminées ou faibles ont besoin de règles spécifiques lors de l'analyse. Dans le premier type de racines, l'une des lettres doublées est enlevée dans la forme finale du mot, l'algorithme de récupération de la racine doit gérer ce cas. Par exemple, le mot (شدّت, elle tire) est généré à partir de la racine (ش د د). Concernant les radicaux faibles, les choses sont plus compliqués. Les radicaux faibles (و ي) se transforment en une voyelle ou sont supprimés en fonction de leur contexte vocalique [Habash, 2010]. La principale difficulté lors de l'extraction des racines est de décider si un mot

est généré à partir de radicaux faibles. Le mot (التنمية, le développement) est généré à partir de la racine (ن م و), nous pouvons remarquer que la lettre (و) ne figure pas dans le mot.

Les fautes d'orthographe sont fréquentes dans les documents Web en arabe. Les erreurs évidentes peuvent être manipulées facilement par les outils d'analyse. Toutefois, les erreurs non évidentes peuvent rendre les systèmes de CDA moins efficaces. Ce type d'erreurs peut être difficile à identifier si le mot mal orthographié se trouve être un mot arabe valide. Par exemple, le mot mal orthographié (أقصد, je voulais dire) dont la première lettre (أ) a été supprimée, reste un mot arabe valide (قصد, Aller à).

Le pluriel et le féminin irréguliers, ainsi que de nombreuses autres particularités de la langue arabe, sont des facteurs qui compliquent encore l'analyse morphologique.

3.2 La racinisation et l'extraction de radicaux

La racinisation et l'extraction de radicaux sont considérées comme parties importantes dans les systèmes de CDA. Elles sont appliquées pour réduire la dimension des vecteurs de représentation des documents. L'extraction de radicaux (en anglais, est souvent connus sous le terme : "root-based stemming") permet de réduire chaque mot dans le document à ses radicaux. Par contre, la racinisation (en anglais : "stem-based stemming") permet juste de supprimer les préfixes et suffixes du mot [Aljlayl and Frieder, 2002].

Nous avons étudié la quasi-totalité des travaux proposés dans la littérature. Nous pouvons classer les travaux recensés comme l'indique la figure 3.1. En effet, les techniques proposées dans la littérature peuvent être principalement classées en deux catégories ; techniques supervisées et techniques non supervisées. Les techniques supervisées sont basées sur des connaissances linguistiques, telles que les règles morphologiques et les modèles grammaticaux. Les techniques non supervisées reposent sur des modèles statistiques de la langue ou sur l'apprentissage non supervisé. Dans les techniques supervisées, le système fait appel à des connaissances supplémentaires de la langue, telles que les dictionnaires des préfixes, suffixes, modèles (anglais : patterns) ou les listes des racines et les règles linguistiques. Par contre, dans les techniques non supervisées on tente de se passer des connaissances linguistiques, juste le texte brute est utilisé.

3.2.1 Techniques d'extraction des radicaux

Plusieurs techniques d'extraction de radicaux ont été intégrées dans les systèmes de CDA. Ces techniques peuvent être classées en deux groupes, selon qu'elles utilisent ou pas des ressources linguistiques.

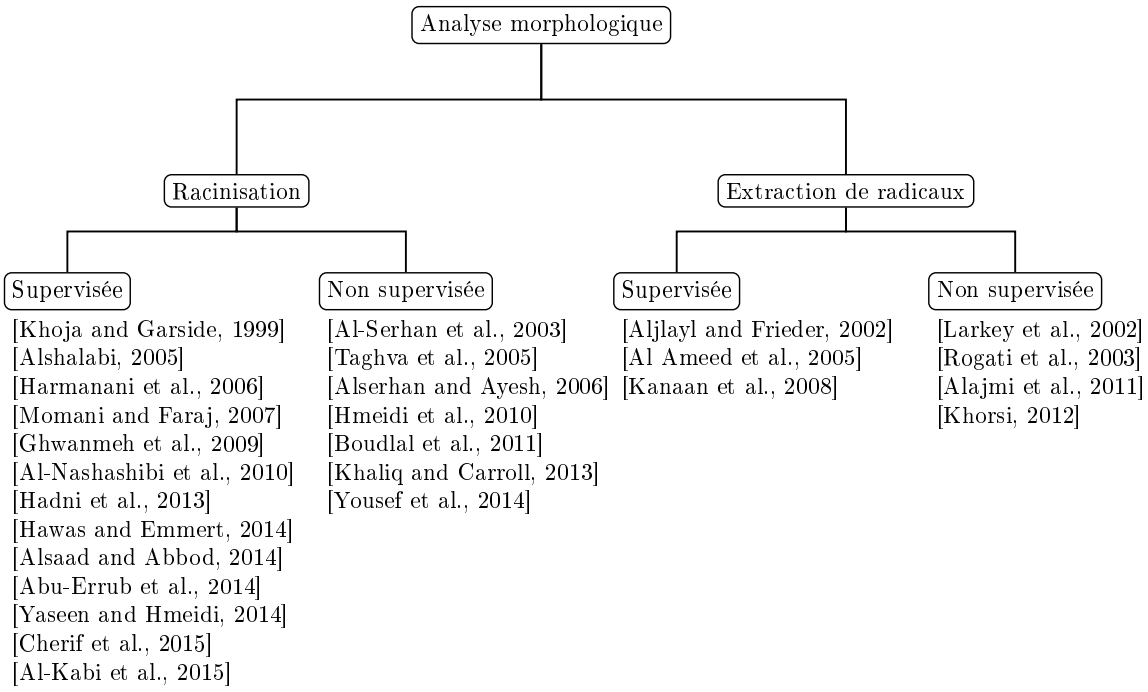


FIGURE 3.1: Classification des techniques de racinisation et d'extraction de radicaux.

3.2.1.1 Techniques supervisées d'extraction de radicaux

Dans cette classe de techniques, les ressources disponibles, à savoir les listes des affixes, patterns, racines et des règles linguistiques de la langue arabe, sont exploitées pour analyser morphologiquement un mot.

L'une des premières techniques proposées est l'analyseur morphologique de Buckwalter, connu sous le nom de BAMA (Buckwalter Arabic Morphological Analyzer) [Buckwalter, 2004]. Il est considéré comme la ressource lexicale la plus respectée dans ce domaine de recherche.

Cet analyseur adopte un système de translittération du mot arabe, qui a été développé par Buckwalter lui même. Ce système de translittération est conçu comme une base de données principale de formes des mots qui interagissent avec d'autres bases de données concaténées. Chaque forme de mot est entrée séparément. Il prend comme forme de base le radical (stem), ensuite il fournit des informations sur la racine. En ce qui concerne sa base de données lexicale, elle contient trois sous base : une base de données des préfixes qui contient 299 entrées, une base de données des suffixes qui contient 618 entrées et une base de données des lexèmes qui contient 82158 entrées représentant

ainsi 38600 lemmes [Buckwalter, 2004].

Une autre technique intéressante est celle due à [Khoja and Garside, 1999]. Dans cet algorithme, on tente de localiser et supprimer les plus longs préfixe et suffixe d'un mot, puis vérifier, par rapport à une liste de patterns, si le mot qui en résulte peut être identifié comme nom ou verbe valide. Cet outil repose sur de nombreuses ressources linguistiques, telles que les listes de tous les caractères diacritiques, de ponctuation, articles et mots outils, ainsi qu'un dictionnaire de toutes les racines valides. Il rapporte de bons résultats en terme de précision, mais nécessite un maintien des différentes ressources et dictionnaires.

Dans d'autres travaux [Harmanani et al., 2006], [Momani and Faraj, 2007], [Ghwanmeh et al., 2009], [Hawas and Emmert, 2014], [Alsaad and Abbod, 2014], [Abu-Errub et al., 2014] et [Cherif et al., 2015], une approche à base de règles a été utilisée. Par exemple, dans [Harmanani et al., 2006], la méthode proposée consiste à extraire la racine en se basant sur un ensemble de règles linguistiques, qui sont interprétées par un moteur de règles.

Dans [Alshalabi, 2005] et [Al-Nashashibi et al., 2010], en plus des règles linguistiques, une liste des modèles de mots est utilisée pour extraire les trois lettres radicaux après avoir supprimé les affixes du mot. Les travaux plus récents ont tendance à utiliser toutes les ressources disponibles (liste des racines, listes des modèles et règles linguistiques) pour extraire la racine d'un mot [Hadni et al., 2013, Yaseen and Hmeidi, 2014, Al-Kabi et al., 2015]. Cette tendance à utiliser toutes les ressources possibles indique que l'extraction des racines est toujours une tâche difficile.

3.2.1.2 Techniques non supervisées d'extraction de radicaux

Dans cette catégorie, l'extraction des radicaux consiste à utiliser les techniques d'apprentissage automatique, telles que les réseaux de neurones, et/ou des techniques statistiques, telles que les N-grammes [Al-Serhan et al., 2003], [Taghva et al., 2005], [Alserhan and Ayesh, 2006], [Hmeidi et al., 2010], [Boudlal et al., 2011], [Khaliq and Carroll, 2013], [Yousef et al., 2014].

La technique proposée dans [Al-Serhan et al., 2003], permet d'extraire les racines des mots arabes sans avoir besoin de ressources linguistiques. La racine d'un mot donné est extraite en attribuant un rang et un poids à chacune de ses lettres. Différents poids sont affectés aux lettres composant le mot (سألتمونيها) et le reste des consonnes sont assignées un poids égale à zéro (voir tableau 3.1) [1]. Le tableau 3.2 montre comment est calculé le rang d'une lettre dans un mot, selon sa position. L'algorithme choisit alors les trois lettres ayant les plus petits produits (poids × rang) comme radicaux. Le

1. Selon les auteurs de [Al-Serhan et al., 2003], ces poids ont été déterminés suite à une étude statistique.

tableau 3.3 montre l'utilisation de cet algorithme pour extraire les radicaux du mot (المحافظة, la préservation).

Lettre arabe	Poids
ا ة	5
ي يء	3.5
ت ى و	3
أ م ن	2
ل س ه	1
Le reste des lettres	0

Tableau 3.1: Poids assignés aux lettres de l'alphabet arabe.

Les auteurs de [Taghva et al., 2005] ont implémenté un algorithme d'extraction de radicaux qui partage beaucoup de caractéristiques avec celui de [Khoja and Garside, 1999], sauf qu'il n'utilise plus des ressources linguistiques. Dans [Alserhan and Ayesh, 2006], les auteurs exploitent les relations numériques entre les lettres, en utilisant un réseau de neurones à rétro-propagation, pour extraire les radicaux. Les modèles cachés de Markov ont été exploités dans [Boudlal et al., 2011], les N-grammes dans [Hmeidi et al., 2010] et [Yousef et al., 2014].

Ces travaux n'utilisent pas les mêmes corpus dans l'expérimentation. Par exemple, [Alserhan and Ayesh, 2006] et [Yousef et al., 2014] utilisent des ensembles de mots générés automatiquement, [Hmeidi et al., 2010] utilisent la liste des mots du Coran, tandis que [Boudlal et al., 2011] utilisent un corpus non libre (Nemlar Wirtten Corpus [2]). Cette diversité des corpus ne facilite pas la comparaison de leurs résultats.

3.2.2 Techniques de racinisation

Les mots dans la langue arabe sont construits, dans la plupart des cas, à partir de radicaux en appliquant des modèles pour former une racine, puis en rajoutant des affixes. Ces affixes peuvent être de quatre types : les antéfixes, préfixes, suffixes et poste-fixes [Kadri and Nie, 2006]. Ainsi, les mots en arabe peuvent avoir des formes

2. http ://www.nemlar.org/

Position de la lettre à partir de droite	Rang si taille du mot est paire	Rang si taille du mot est impaire
1	N	N
2	$N-1$	$N-1$
3	$N-2$	$N-2$
⋮	⋮	⋮
$[N/2]$	$N/2+1$	$[N/2]$
$[N/2]+1$	$N/2+1-0.5$	$[N/2]+1-1.5$
$[N/2]+2$	$N/2+2-0.5$	$[N/2]+2-1.5$
$[N/2]+3$	$N/2+3-0.5$	$[N/2]+3-1.5$
⋮	⋮	⋮
N	$N-0.5$	$N-1.5$

Tableau 3.2: Rangs des lettres.

Mot	المحافظة							
Lettres	ة	ظ	ف	ا	ح	م	ل	ا
Poids	5	0	0	5	0	2	1	5
Rangs	7.5	6.5	5.5	4.5	5	6	7	8
Produits	37.5	**0**	**0**	22.5	**0**	12	7	40
Radicaux	حفظ							

Tableau 3.3: Exemple d'utilisation de l'algorithme d'Al-Serhan *et al.*

compliquées si tous les affixes sont utilisés. Par exemple le mot (ليعلّموهم, pour qu'ils les apprennent), construit à partir de la racine (علّم) possède les affixes indiquées sur le tableau 3.4.

Il peut paraître simple de retrouver la racine en procédant par troncature de tous les affixes possibles du mot. Mais cela n'est pas toujours facile du fait qu'une séquence de lettres peut jouer le rôle d'un affixe dans un mot, et d'une partie intégrante dans un autre mot.

Par exemple, les deux mots : *(*مدرسة*), école* et *(*دراسة*, étude)* sont construits à

Antéfixe	préfixe	racine	suffixe	poste-fixe
ل	ي	علّم	و	هم

Tableau 3.4: Un mot avec ses affixes.

partir des mêmes radicaux (د ر س), malgré qu'ils ont des sens différents. Ainsi, réduire deux mots différents au même radical peut engendrer des erreurs dans les systèmes de recherche d'informations. Pour cela, la racinisation, qui consiste à repérer et supprimer les affixes d'un mot, tente d'améliorer la performance des systèmes tout en conservant le sens des mots [Aljlayl and Frieder, 2002, Larkey et al., 2007].

Comme pour l'extraction des radicaux, les techniques de racinisation proposées dans la littérature peuvent être classées en deux catégories, selon qu'elles reposent ou pas sur l'utilisation de ressources linguistiques.

3.2.2.1 Techniques supervisées de racinisation

Dans le contexte de la recherche d'information et à cause de la nature flexionnelle de la langue arabe, il a été constaté que l'extraction des radicaux entraine une dégradation de la précision de ces systèmes. Afin d'y remédier, [Aljlayl and Frieder, 2002] ont proposé un algorithme de racinisation basé sur des règles linguistiques. Cet algorithme consiste à effectuer plusieurs passages du texte, qui tentent de localiser et supprimer les préfixes et suffixes récurrents d'un mot.

D'autres travaux se concentrent aussi sur l'effet de la racinisation sur les systèmes de recherche d'information [Al Ameed et al., 2005],[Harmanani et al., 2006] et [Kanaan et al., 2008]. Ils concluent tous que la racinisation donne de meilleurs résultats par rapport à l'extraction des radicaux dans le contexte de la recherche d'information.

3.2.2.2 Techniques non supervisées de racinisation

Ce type de techniques n'exige pas l'utilisation de ressources linguistiques, seul le texte brute est exploité. L'un des premiers travaux sur la racinisation non supervisée des mots arabes est due à [Larkey et al., 2002]. Les auteurs se sont inspirés d'une approche qui a déjà fait ses preuves à la langue Anglaise. Cette approche repose sur une mesure des co-occurrences. Elle assume que les formes d'un mot, qui devraient être regroupées pour un corpus donné, doivent se présenter dans les mêmes documents à partir de ce corpus [Xu and Croft, 1998]. Les résultats ont montré que l'approche de co-occurrences seule n'est pas autant efficace pour l'arabe que pour l'anglais et l'espagnol. Dans [Rogati et al., 2003], un racineur d'anglais et un corpus parallèle ont été utilisés pour construire un racineur arabe non supervisé. Plus récemment, les auteurs de [Alajmi

et al., 2011] ont proposé une technique, basée sur les modèles cachés de Markov, qui extrait les modèles des mots en dépouillant des préfixes et suffixes à partir d'un mot donné. [Khorsi, 2012] a proposé une approche de racinisation des mots pour la langue arabe classique en se basant sur les fréquences des N-grammes.

3.3 État de l'art des techniques de CDA

Plusieurs travaux ont été réalisés en quête de systèmes robustes de CDA. On rappelle qu'un système de classification de documents en arabe garde le même schéma que celui des langues latines et anglophones. La principale différence réside dans la phase de pré-traitement et la spécificité des techniques de racinisation et d'extraction de radicaux utilisées. Nous avons recensé plus de 40 travaux sur la CDA, on a choisi de relater les plus importantes contributions.

Nous avons montré sur le tableau 3.5 les critères suivants :

1. Techniques utilisées dans la phase de pré-traitement, la racinisation ainsi que la réduction de dimension.
2. Modèle de représentation des documents : en effet, les documents sont souvent réduits en une représentation facile à manipuler (le vecteur).
3. Algorithme d'apprentissage du classificateur : nous pouvons classé les travaux selon la nature et le type de l'algorithme utilisé dans la phase d'apprentissage (Figure 3.2).
4. Type du corpus utilisé : le manque des ressources a fait que les travaux de CDA ont été souvent évalués sur des ensembles de documents collectés par les chercheurs ou "fait maison". Peu de travaux ont été évalués sur des corpus standards, mis à la disposition des autres chercheurs.

Nous avons opté pour une présentation des contributions par ordre chronologique. Cette chronologie des travaux a fait déceler quatre périodes regroupant des travaux ayant des caractéristiques communes. Il faut aussi mentionner que l'on va omettre d'analyser les résultats vu que la plupart des travaux n'utilisent pas des corpus standards, pour lesquels les résultats seront comparables.

Tableau 3.5: Résumé des travaux sur la CDA.

Référence	Corpus				Validation	Pré-traitement	Stemming	Pondération	Sélection	Algorithme
	Appellation	Type	# docs	# cats	croisée				des attributs	de classification
[Sawaf et al., 2001]	Newswire	Collection	33k	34	Non	Aucun	Non	TF	Aléatoire	entropie maximale
[El Kourdi et al., 2004]	Aljazeera News	Collection	1k	10	Oui	Mots outils, Voyelles	Extraction de radicaux	TF-IDF (2000)	-	NB
[Duwairi, 2006]		Collection	1k	10	Non	Mots outils, Ponctuations	Extraction de radicaux	-	-	Distance-based, Dice measure
[Al-Shalabi et al., 2006]	-	Collection	621	6	Non	Mots outils	Racinisation	TF-IDF		Cosine
[Syiam et al., 2006]	-	Collection	1,1k	6	Non	Mots outils	Racinisation	Seuillage de fréquence des documents	IG	KNN, Rocchio
[Moh'd A Mesleh, 2007]	-	Collection	1,4k	9	Non	Ponctuations, Mots outils	Non	TF-IDF	Chi2	SVM, NB, KNN
[Duwairi, 2007]	-	Collection	1k	10	Non	Ponctuations, Mots outils, prépositions, pronoms, Conjonctions, Auxiliaires	Extraction de radicaux	-	-	NB, KNN, Distance Based
[El-Halees, 2007]	-	Collection	-	6	Non	Mots outils	Extraction de radicaux	-	-	entropie maximale
[Al-Harbi et al., 2008]	-	Collection	17,6k		Non	Mots outils	Non	Binaire	Chi2	SVM, C5.0

Suite dans la page suivante...

Résumé des travaux sur la CDA (suite).

Référence	Corpus				Validation croisée	Pré-traitement	Stemming	Pondération	Sélection des attributs	Algorithme de classification
	Appellation	Type	# docs	# cats						
[Bawaneh et al., 2008]	-	Collection	242	6	Oui	Mots outils	Racinisation	Non	-	KNN, NB
[Thabtah et al., 2009]	SPA	Standard	1562	6	Non	Ponctuations, Mots-outils, Noms propres	-	-	Chi2	NB
[Thabtah et al., 2009]	SPA	Standard	1,5k	6	Non	Ponctuations, Mots outils, Noms propres	-	-	Chi2	NB
[Gharib et al., 2009]		Collection	1,1k	6	Oui ($k = n$)	Mots outils	Racinisation	TF-IDF	IG	SVM, KNN, Rocchio, NB
[Khreisat, 2009]		Collection	-	4		Ponctuations, Mots outils, Noms propres	Non	Fréquence	-	Distance-based : Dice , Manhattan
[Harrag and El-Qawasmah, 2009]	Hadith	spécifique	453	14		Ponctuations, Mots outils	Racinisation	SVD (Décomposition en valeurs singulières)		RNA
[Harrag et al., 2009]	-	Collection	373 453	8 14	Non	Ponctuations, Mots outils	Extraction de radicaux	TF, DF, TF-IDF	IG	Arbres de décision
[Saad and Ashour, 2010]		Collection	119	3	10-folds	Ponctuations, Mots outils, Noms propres		Binaire, TF, DF, TF-IDF	-	C4,5 arbres de décision

Suite dans la page suivante...

Résumé des travaux sur la CDA (suite).

Référence	Corpus				Validation	Pré-traitement	Stemming	Pondération	Sélection	Algorithme
	Appellation	Type	# docs	# cats	croisée				des attributs	de classification
[Alsaleem, 2011]	SNP	Standard	5,1k	7	10-folds	Ponctuations, Mots outils, Noms propres	-	-	-	SVM, NB
[Al-Salemi and Aziz, 2011]	-	Collection	3,1k	4	Non	Ponctuations, Mots outils, Noms propres	Racinisation	TF	Chi2, GSS, MI et OR	NB
[Al-diabat, 2012]	SPA	Standard	1,5k	6	10-folds	Ponctuations, Mots outils, Noms propres	Non	TF	Chi2	Arbres de décision, règles d'induction
[Belkebir and Guessoum, 2013]	OSAC	Standard	22,4k	10	Non	Ponctuations, Mots outils, Noms propres	Extraction de radicaux/Racinisation		BSO-Chi2	SVM
[Sharef et al., 2014]	-	collection	3,1k	4	Non	Ponctuations, Mots outils, Noms propres	Racinisation	TF		FRAM
[Al-Tahrawi and Al-Khatib, 2015]	Aljazeera News Arabic Dataset	Standard	1,5k	5	Non	Ponctuations, Mots-outils, Noms propres	Extraction de radicaux	TF	Chi2	réseaux de neurones

Légende :
Type du corpus :
— Collection : ensemble de documents collectés par les auteurs (fait-maison).

— Standard : ensemble de documents structurés et mis à la portée des chercheurs.
— Spécifique : ensemble de documents collectés pour une utilisation particulière, par exemple un ensemble de documents contenant des textes religieux.

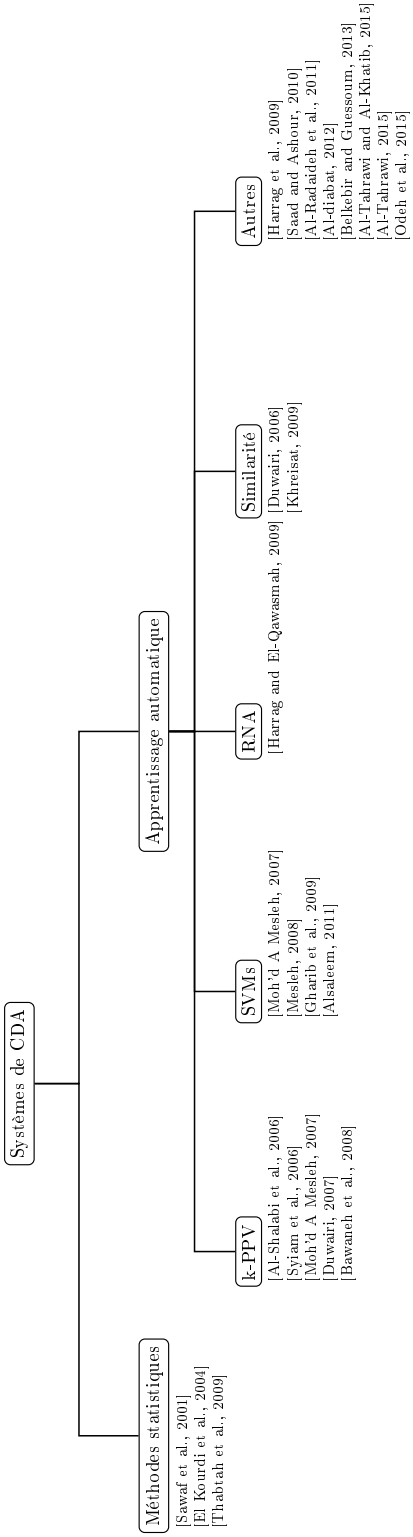


FIGURE 3.2: Taxonomie des systèmes de CDA selon la nature de l'algorithme utilisé.

3.3.1 Période 2001–2004

Les premiers travaux réalisés sur la CDA ont été marqués par l'utilisation des méthodes statistiques. Le recourt à ces dernières est motivé par leur succès pour les langues latines et anglophones. Dans [Sawaf et al., 2001], les auteurs ont traité le problème de CDA indépendamment de la langue arabe. Des méthodes statistiques (l'entropie maximale et l'information mutuelle) ont été appliquées pour la classification et la catégorisation des documents du volume de l'année 1994 du corpus Newswire (LDC). Aucun pré-traitement n'a été réalisé. Les résultats de la classification, pour quatre classes, étaient moyens (précision = 50% et rappel = 84.2% pour un meilleur F1 = 62.7%). Dans [El Kourdi et al., 2004], l'algorithme de Naive Bayes a été utilisé après avoir fait un pré-traitement sur les documents (élimination des mots outils et extraction des radicaux). L'apprentissage a été réalisé sur 5 classes de 300 documents chacune. La représentation des documents utilise la fréquence inversée TF-IDF (voir section 1.2.3.1). La validation croisée, pour plusieurs valeurs de TF-IDF, donne de meilleurs résultats en terme de précision (68.78%) pour 2000 caractéristiques.

3.3.2 Période 2005–2006

L'approche basée sur l'apprentissage commence à être adoptée pour la CDA avec les travaux de [Duwairi, 2006], [Al-Shalabi et al., 2006] et [Syiam et al., 2006]. Dans [Duwairi, 2006], un algorithme de classification basé sur la similarité est utilisé. L'algorithme des K-PPV est utilisé dans [Al-Shalabi et al., 2006] et [Syiam et al., 2006]. Ces travaux ont tenté d'appliquer les algorithmes conventionnels d'apprentissage pour les documents en arabe. A part la racinisation, aucun traitement spécifique à la langue arabe n'a été appliqué. [Syiam et al., 2006] ont testé plusieurs techniques de racinisation, sélection et pondération d'attributs et algorithmes de classification. Ils ont recommandé la combinaison suivante pour la CDA :

- La racinisation en utilisant une approche statistique (N-grammes).
- Sélection de caractéristiques basée sur le gain d'information et un seuil de la fréquence des termes dans les documents.
- L'algorithme Rocchio pour la classification.

Il est à noter que tous les travaux précédents ont été menés sur des corpus recueillies manuellement, de taille variant de 621 à 1132 documents, répartis sur 6 ou 10 catégories. A l'exception de [El Kourdi et al., 2004], aucun de ces travaux n'a réalisé une validation croisée.

3.3.3 Période 2007-2011

Plusieurs algorithmes conventionnels ont été aussi appliqués au problème de CDA :

- Les SVMs dans : [Moh'd A Mesleh, 2007], [Mesleh, 2008], [Gharib et al., 2009], [Alsaleem, 2011] et [Mesleh, 2011].
- Les k-PPV ([Al-Shalabi et al., 2006], [Syiam et al., 2006], [Moh'd A Mesleh, 2007], [Duwairi, 2007] et [Bawaneh et al., 2008]).
- Les arbres de décision ([Saad and Ashour, 2010] et [Harrag et al., 2009]).
- Les règles d'association ([Al-Radaideh et al., 2011]).
- Les algorithmes basés sur la distance ([Khreisat, 2009]) et les réseaux de neurones ([Harrag and El-Qawasmah, 2009]).

Cependant, la richesse de la langue arabe et la taille importante de son lexique, font que les chercheurs ont remarqué la nécessité de réduire la taille des vecteurs de représentation. Ceci a été réalisé de deux manières :

(i) Sélection ou extraction des caractéristiques les plus pertinentes pour les classes de documents.

(ii) Réduction de dimension.

Plusieurs travaux sont entrepris pour mesurer l'effet des techniques de sélection d'attributs sur les performances de la CDA ([Moh'd A Mesleh, 2007], [Mesleh, 2008], [Al-Harbi et al., 2008], [Thabtah et al., 2009], [Raheel and Dichy, 2010] et [Mesleh, 2011]). Dans [Mesleh, 2008] une évaluation des SVMs avec cinq techniques de sélection d'attributs a montré la supériorité de la technique Chi 2. [Thabtah et al., 2009] recommandent l'utilisation de 800 termes choisis par Chi 2 avec l'algorithme de Naive Bayes. Le travail mené dans [Harrag et al., 2010] sur l'effet de la réduction de dimension a montré que la technique TF-IDF est plus performante que la DF et LSA (Latent Semantic Analysis).

Pour la racinisation et l'extraction des radicaux, qui sont considérées comme des techniques de réduction de dimension, leur effet sur la CDA a fait l'objet des travaux : [Duwairi et al., 2009] ,[Harrag et al., 2010] et [Harrag et al., 2011]. Dans l'étude menée par [Duwairi et al., 2009], trois techniques de réduction de dimension (à savoir la racinisation, l'extraction des radicaux et la segmentation des mots) ont été expérimentées avec l'algorithme des K-PPV. Le corpus utilisé contient 15 000 documents répartis sur trois catégories. En terme d'exactitude, la racinisation a donné le meilleur résultat. Le même résultat a été confirmé dans [Harrag et al., 2011], qui ont comparé eux aussi trois techniques de réduction de dimension (la racinisation, l'extraction des radicaux et la racinisation à base de dictionnaire), mais cette fois avec les SVMs et les réseaux de neurones.

Cette période a été aussi marquée par l'utilisation de corpus recueillis manuellement, à l'exception de [Thabtah et al., 2009] et [Alsaleem, 2011], qui ont utilisé les corpus SPA (Saudi Press Agency) et SNP (Saudi Newspapers) [Althubaity et al., 2008].

La validation croisée commence à être utilisée dans les travaux réalisés à partir de 2010 dans [Saad and Ashour, 2010], [Alsaleem, 2011], qui ont opté pour une valeur $k = 10$.

3.3.4 Période 2012–2015

Dans les travaux précédents, les N-grammes et les termes ont été utilisés comme entrées du vecteur de représentation des documents. Si cette approche permet de capturer l'information du niveau lexical de la langue arabe, il en est pas de même pour le niveau sémantique. En effet, l'utilisation du modèle vectoriel indexé par des termes (mots brutes, racines ou radicaux) ne permet pas de capturer des relations sémantiques importantes entre les termes et traite les mots synonymes indépendamment. De plus, lorsque l'extraction des radicaux est utilisée, ceci peut entrainer une confusion des mots de sens différents lorsqu'ils sont originaires des mêmes radicaux. Afin de remédier à ce problème, plusieurs travaux ont proposé de capturer la similarité entre les documents au niveau sémantique : [Alahmadi et al., 2014b], [Alahmadi et al., 2014a], [Yahya and Salhi, 2014] et [Yousif et al., 2015]. Par exemple, dans [Alahmadi et al., 2014b], les auteurs utilisent les concepts pour comparer deux documents. Le modèle proposé (nommé sac-de-concepts) est basé sur l'utilisation de Wikipedia arabe comme base de connaissances pour la détection des concepts. Ils ont combiné, ensuite ce modèle avec celui du sac-de-mots [Alahmadi et al., 2014a].

Dans cette période on constate que le modèle vectoriel est toujours maintenu pour représenter les documents. Par contre, la réduction de dimension a motivé plusieurs travaux, notamment ceux qui portent sur la sélection des caractéristiques [Haralambous et al., 2014], [Al-Thubaity et al., 2015], [Gadri and Moussaoui, 2015], [Alhutaish and Omar, 2015].

En plus de revisiter certaines algorithmes conventionnels de classification [Alhutaish and Omar, 2015], le problème de la CDA a motivé aussi l'investigation d'autres techniques de classification, telles que la fouille des règles [Al-diabat, 2012], réseaux polynomiaux [Al-Tahrawi and Al-Khatib, 2015], la régression logistique [Al-Tahrawi,], l'évaluation de vecteurs [Odeh et al.,] et de faire recours à l'hybridation des techniques d'apprentissage conventionnelles avec des techniques d'optimisation [Belkebir and Guessoum, 2013].

3.4 Discussion

La synthèse des travaux de CDA proposés dans la littérature, nous a permis d'émettre les constats suivants :

1. Le modèle vectoriel a été adopté dans tous les travaux pour représenter les documents. Ceci est expliqué par le fait que la plupart des algorithmes conventionnels de classification acceptent en entrée des données structurées en vecteurs. Cette transformation des données textuelles non structurées vers des données vectorielles structurées induit une perte d'information. Par exemple, l'ordre des mots dans une phrase, la co-occurrence des termes et la succession des phrases sont perdues.
2. La richesse lexicale de la langue arabe, menant à des vecteurs de représentation énormes, a impliqué l'utilisation de méthodes de réduction de dimensions.
3. Les travaux sur la CDA utilisent tantôt l'extraction des radicaux, tantôt la racinisation. A part dans le travail de [Belkebir and Guessoum, 2013], l'effet de l'une ou l'autre sur ces systèmes n'a pas été largement étudié. Ce qui n'est pas le cas avec la recherche d'information, pour laquelle il a été établit que la racinisation a un effet positif [Aljlayl and Frieder, 2002].
4. L'utilisation des N-grammes au niveau caractère, a fait l'objet de plusieurs travaux. Par contre, à part le travail du à [Al-Thubaity et al., 2015], les N-grammes au niveau mot n'ont pas été bien exploré. L'effet de la taille des N-grammes et de l'insertion des trous[3], sur la CDA, n'a fait objet d'aucune recherche.
5. La majorité des systèmes de CDA proposés n'ont pas procédé à une validation croisée. Ceci est peut être expliqué par le temps important consommé pour l'apprentissage.

3. C'est des N-grammes avec symboles non contigus.

Chapitre 4

Approche pour l'extraction des racines

Dans ce chapitre, nous proposons une nouvelle technique pour l'extraction de racines des mots en arabe. L'approche proposée fait appel aux transducteurs pour modéliser les préfixes, les motifs des mots et les suffixes. Elle permet d'extraire la racine d'un mot en deux phases : (i) on commence, d'abord, par générer les racines potentielles du mot, c'est-à-dire toutes les racines qui répondent aux formes préfixe-modèle-suffixe modélisées ; (ii) puis, on choisit la racine la plus probable, en se basant sur une étude statistique des racines.

Nous commençons ce chapitre par expliquer comment déployer les transducteurs pour représenter les modèles de mots. Puis, nous expliquons l'utilisation de l'opération de composition entre transducteurs pour extraire une racine. Ensuite, nous montrons les étapes nécessaires pour la construction du racineur. Nous mettons l'accent en particulier sur l'aspect non déterministe de notre racineur et comment résoudre ce problème. Une étude expérimentale pour l'évaluation de l'approche proposée est présentée dans la deuxième partie de ce chapitre.

4.1 Extraction de racines par les transducteurs

4.1.1 Modèles de mots en arabe

L'arabe, une langue sémitique, est différente des autres langues de nombreux aspects. Ces aspects sont d'ordre syntaxique, morphologique et sémantique. Elle est une langue très flexionnelle. En effet, l'une de ces principales propriétés est que les mots, pour la plupart, sont créés à partir de radicaux en suivant certains modèles et en ajoutant des

préfixes et des suffixes. Par exemple, le mot arabe الشراكة (Partenariat) est construit à partir des trois lettres radicaux (ش ر ك) en utilisant le motif فعال, puis le préfixe ال et le suffixe ة (qui est utilisé pour désigner le sexe féminin) sont ajoutés. Il en résulte la forme الفعالة. Le tableau 4.1 illustre des modèles ainsi que des exemples de mots associés. Notez ici que la lettre ف représente la première lettre de la racine à trois lettres, ع désigne la seconde lettre et ل désigne la troisième.

Tableau 4.1: Exemples de formes pour la racine à 3 lettres ش ر ك et les mots associés.

Formes	مفاعلة	فاعل	الفعالة	يفاعل	يتفاعل
Mots	مشاركة	شارك	الشراكة	يشارك	يتشارك

Si nous prenons en compte les signes diacritiques, le nombre de modèles (motifs) peut dépasser des centaines [ابن الحسن العلمي، ادريس, 2011]. Comme nous ne considérons pas les diacritiques dans notre travail, le nombre de modèles est considérablement réduit, pas plus de deux cents, dont beaucoup ne sont pas employés par l'arabe moderne standard (MSA). En effet, les motifs (فَعَلَ، فُعَلَ، فُعَلٌ، فَعِلٌ) se traduiront par un seul motif (فعل) après avoir enlevé les diacritiques. Pour illustration, les tableaux 4.2 et 4.3 montrent des exemples de modèles de noms et de verbes de différentes longueurs.

4.1.2 Transducteurs pour les modèles de mot

Nous utilisons les modèles de mots pour construire un transducteur qui fait l'extraction des trois radicaux d'un mot. La figure 4.1 illustre un exemple de transducteur pour le modèle فاعل, qui sera utilisé pour obtenir la racine de n'importe quel mot adapté à ce modèle, en appliquant l'opération de composition (équation (2.4)).

Tableau 4.2: Exemples de modèles de noms.

Modèle de nom				
3-lettres	**4-lettres**	**5-lettres**	**6-lettres**	**7-lettres**
فعل	فاعل	مفاعل	متفاعل	استفعال
	فعول	مفتعل	مفعوعل	افعيلال
	مفعل	مفتعل	مستفعل	افتعالة

Tableau 4.3: Exemples de modèles de verbes.

Modèle de verbe					
3-letters	**4-letters**	**3-letters +1**	**3-letters +2**	**3-letters +3**	**4-letters +1**
فعل	فعلل	فاعل	افتعل	استفعل	تفعلل
			انفعل	افعوال	افعنلل
			تفاعل		

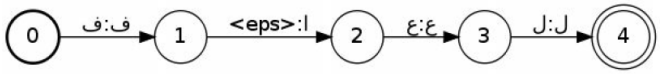


FIGURE 4.1: Exemple d'un transducteur associé au modèle فاعل.

On dénote par T_{modele} le transducteur permettant d'obtenir la racine y d'un mot *terme* en entrée. On dénote par T_{terme} le transducteur qui renvoie une sortie identique à l'entrée, c'est-à-dire le seul chemin possible est celui donné par : $P(\{0\}, terme, terme, \{i\})$. La figure 4.2 montre ce transducteur, associé au mot arabe المدرسة.

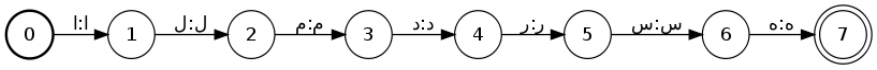


FIGURE 4.2: Transducteur associé au mot المدرسة (école).

La composition de deux transducteurs donne un résultat qui est aussi un transducteur :

$$(T_{terme} \circ T_{modele})(terme, y) \quad = \sum_{z \in \Sigma^*} T_{terme}(terme, z) \cdot T_{modele}(z, y)$$

Étant donné que la seule correspondance possible pour la chaîne z est $z = terme$, nous concluons que :

$$(T_{terme} \circ T_{modele})(terme, y) \quad = T_{terme}(terme, terme) \cdot T_{modele}(terme, y)$$

Nous savons que $T_{terme}(terme, terme) = 1$, donc :

$$(T_{terme} \circ T_{modele})(terme, y) \quad = T_{modele}(terme, y)$$

Si le mot `*terme*´ est adapté au modèle utilisé $T_{modele}$ (c-à-d que le transducteur reconnait le mot `*terme*´), alors le transducteur résultant renvoie la racine y associée à ce mot.

4.1.3 Construction du racineur

Un mot en arabe répond, en général, à une forme préfixes-modèle-suffixes. Pour cela, nous devons aussi modéliser les préfixes et suffixes des noms et des verbes. Il y a 4 préfixes de verbes (ن ا ي ت), 12 préfixes de noms (ا، ال، ب، ت، س، ف، لل، ل، ي،) et plus de 20 suffixes : (و، ن، م تما، كما، ان، ها، وا، تم ، كم ، تن، كن، نا، تا، ما، ون، ين، هن، هم، ته، تي، ني، ن، ك، ه، ة، ت، ا، ات، ي). On construit un transducteur pour chacun de ces groupes (Figures 4.3 et 4.4).

Nous allons suivre les étapes suivantes, pour construire le transducteur d'extraction de racines, ce qui nous permettra de considérer tous les modèles :

1. Construire le transducteur de tous les préfixes de noms (resp. préfixes de verbes) ;
2. Construire le transducteur de tous les modèles de noms (resp. modèles de verbes) ;
3. Construire le transducteur de tous les suffixes de noms (resp. suffixes de verbes) ;
4. Concaténer les transducteurs obtenus dans les étapes 1 et 3 avec le transducteur des noms (resp. le transducteur des verbes) dans l'ordre de leur construction ;
5. Faire la somme des deux transducteurs obtenus de l'étape 4.

Les étapes 1 et 3 sont similaires, un transducteur pour chaque préfixe (resp. suffixe) est construit. Puis, on fait l'union de tous ces transducteurs. Le résultat représente le transducteur des préfixes (resp. suffixes) (voir Figures 4.3 et 4.4). Dans l'étape 2, nous construisons un transducteur par modèle de nom. Les modèles de noms retenus sont ceux les plus utilisés dans le cadre de la arabe standard moderne[1]. Ensuite, l'union de ces transducteurs donne le transducteur de tous les modèles de noms. Le transducteur de tous les modèles de verbes est construit de la même manière (Figure 4.5).

Dans la quatrième étape, les transducteurs, résultants des étapes précédentes, sont concaténés. Le transducteur final est obtenu par l'opération de somme des deux transducteurs à partir de la quatrième étape.

Le transducteur $T_{stemmer}$ résultant est si grand qu'il ne peut pas être représentée graphiquement, il comprend plus de 400 états. Il peut extraire la racine d'un mot arabe grammaticalement correct, c'est-à-dire un mot qui correspond à un certain modèle. Cependant, ce transducteur ne sera pas en mesure de traiter correctement les mots arabes issus d'une racine ayant une consonne faible. Ce genre de mots pourrait être manipulé avec l'utilisation de règles phonologiques, qui ne sont pas pris en charge dans ce travail.

Comment faire face au non déterminisme ?

1. Voir l'annexe A pour une liste complète des modèles utilisés dans le cadre de notre travail.

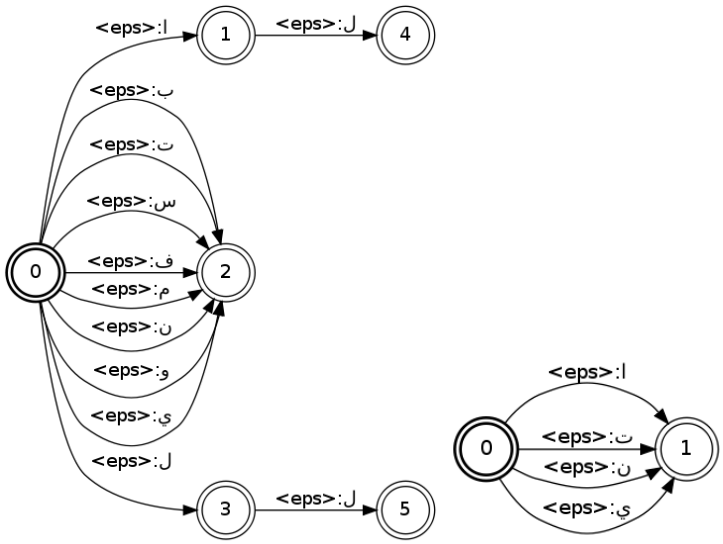


FIGURE 4.3: Transducteurs des préfixes de noms (gauche) et les préfixes des verbes (droite).

Il est important de noter que la composition de $T_{stemmer}$ avec un transducteur d'un mot donné T_{terme} donne lieu à un transducteur pouvant inclure de nombreux chemins, par conséquent, de nombreuses racines possibles. En effet, un mot arabe pourrait correspondre à plus d'un modèle en même temps. Prenons le mot انتصر (gagner). Ce mot correspond, au moins, à deux modèles : انفعل et افتعل donnant les racines تصر et نصر respectivement. Ainsi, l'utilisation de $T_{stemmer}$ conduit à un ensemble contenant une ou plusieurs racines possibles. Cependant, il est assuré que la racine correcte appartient à cet ensemble des racines possibles.

Afin de contourner cette situation d'indéterminisme, le transducteur $T_{stemmer}$ d'extraction nécessite une pondération. Nous avons opté pour l'utilisation des fréquences d'apparition des lettres dans les racines des mots arabes. En effet, nous utilisons une technique basée sur les bi-grammes pour affecter un score à une racine. Cette technique repose sur une étude statistique des fréquences des

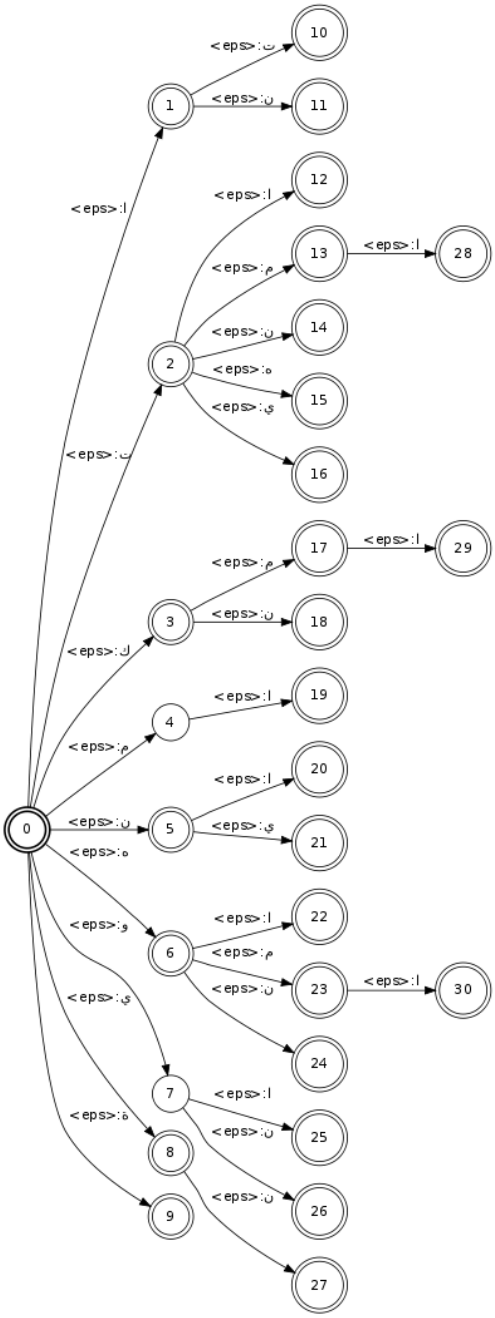


FIGURE 4.4: Transducteur des suffixes de noms et des verbes.

lettres dans le corpus des racines arabes [علي حلمي موسى, 1978]. Ce corpus

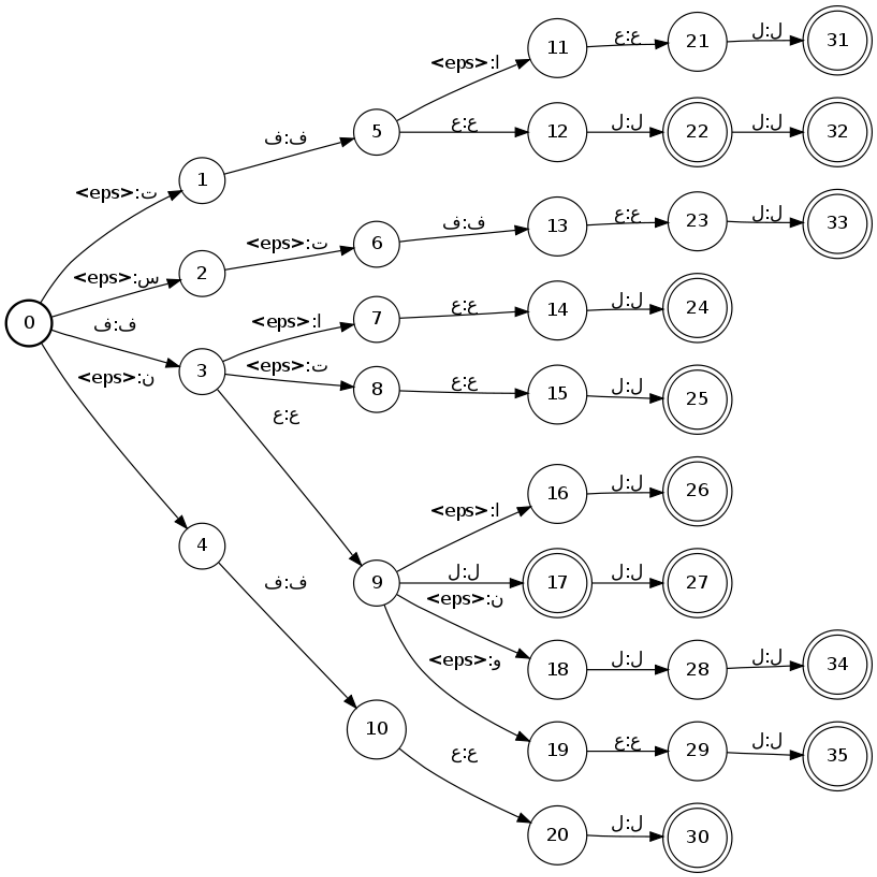


FIGURE 4.5: Transducteur des modèles des verbes.

contient plus de 10 milles racines à trois lettres. Le score affecté à une racine donnée est calculé via les probabilités d'occurrence des lettres dans les différentes positions. En effet, si $s = c_1c_2c_3$ est une racine à trois lettres, alors le score de s, noté $Score(s)$ est calculé par :

$$Score(s) = P_1(c_1, c_2) \times P_2(c_2, c_3) \tag{4.1}$$

où $P_1(c_1, c_2)$ est la probabilité d'avoir la lettre c_2 en deuxième position précédé par la lettre c_1, et $P_2(c_2, c_3)$ est la probabilité d'avoir la lettre c_3 dans la troisième position précédé par c_2. Ainsi, nous considérons la bonne racine est celle qui a le meilleur score s_{best} :

$$best = Arg(Max\{Score(s) \;\; s \in \{\text{racines possibles}\}\}) \tag{4.2}$$

4.2 Expérimentation et résultats

Afin d'évaluer les performances de notre racineur $T_{stemmer}$, nous avons effectué un lot d'expérimentation dont le but est de comparer notre racineur avec deux autres racineurs très cités dans la littérature. Le premier racineur utilise une approche supervisé de racinisation [Khoja and Garside, 1999]. Par contre, le deuxième racineur, du à [Al-Serhan et al., 2003], est non supervisé [2].

4.2.1 Corpus et outils utilisés

L'absence de jeux de mots d'essai, annotés par leurs racines, nous a conduit à annoter un ensemble de mots pour l'évaluation de la performance de notre technique d'extraction des racines. Cet ensemble, contenant 2523 mots, est composé de trois collections. La première collection (Gold1) est un échantillon prélevé du corpus de l'arabe contemporaine [Sawalha, 2011]. Les deux autres collections (Gold2 et Gold3) sont des ensembles recueillis manuellement. Tous les mots de ces trois ensembles ont été annotés à la main avec leur racine adéquate. Les racines ont été confirmés par des experts en langue arabe. Les trois ensembles sont choisis au hasard à partir de différentes catégories, y compris la politique, la culture, le sport et les nouvelles. Le tableau 4.4 donne un aperçu de ces trois collections. Nous donnons pour chaque collection le nombre de mots (# mots).

Tableau 4.4: Détails des collections de mots.

Collection	# **mots**
Gold1	679
Gold2	844
Gold3	1,000
Total	2,523

Concernant les transducteurs, ils sont créés et manipulés à l'aide de la bibliothèque OpenFst [Allauzen et al., 2007], qui est une bibliothèque Open Source pour la construction, la combinaison, l'optimisation et la recherche des transducteurs pondérés.

2. Ces racineurs sont décrit en détail dans le chapitre 3

Les principales commandes de la bibliothèque OpenFst, utilisées pour mettre en œuvre notre outil d'extraction de racines, sont :

```
fstcompile --isymbols=ialphabet.txt --osymbols=oalphabet.txt text.fst binary.fst
fstprint --isymbols=isyms.txt --osymbols=osyms.txt binary.fst text.fst
fstunion modele1.fst modele2.fst resultat.fst
fstconcate modele1.fst modele2.fst resultat.fst
fstcompose word.fst stemmer.fst racine.fst
```

La première commande permet de créer un transducteur binaire (binary.fst) à partir d'un fichier texte (text.fst) qui spécifie la forme du transducteur. Les étiquettes symboliques des alphabets seront converties en nombres entiers en utilisant les fichiers des tables de symboles (données dans ialphabet.txt et oalphabet.txt). La deuxième commande permet d'imprimer un transducteur (binary.fst), en utilisant les fichiers des tables des symboles, dans le fichier texte `text.fst´. Les trois dernières commandes permettent, respectivement, de faire l'union, la concaténation et la composition de deux transducteurs.

4.2.2 Résultats et discussions

Le tableau 4.5 rapporte la performance de notre technique, en terme d'exactitude, sur les trois collections de mots.

Tableau 4.5: Exactitude des différents racineurs.

Collection	Racineur de Khoja %	Notre racineur %	Racineur de Al-Serhan, et al. %
Gold1	82,77	71,68	51,40
Gold2	85,55	74,82	49,64
Gold3	87,60	80,30	56,40
Moyenne	85,30	75,60	52,48

Les résultats montrent l'efficacité de notre approche d'extraction des racines. Les résultats sur les différentes collections sont stables et le meilleur score est obtenu avec la plus grande collection (Gold3).

L'exactitude de notre outil est plus proche à celle du racineur de Khoja que celle d'Al-Serhan. Ceci peut être expliqué par le fait que le racineur de Khoja est un outil basé sur l'utilisation des dictionnaires, ce qui le rend dépendant de la langue. Ainsi, un maintien est nécessaire à chaque mise à jour des dictionnaires. Par contre, l'outil d'Al-serhan utilise peu d'information sur la langue. Cependant, notre approche d'extraction de racine fait appel aux modèles, mais juste en phase de construction du racineur. Vu que les modèles de mots sont figés dans le temps, notre racineur n'a pas besoin d'être mis à jour.

Afin de faire une analyse plus approfondie, nous présentons les résultats des racineurs pour chaque catégorie de mots en fonction de la longueur du mot. La figure 4.6 donne la fréquence des mots pour chaque catégorie dans le corpus global. Il est intéressant de noter que les mots de longueurs $4, 5, 6$ ou 7 représentent 78% de la taille du corpus.

La figure 4.7 montre la précision de la prédiction des trois racineurs pour chaque catégorie de mots. Tout d'abord, nous pouvons remarquer que notre racineur et celui de Khoja surpassent celui d'Al-serhan pour toutes les catégories. Par rapport au racineur de Khoja, les résultats montrent que notre approche donne une meilleure exactitude pour les catégories 3-L et 4-L, elle reste compétitive pour les catégories 5-L, 6-L et 10-L. Pour les catégories 7-L, 8-L et 9-L, le racineur de Khoja est plus performant que le notre. Cela révèle que nous avons besoin d'analyser les résultats erronées issus de notre outil pour ces catégories. Le tableau 4.6 montre des exemples de résultats incorrects, de notre racineur, pour différentes catégories de longueur.

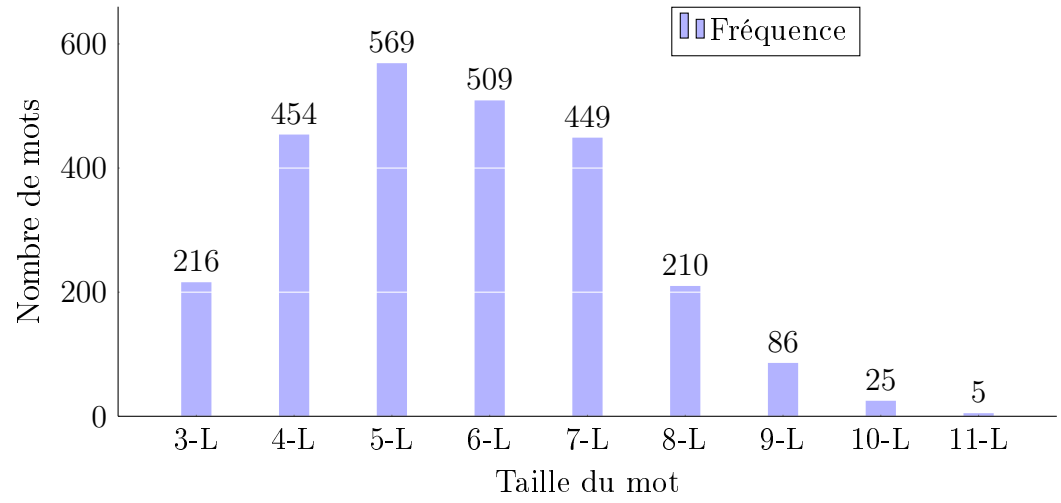


FIGURE 4.6: Fréquences des mots dans la collection, selon la taille du mot.

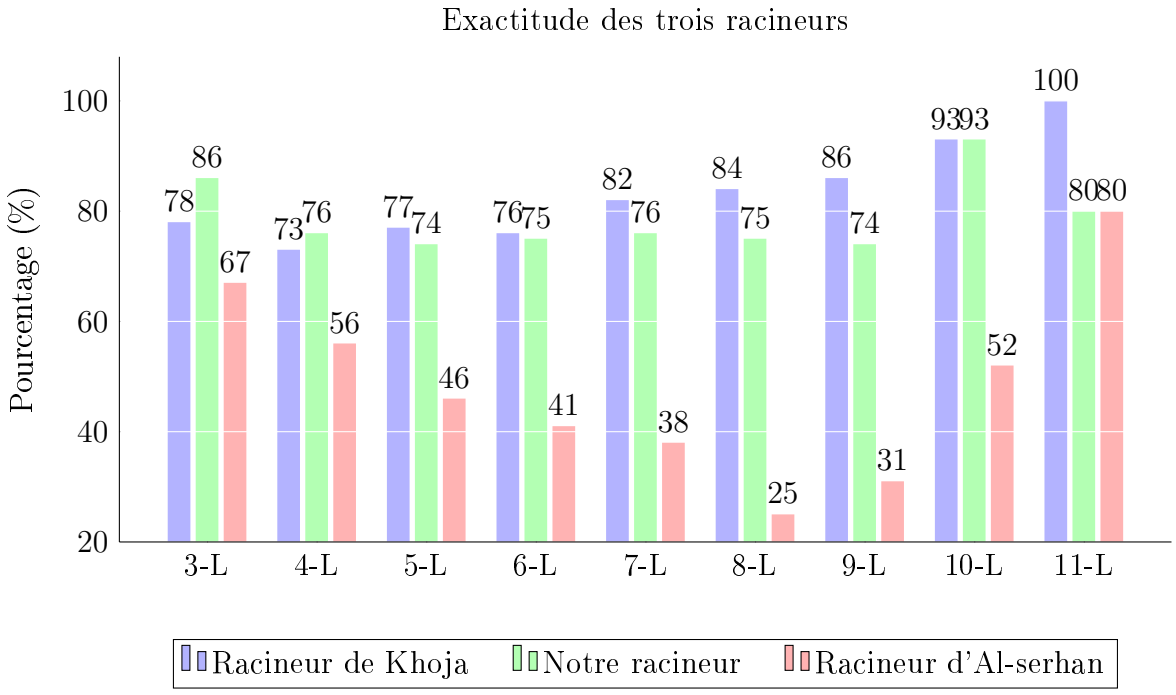


FIGURE 4.7: Exactitude des racineurs par catégorie des tailles de mots.

Les erreurs peuvent être groupées en quatre classes :

1. Considérons les trois premiers exemples dans le tableau 4.6. Ce genre d'erreurs se produisent lorsque l'un des radicaux de la racine correcte est une consonne faible ("ي", "و" ou "ا"). En effet, la racine correcte du mot (Uni, "المتحدة") est ("وحد"), avec un radical faible à la première position ("و "). Les radicaux faibles se transforment des fois en une voyelle, d'autres fois sont supprimés, en fonction de leur environnement vocalique. Il existe plusieurs règles avec des conditions différentes. Ces règles ne sont pas pris en charge dans notre racineur.
2. La deuxième classe d'erreurs est liée aux racines quadrilitères (les racines avec quatre consonnes). Pour les mots (Racial, "العنصري") et (Militaires, "العسكرية"), notre racineur ne parvient pas à obtenir les racines correctes parce que nous n'avons pas considéré les motifs formés à partir des racines quadrilitères.
3. Pour la troisième classe, compte tenu des exemples (Étrangère, "الخارجية"), (Humanité, "الانسانية") et (Avec les abonnements, "بالاشتراكات"), on peut facilement remarquer que notre outil échoue également pour les mots ayant des préfixes/suffixes composés (plus d'un préfixe/suffixe). Dans les premier et second exemples, les suffixes ("ي" et "ة" donnant "ية") sont utilisés.

   Dans le troisième exemple, les préfixes ("ب" et "ال" donnant "بال") sont utilisés. Ceci pourrait être expliqué par le fait que nous faisons composition une seule fois entre les transducteurs des préfixes, des modèles et des suffixes (Voir la section 4.1.3).
4. La dernière classe d'erreurs est liée aux racines avec des radicaux géminées. Les deux derniers exemples dans le tableau 4.6, (La réponse, "الرد") et (Devoirs, "مهام"), illustrent les cas où le dernier radical est supprimé lors de l'utilisation des modèles particuliers. Notre racineur ne pouvait pas extraire les racines correctes dans ce cas.

Tableau 4.6: Exemples de racines incorrectes par notre racineur.

Mot en Arabe (avec la racine correcte)	Nombre de caractères	Notre racineur	Racineur de Khoja	Racineur d'al-Serhan
(Il semble, "يبدو","بدا")	4	بدو	بدا	بدو
(Uni, "وحد","المتحدة")	7	حده	حدد	محد
(Sa démission, "قيل","استقالته")	8	سقل	قول	سقل
(Racial, "عنصر","العنصري")	7	-	عنصر	عنص
(Militaire, "عسكر","العسكرية")	8	-	عسكر	عكر
(Étrangère, "خرج","الخارجية")	8	-	خرج	خرج
(Humanité, "انس","الانسانية")	9	-	انس	نسن
(Avec les abonnements, "شرك","بالاشتراكات")	11	-	شرك	بشر
(La réponse, "ردد","الرد")	4	لرد	ردد	لرد
(Devoirs, "همم","مهام")	4	هام	هوم	هام

4.3 Conclusion

Dans ce chapitre, nous avons introduit une nouvelle technique pour l'extraction des radicaux de mots en arabe. Elle est basée sur l'utilisation des transducteurs pour la modélisation des formes préfixes-motif-suffixes. Nous avons montré aussi comment sélectionner la racine correcte, parmi plusieurs racines candidates, en se basant sur une étude statistique de la co-occurrence des lettres radicaux.

L'analyse des résultats obtenus montre l'efficacité de cette technique comparée à celle d'Al-serhan. Cependant, ces mêmes résultats montrent la supériorité de la technique supervisé de Khoja. Ceci est prévue du fait que notre approche ne fait pas appel à des ressources linguistiques autres que les modèles de mots.

Une analyse profonde des résultats à permis de dégager les faiblesses et lacunes de cette approche.

Chapitre 5

Approche pour classification de documents en arabe

Nous abordons dans ce chapitre notre approche pour la classification de documents en Arabe, qui se base sur l'utilisation des transducteurs pour la représentation des documents, et les noyaux rationnels pour calculer la distance entre ces documents. Cette représentation permet de capturer l'ordre et la co-occurrence des termes. Plusieurs configurations sont possibles selon le choix des paramètres suivants :

- Avec ou sans racinisation.
- La technique de racinisation adoptée.
- La granularité des N-grammes utilisées (niveau caractère ou mot).
- Taille des N-grammes.

5.1 Noyaux rationnels pour la CDA

Selon le niveau retenu de granularité des N-grammes, nous réalisons deux lots d'expérimentation : dans le premier lot, nous considérons les caractères de la langue arabe comme composantes des N-grammes ; dans le deuxième, on considère les termes (mots présents dans le corpus) comme unités des N-grammes.

5.1.1 Niveau caractères

Pour le niveau caractère, le système de classification est construit comme suit :

1. Étape de pré-traitement.
2. Représentation : nous alimentons le transducteur réalisé dans le chapitre précédent ($T_{stemmer}$), par les termes des documents obtenus à partir de

l'étape 1. Ceci va produire un transducteur par mot. Ensuite, nous concaténons ces transducteurs pour donner lieu à un nouvel transducteur, qui représentera le document à l'étape suivante.

3. Étape d'apprentissage : Les noyaux rationnels sont utilisés pour mesurer la distance entre les documents [Cortes et al., 2004, Cortes et al., 2007], et les SVM sont utilisées pour la classification.

Considérons un ensemble de documents $S = \{d_1, d_2, \ldots, d_N\}$. Un document d_i consiste en une séquence de termes : $w_1^i w_2^i \ldots w_m^i$. En appliquant notre racineur sur chaque terme de d_i et puis en concaténant les transducteurs résultants, ceci va transformer ce document en un transducteur T_{d_i}. Les transducteurs obtenus à partir de l'ensemble des documents seront empaquetés dans un fichier d'archive (d'extension .far) pour être traités par l'algorithme d'apprentissage (Figure 5.1).

Les noyaux sur les chaines, qui sont des noyaux définis sur des paires de chaînes, peuvent être étendus aux transducteurs. Ils sont généralement représentés par des transducteurs pondérés, pour mesurer la similarité entre les documents. La figure 5.2 montre un exemple d'un transducteur pondéré $T_{\text{2-grammes}}$, calculant un noyau sur des chaînes. Afin de simplifier l'illustration, nous avons pris des N-grammes avec $N = 2$ et un alphabet $\Sigma = \{a, b\}$.

Soient T_{d_1} et T_{d_2} et T_{d_3} les transducteurs représentants respectivement les trois documents d_1, d_2 et d_3 :

$d_1 =$ "اجتمع وزير الشباب و الرياضة برؤساء الفرق"

$d_2 =$ "اجتماع الوزير بممثلي الفرق الوطنية"

$d_3 =$ "ارتفع الميزان التجاري الوطني خلال السداسي الأول"

La similarité entre les deux documents d_1 et d_2 est calculée sur la base du noyau bi-grammes comme suit :

$$K(d_1, d_2) = \varphi(T_{d_1} \circ T_{\text{2-grammes}} \circ T_{d_2}) \tag{5.1}$$

où φ est une fonction calculant la somme des poids de tous les chemins d'acceptation dans le transducteur $(T_{d_1} \circ T_{\text{2-grammes}} \circ T_{d_2})$, et $\circ$ est l'opération de composition (voir équation 2.4).

5.1.2 Niveau termes

Concernant notre deuxième classificateur, la construction est similaire sauf pour la représentation des documents qui est différente. En effet, l'alphabet cette

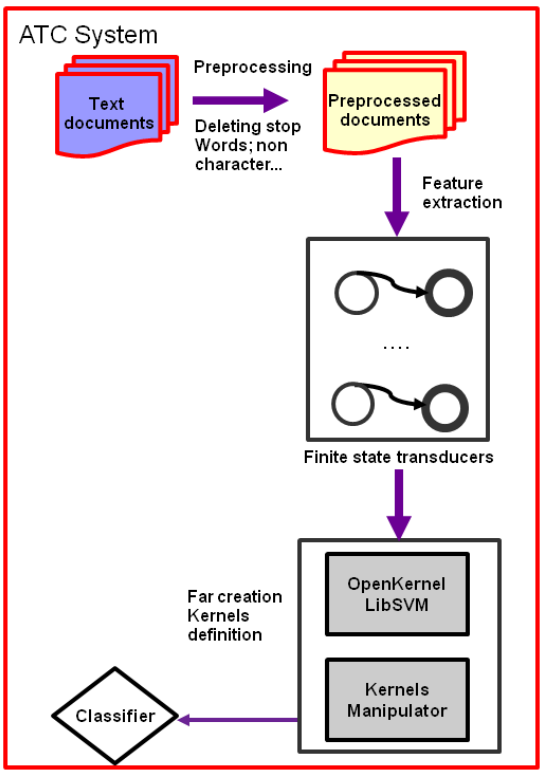


FIGURE 5.1: Composantes du système de CDA.

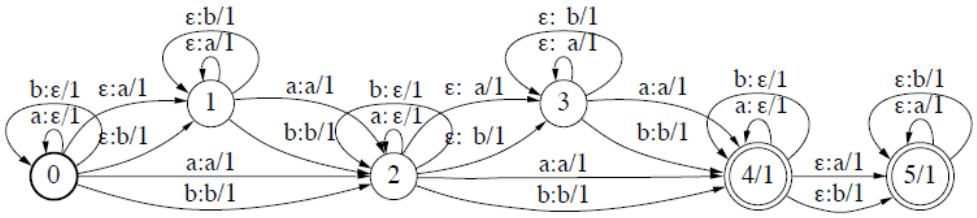


FIGURE 5.2: Noyau bi-grammes pour l'alphabet $\Sigma = \{a, b\}$.

fois-ci n'est plus les lettres de la langue arabe, mais les termes qui apparaissent dans les documents de la collection d'apprentissage et de test. Un document est alors représenté par un transducteur linéaire dont les transitions portent les termes composant le document dans leur ordre d'apparition. La figure 5.3 montre le transducteur linéaire représentant le document d_2 de l'exemple précédent, com-

mençant par le mot (جمع) et se terminant par le mot (وطن). On utilise l'équation 5.1 aussi pour mesurer la similarité entre deux documents avec le noyau bi-grammes, sachant que les bi-grammes cette fois-ci sont des couples de termes figurant l'un à la suite de l'autre, et non pas des caractères.

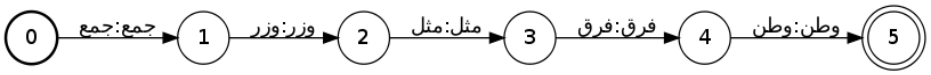


FIGURE 5.3: Transducteur associé à un document en arabe.

Le transducteur résultant de la composition $(T_{d_1} \circ T_{2\text{-grammes}} \circ T_{d_2})$, schématisé dans la figure 5.4, nous montre qu'il existe un seul chemin ayant le couple de termes (جمع :وزر) comme bi-gramme en sortie. Cette paire de termes représente le bi-gramme commun entre les documents d_1 et d_2.

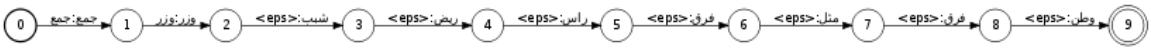


FIGURE 5.4: Transducteur résultant de la composition $(T_{d_1} \circ T_{2\text{-grammes}} \circ T_{d_2})$.

La matrice noyau des bi-grammes, pour les trois documents d_1, d_2 et d_3, est obtenue en calculant le noyau des bi-grammes de chaque paire de document :

$$K = \begin{bmatrix} 1 & 0,2236 & 0,1690 \\ 0,2236 & 1 & 0,1890 \\ 0,1690 & 0,1890 & 1 \end{bmatrix}$$

Sachant qu'un élément $k(d_i, d_j)$ est calculé en utilisant la formule suivante :

$$k(d_i, d_j) = \frac{\#\text{ bi-grammes communs entre } d_i \text{ et } d_j}{\sqrt{\#(\text{ bi-grammes } d_i) \times (\#\text{ bi-grammes dans } d_j)}} \tag{5.2}$$

5.2 Expérimentation et résultats

Afin d'évaluer les performances de notre approche de classification, nous avons effectué une série d'expérimentations, dont les buts sont multiples. Pour le niveau de granularité caractère, nous cherchons à :

— Mesurer l'effet de la racinisation, en comparant le classificateur 1, qui ne procède pas à la racinisation, avec le classificateur 2 implémentant la technique de racinisation d'Al-serhan et le classificateur 3 implémentant notre technique de racinisation.
— Comparer notre technique de racinisation avec celles de Khoja et d'Al-serhan, dans le contexte de la CDA.
— Mesurer l'efficacité des classificateurs en utilisant différentes valeurs pour le noyaux N-grammes ($N = 2, N = 3$ et $N = 4$).

Pour le niveau de granularité terme, nous cherchons à atteindre les objectifs suivants :

— Montrer l'effet de la taille des N-grammes sur les différents classificateurs.
— Montrer l'effet de la racinisation sur la performance du classificateur.
— Mesurer l'impact de la non contiguïté des termes sur les différents classificateurs.
— Evaluer l'effet de la pénalisation des trous introduits dans les N-grammes.

5.2.1 Corpus et outils utilisés

Le manque de corpus standards pour la CDA a fait que plusieurs chercheurs ont utilisé des collections de documents recueillis du Web pour évaluer leurs systèmes de classification (voir tableau 3.5). Les premiers corpus, qui existaient en 2005, et qui sont destinés à la CDA, ont été proposés par le consortium de données linguistiques et par l'agence européenne de distribution de ressources [1]. Ces corpus présentaient l'inconvénient d'être non représentative, du fait qu'ils englobent des documents collectés des agences de presse.

Dans les années qui suivaient, les efforts se sont intensifiés à la quête de corpus standards représentatifs et qui couvrent différents genres de documents.

Parmi les corpus qui ont été développés, celui proposé dans le cadre du projet de la classification de documents en arabe, par la ville du roi Abdulaziz pour la science et la technologie [Althubaity et al., 2008]. Dans ce projet, sept collections ont été assemblées, comprenant 17,658 documents et ayant plus de 11,500,000 mots. Elles couvrent sept genres différents de documents.

Nos expériences sont réalisées sur une de ces collections, issue de l'agence de presse saoudienne (Saudi Press Agency (SPA)) [Althubaity et al., 2008]. Comme indiqué dans le tableau 5.1, cette collection contient 1526 documents appartenant

1. En anglais : Linguistic Data Consortium (LDC) et European Language Ressource Distribution Agency (ELRDA)

à l'une des six catégories : culture, économie, sociologie, politique, généralités et le sport.

Comme pré-traitement, les mots inutiles, lettres non arabes, les symboles et les chiffres ont été supprimés du texte. Pour le niveau caractères, nous avons utilisé un échantillonnage stratifié des documents, avec 80% des documents pour l'apprentissage et 20% pour le test. Par contre, pour le niveau termes, nous avons opté pour une validation croisée avec $k = 10$.

L'apprentissage se fait en utilisant l'outil libsvm [Chang and Lin, 2011], inclus dans la bibliothèque OpenKernel[2].

Puisque nous voulons montrer l'effet de l'extraction des racines, nous présentons les résultats de trois versions du classificateur :

— sans extraction de racine (classificateur 1),
— avec extraction de racine par la technique d'Al-Serhan (classificateur 2),
— avec notre racineur (classificateur 3).

Tableau 5.1: Détails de la collection SPA.

Catégories	**Ensemble d'apprentissage**	**Ensemble de test**	**Total**
Culture	201	57	258
Économie	200	50	250
Sociologie	203	55	258
Politique	200	50	250
Généralités	205	50	255
Sport	205	50	255
	1,214	312	1,526

La bibliothèque OpenKernel, qui est utilisée pour créer, combiner et appliquer des noyaux pour des applications d'apprentissage automatique, permet d'accélérer les expériences. Le script suivant présente les principales commandes de cette bibliothèque, utilisées pour mettre en œuvre notre système de classification.

```
fstcompose mot.fst modele.fst resultat.fst
fstconcate doc.fst result.fst doc.fst
farcreate data.list data.far
klngram -order=3 -sigma=29 data.far 3gram.kar
```

2. Disponible sur le site : http ://www.openkernel.org/

```
5 svm-train -k openkernel -K 3gram.kar cul.train cul.train.3gram.mdl
6 svm-predict cul.test cul.train.3gram.mdl cul.test.3gram.pred
```

Pour extraire la racine de chaque mot dans le document, nous le parcourons en appliquant la commande `fstcompose` (ligne 1) sur chacun des mots, où *mot.fst* est un transducteur linéaire avec entrée et sortie identiques, qui représente le mot, et *modele.fst* représente notre transducteur d'extraction de racines. Le résultat *resultat.fst* représente la meilleure racine du mot. Ces transducteurs sont concaténés à droite au transducteur qui représente le document entier (*doc.fst*) en utilisant la commande `fstconcate` (ligne 2).

L'ensemble des transducteurs obtenus pour tous les documents sont alors empaquetés dans une archive avec la commande `farcreate` (ligne 3), où *data.list* contient la liste de tous les documents, un fichier par ligne, et *data.far* représente l'archive.

De nombreux noyaux peuvent être créés en utilisant la bibliothèque OpenKernel. Les noyaux N-grammes sont créés en utilisant la commande `klngram`. Par exemple, le noyau 3-grammes est crée avec la commande à la ligne 4, dans laquelle on spécifie la taille avec l'argument *–order*, la taille de l'alphabet avec l'argument *–sigma* (l'epsilon n'est pas inclus, l'alphabet est de taille 29). On spécifie aussi l'archive (*data.far*) et le noyau résultant (*3gram.kar*).

La bibliothèque OpenKernel comprend une extension pour la mise en œuvre des SVM (libsvm [Chang and Lin, 2011]). Cela nous permet de faire l'apprentissage, la prédiction et l'évaluation en utilisant notre collection de documents. La commande `svm-train` crée un modèle en utilisant l'ensemble d'apprentissage (ligne 5), où le premier argument -k spécifie le format du noyau, le second (-K) spécifie l'archive du noyau N-gramme. Le premier paramètre spécifie un sous-ensemble correctement classé de l'ensemble d'apprentissage, le second paramètre représente le modèle qui en résulte. Dans cette commande, *cul.train* contient un sous ensemble, correctement étiqueté, des documents d'apprentissage.

Ayant le modèle, nous pouvons l'utiliser pour classer les documents de l'ensemble de test avec la commande `svm-predict` (ligne 6), où le premier paramètre spécifie un sous-ensemble de test correctement classé, le second paramètre représente le modèle résultant de la commande précédente. Le dernier paramètre est le résultat de prédiction en utilisant ce modèle.

5.2.2 Résultats et discussions

5.2.2.1 Niveau caractères pour les N-grammes

Les figures 5.5, 5.6, 5.7 et 5.8 montrent schématiquement les performances en termes de la moyenne d'exactitude, de précision, de rappel et de F1 respectivement, pour les différents classificateurs et pour les trois noyaux N-grammes (n=2,3,4). Les figures 5.9, 5.11 et 5.13, représentent plus de détails sur les performances de ces classificateurs, par catégorie de document, en termes d'exactitude et de précision. Les figures 5.10, 5.12 et 5.14 représentent les mêmes résultats, mais en termes de rappel et F1.

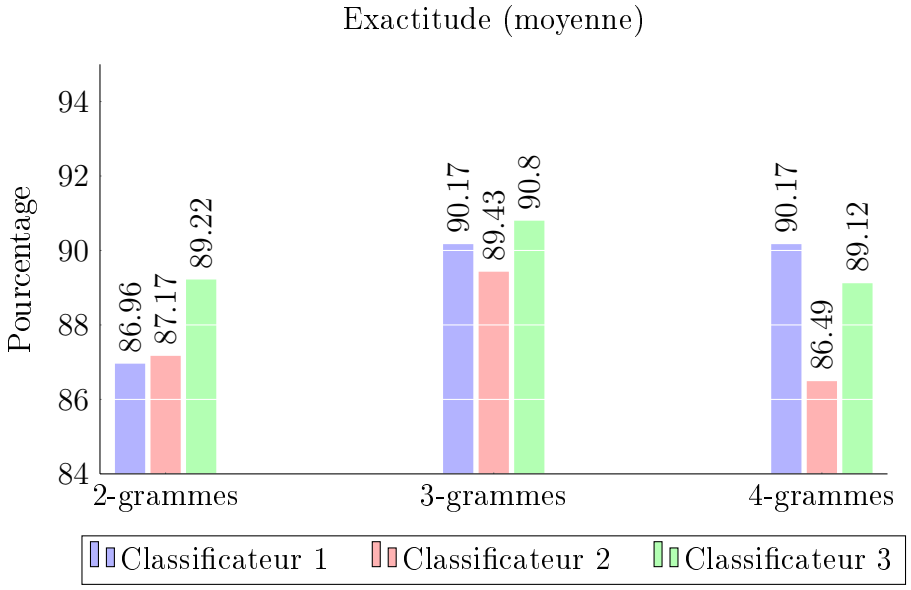


FIGURE 5.5: Moyenne de l'exactitude des différents classificateurs.

Discussion

Notre interprétation portera sur les valeurs moyennes de ces performances. En effet, pour la qualité de la classification, les figures 5.5, 5.6, 5.7 et 5.8 montrent que les meilleurs résultats ont été atteints avec le noyau 3-grammes pour les différentes mesures. Ceci peut être expliqué par le fait que plus de 80% des mots arabes sont à base de racines à 3 lettres.

Pour une analyse plus fine, nous considérons le noyau 3-grammes. En effet, nous mesurons l'effet de l'extraction des racines sur la classification par catégorie

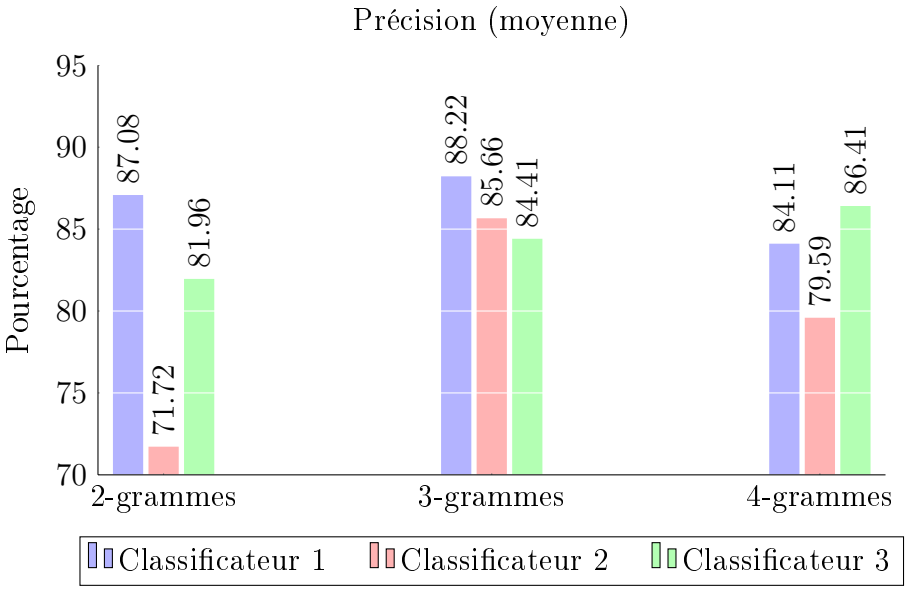


FIGURE 5.6: Moyenne de la précision des différents classificateurs.

de documents. On peut remarquer que notre classificateur surpasse les autres classificateurs dans la plupart des cas. Pour la plupart des classes, l'extraction des racines a amélioré les résultats en termes d'exactitude, de rappel et de F1 (voir les figures 5.11 et 5.12). Cependant, pour la précision, on constate que l'extraction des racines a affecté négativement les performances (voir la figure 5.11).

Les meilleurs scores observés sont enregistrés par la classe *sport*. Ceci peut être expliqué par le fait que cette classe utilise un vocabulaire spécifique. Les mauvais résultats rapportés pour la classe *généralités* peuvent être dus aux termes génériques utilisés dans ce genre de documents. On remarque aussi que l'extraction des racines a affecté négativement la précision pour cette classe.

5.2.2.2 Niveau termes pour les N-grammes

Tout d'abord, on commencera par montrer l'effet de la taille des N-grammes sur les différents classificateurs avec et sans extraction de racines (figures 5.15 et 5.16). Puis, on mesure l'impact de la non contiguïté des termes sur les classificateurs, par l'expérimentation de différentes valeurs pour k ; la taille globale des trous, pour les N-grammes $N = 2$ et $N = 3$. Par exemple, pour $N = 3$ et $k = 2$, les sous séquences considérées se composent de trois termes pouvant s'étaler sur 5 termes au maximum du document. Dans cette partie, nous avons testé les valeurs

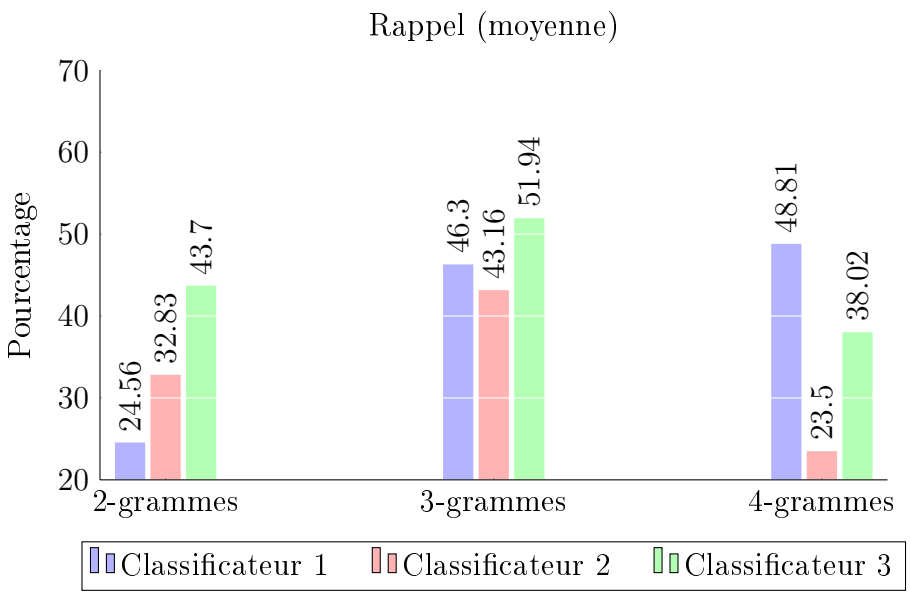


FIGURE 5.7: Moyenne du rappel des différents classificateurs.

$k = 1, \ldots, 5$, en fixant la pénalité λ à 0.1. On s'est basé sur l'idée que les termes s'étalant sur des sous séquences plus longues sont considérés comme des termes non liés. Les résultats obtenus pour les différentes mesures sont donnés dans la figure 5.17.

Enfin, pour évaluer l'effet de la pénalité λ, les valeurs suivantes ont été testées : $\lambda = 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1$ avec les deux N-grammes $N = 2$ et $N = 3$. Les résultats sont présentés dans la figure 5.18.

Discussion

D'une manière générale, on peut remarquer que l'extraction des racines a amélioré légèrement la qualité des classificateurs. On remarque aussi que l'augmentation de N influe négativement les performances pour toutes les mesures, sauf pour la précision, pour laquelle le meilleur résultat a été obtenu pour $N = 3$ en utilisant l'extraction des racines.

Pour le noyau de sous séquences, on peut constater que des sous séquences de termes éloignés ne permettent pas de capturer la similarité entre documents. L'introduction des trous dans les N-grammes n'a pas amélioré les performances. Ceci peut être compris du fait que les termes qui occurrent ensemble se présentent le plus souvent l'un après l'autre.

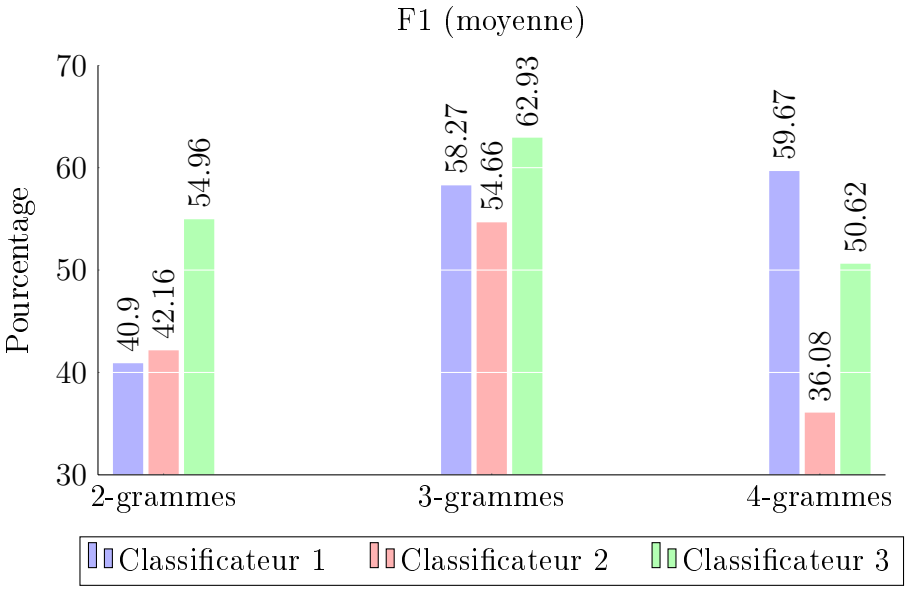


FIGURE 5.8: Moyenne du F1 des différents classificateurs.

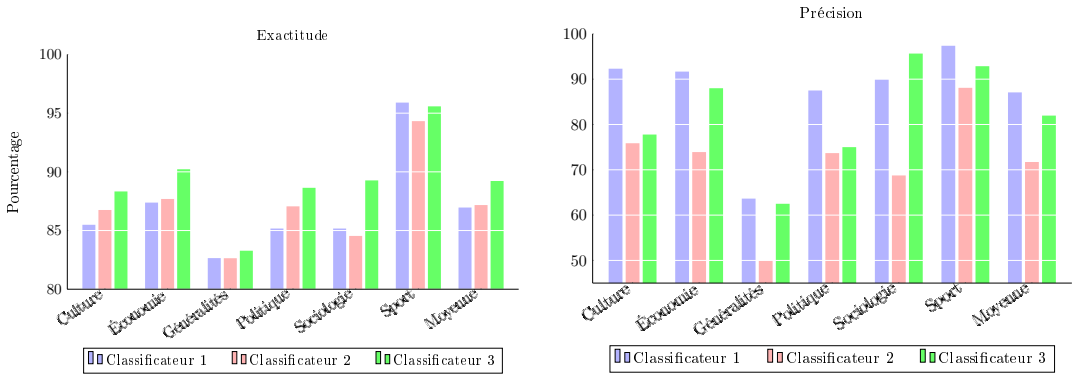


FIGURE 5.9: Exactitude et précision des classificateurs avec le noyau 2-grammes.

Une pénalisation faible ($\lambda = 0.1$) des trous introduits dans le noyau des sous séquences a permis d'atteindre les meilleurs résultats. Ce résultat était prévu, du fait que la pénalisation faible des trous dans les noyaux de sous séquences revient à ignorer ces trous dans le calcul de la distance. En d'autres termes, plus est faible cette pénalisation plus le noyau de sous séquence se rapproche d'un noyau N-grammes.

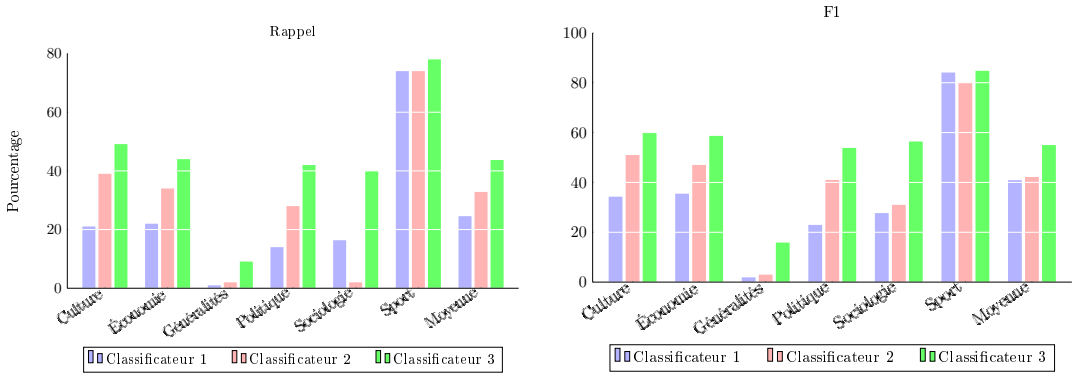


FIGURE 5.10: Rappel et F1 des classificateurs avec le noyau 2-grammes.

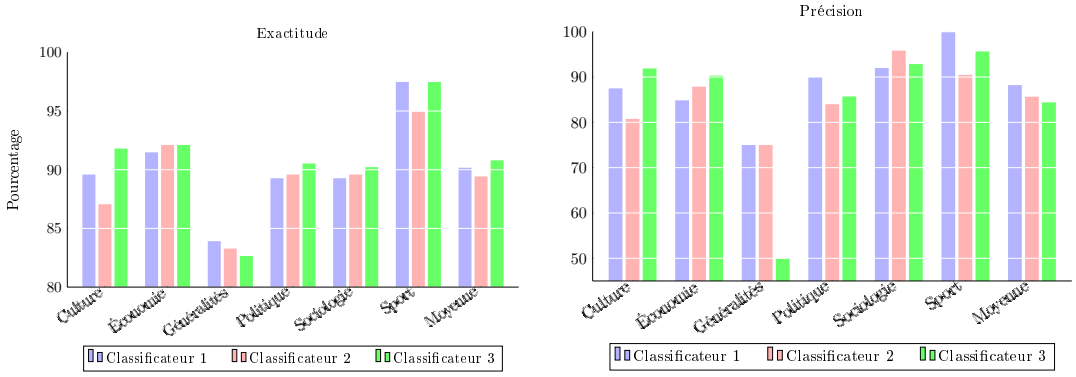


FIGURE 5.11: Exactitude et précision des classificateurs avec le noyau 3-grammes.

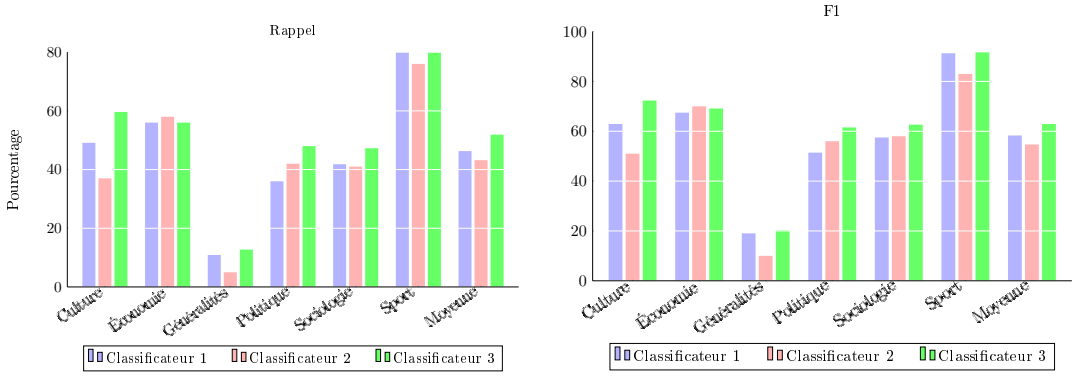


FIGURE 5.12: Rappel et F1 des classificateurs avec le noyau 3-grammes.

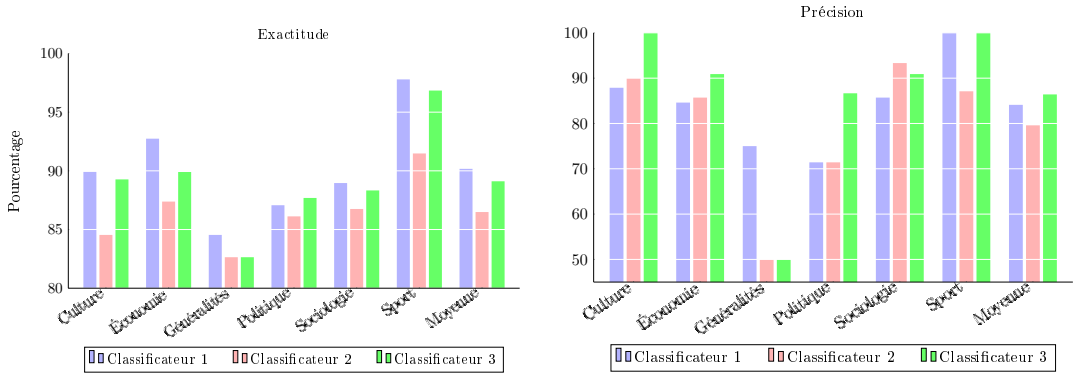


FIGURE 5.13: Exactitude et précision des classificateurs avec le noyau 4-grammes.

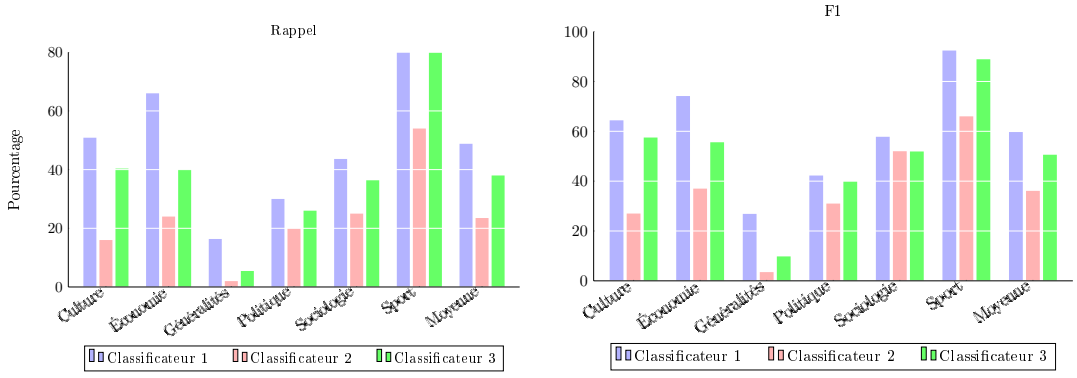


FIGURE 5.14: Rappel et F1 des classificateurs avec le noyau 4-grammes.

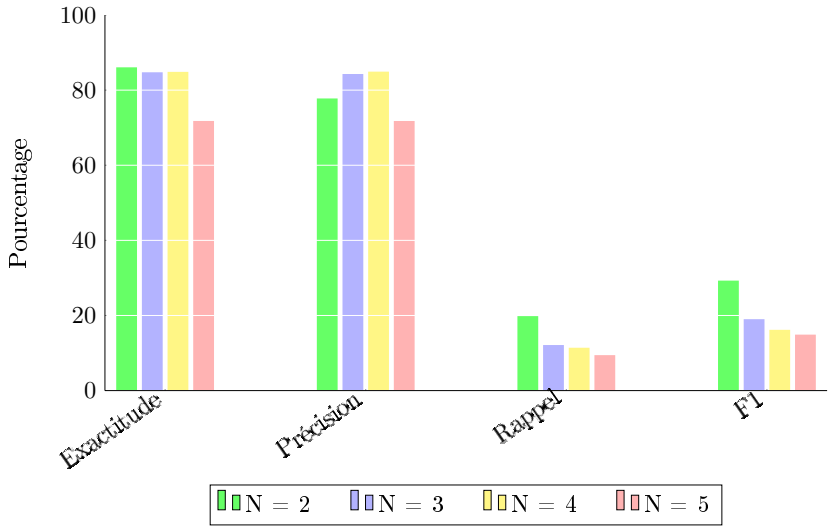


FIGURE 5.15: Effet de la taille des N-grammes (sans racinisation).

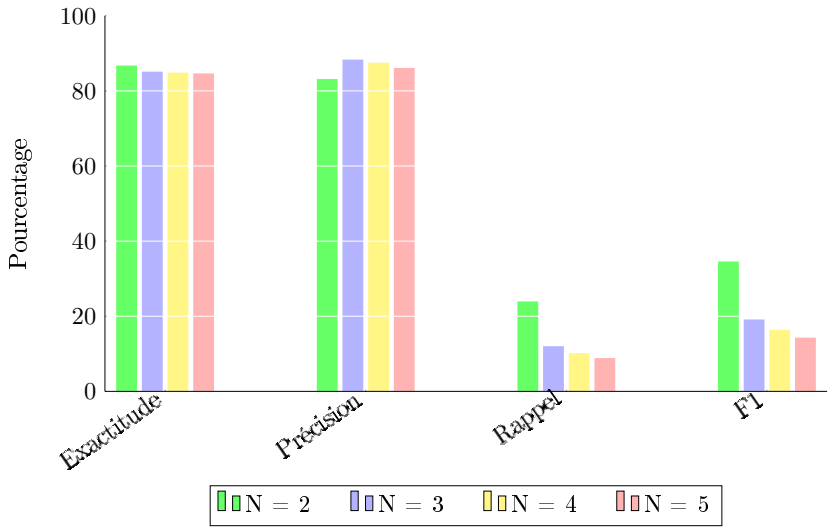


FIGURE 5.16: Effet de la taille des N-grammes (avec racinisation).

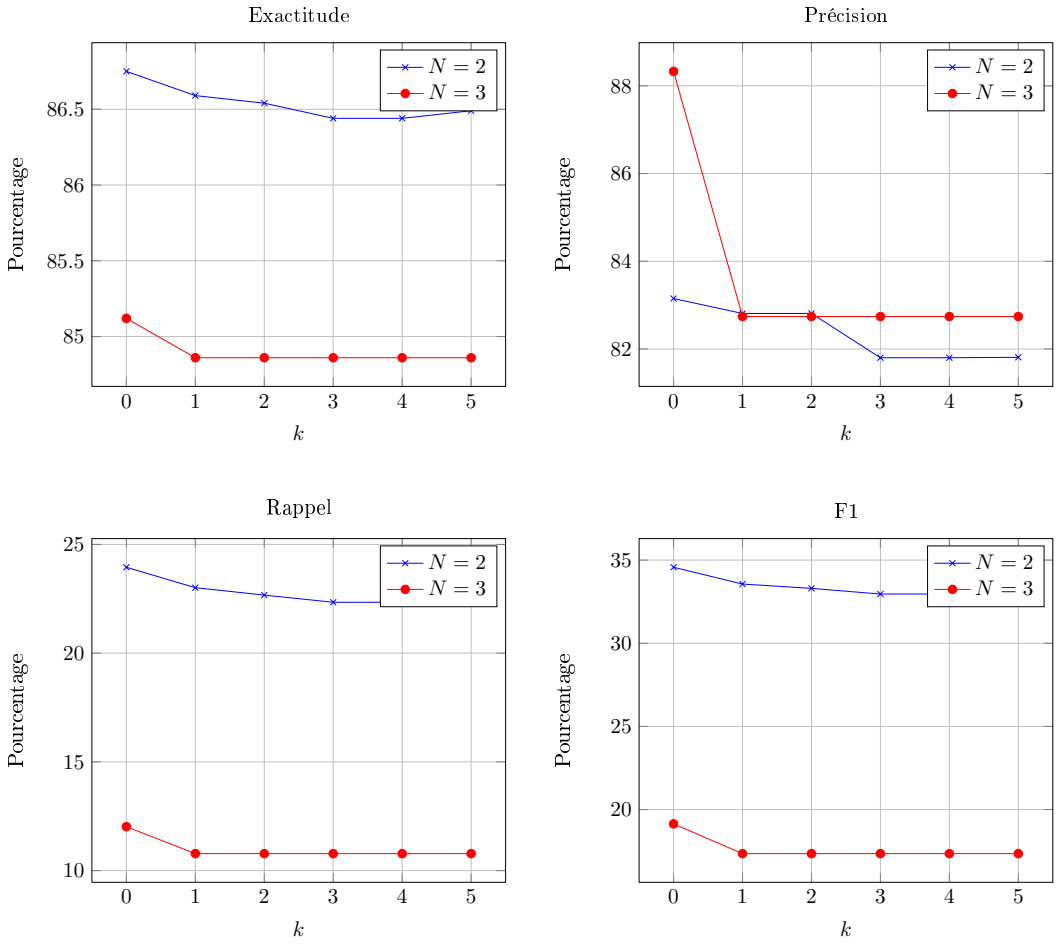


FIGURE 5.17: Effet de la taille du gap k.

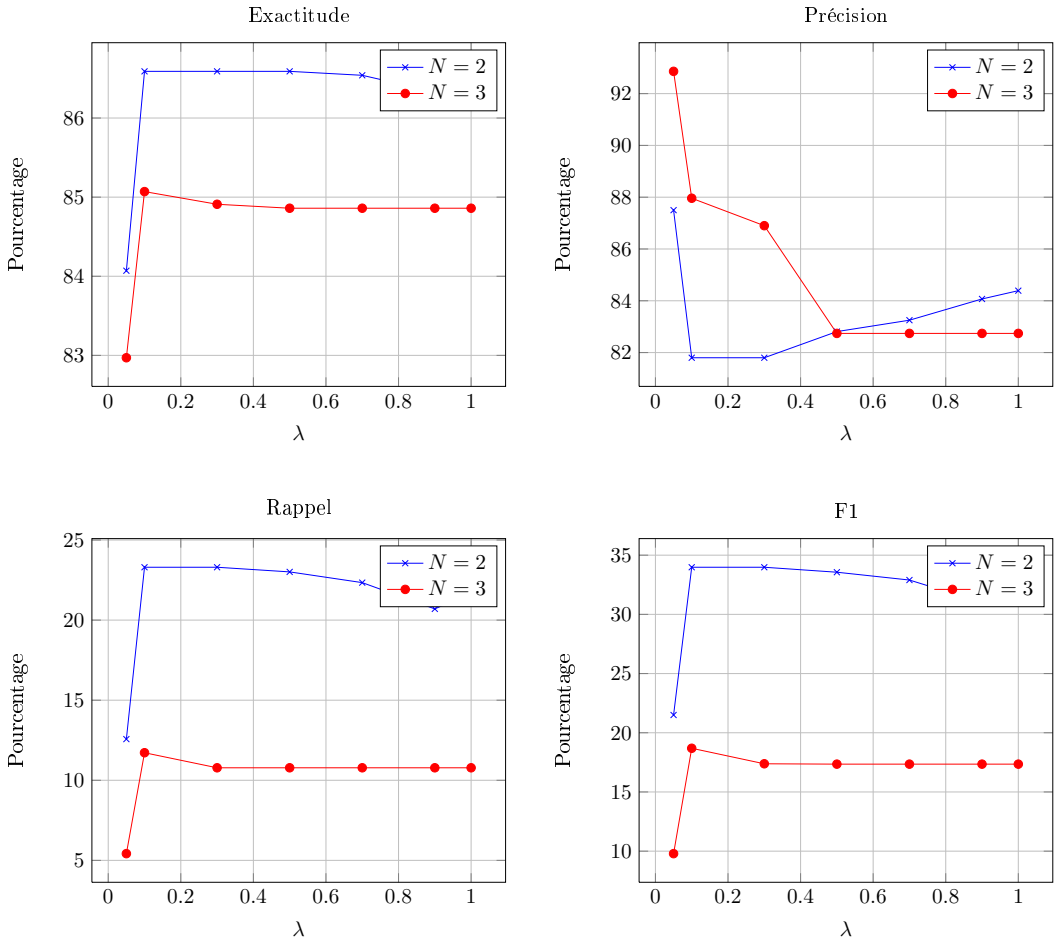


FIGURE 5.18: Effet de la pénalité λ pour un gap $k = 1$.

5.3 Conclusion

Dans ce chapitre, nous avons introduit un nouveau cadre pour la classification de documents en arabe, basé sur l'utilisation des transducteurs pour l'extraction des racines de mots et la représentation des documents, et qui utilise les noyaux rationnels pour mesurer la similarité entre ces documents. Ce cadre nous a permis d'explorer plusieurs noyaux rationnels sur les deux niveaux de granularité des documents.

L'analyse des résultats obtenus montrent que :

— Pour le niveau caractère, le noyau 3-grammes a donné les meilleurs résultats pour les différentes mesures. Mais, pour le niveau terme, l'augmentation de la taille des N-grammes n'a pas amélioré les performances.

— L'extraction des racines a amélioré les performances des classificateurs pour toutes les métriques sauf pour la précision. Ceci nous laisse recommander cette opération pour les applications dans lesquelles le rappel est prioritaire à la précision.

— Le noyau des sous séquences n'a pas amélioré les performances des classificateurs.

Conclusion générale

Introduit depuis plus d'une dizaine d'années, le problème de classification de documents en arabe suscite toujours un grand intérêt compte tenu des différents types d'applications qui peuvent l'utiliser. L'analyse des sentiments et l'analyse des réseaux sociaux, à titre d'exemples, recourent massivement aux systèmes de classification de documents. La nature complexe de la langue arabe fait que, en plus des opérations usuelles de pré-traitement, les systèmes de classification incluent d'autres opérations liées à la langue. L'effet de l'extraction de radicaux, qui a été expérimenté dans le contexte de recherche d'information, n'a pas été évalué pour la CDA.

Dans ce contexte, nous avons introduit un nouveau cadre pour la classification de documents en arabe. Il est basé sur l'utilisation des transducteurs pour la représentation des documents, et les noyaux rationnels pour la mesure de similarité entre ces documents. Dans ce système, les opérations liées à la langue, telles que la racinisation et l'extraction de racines, jouent un rôle primordial. Pour cela, nous avons aussi proposé une nouvelle technique d'extraction de radicaux, basée sur les transducteurs. Ce cadre nous a permis de considérer deux niveaux de granularité avec des tailles différentes pour les N-grammes.

Pour le premier apport, les résultats obtenus ont montré que l'utilisation des transducteurs pour l'extraction de radicaux constitue un choix naturel, vu leur capacité à modéliser la forme flexionnelle des mots en langue arabe. L'efficacité de cette technique a été comparée à deux techniques de référence, l'une supervisée et l'autre non supervisée. Les résultats ont montré que notre approche présente un bon compromis (efficacité/ressources linguistiques requises), comparée aux deux autres techniques. Cependant, ces mêmes résultats ont pu dégager les faiblesses et lacunes de cette approche.

Pour le second apport, les résultats obtenus montrent que l'extraction de racines améliore la qualité des classificateurs en termes d'exactitude, rappel et F1, mais elle diminue légèrement la précision. Les classificateurs basés sur le noyau 3-grammes ont atteint les meilleurs résultats. Pour le niveau N-grammes termes, les résultats ont montré que l'insertion des trous n'améliore pas les performances. Le modèle de sac-à-mots donne de meilleurs résultats par rapport aux modèles expérimentés.

Perspectives

Il ressort principalement de ce travail, que le recours aux transducteurs pour la représentation des documents constitue un choix adéquat dans le contexte de l'analyse de la langue arabe. D'une part, ceci a permis d'utiliser les motifs de mots pour l'extraction des radicaux. D'autre part, il nous a rendu possible l'expérimentation des noyaux rationnels dans une plateforme unifiée. Néanmoins, l'analyse des résultats et des performances obtenus ont permis de détecter quelques limites de l'approche proposée, différentes pistes ont alors émergées :

1. Pour l'extraction des radicaux, la mutation des lettres, dans le cas des mots à base de radicaux faibles, pose un problème sérieux. Ce phénomène nécessite d'être abordé en utilisant plus de connaissances sur les règles linguistiques et morphologiques. Résoudre ce problème passe par une étape principale qui consiste à décider d'abord si un mot a subit ou pas une mutation de ces radicaux.
2. La technique que nous avons proposé pour l'extraction des radicaux peut être naturellement étendue pour extraire aussi les racines ("light stems"). Ceci est possible du fait qu'elle se base sur les motifs pour analyser un mot.
3. Concernant le problème de CDA, la représentation d'un document par un transducteur linéaire a empêché tout recours aux techniques de sélection de caractéristiques. Ces techniques peuvent améliorer grandement les performances d'un système de CDA, une solution consiste à pondérer les transducteurs. Les termes à sélectionner recevront des valeurs importantes, tandis que les termes à exclure auront des valeurs faibles.
4. Une autre piste envisageable concerne l'analyse de l'effet d'extraction des racines ("light stems") sur les systèmes de CDA. Cette piste peut être explorée conjointement avec la deuxième piste.

Bibliographie

[ابن الحسن العلمي، ادريس ,2011] ابن الحسن العلمي، ادريس (2011). لا اتزان الا بالأوزان. مجلة اللّسان العربي . 70

[Abu-Errub et al., 2014] Abu-Errub, A., Odeh, A., Shambour, Q., and Hassan, O. A.-H. (2014). Arabic Roots Extraction Using Morphological Analysis. *International Journal of Computer Science Issues (IJCSI)*, 11(2) :128. 55

[Abu-Mostafa et al., 2012] Abu-Mostafa, Y. S., Magdon-Ismail, M., and Lin, H.-T. (2012). *Learning From Data*. AMLBook. 9

[Aggarwal and Zhai, 2012] Aggarwal, C. C. and Zhai, C. (2012). A Survey of Text Classification Algorithms. In Aggarwal, C. C. and Zhai, C., editors, *Mining Text Data*, pages 163–222. Springer. 23, 24

[Al Ameed et al., 2005] Al Ameed, H., Al Ketbi, S., Al Kaabi, A., Al Shebli, K., Al Shamsi, N., Al Nuaimi, N., and Al Muhairi, S. (2005). Arabic light stemmer : A new enhanced approach. In *The Second International Conference on Innovations in Information Technology (IIT'05)*, pages 1–9. 58

[Al-diabat, 2012] Al-diabat, M. (2012). Arabic text categorization using classification rule mining. *Applied Mathematical Sciences*, 6(81) :4033–4046. 62, 67

[Al-Harbi et al., 2008] Al-Harbi, S., Almuhareb, A., Al-Thubaity, A., Khorsheed, M., and Al-Rajeh, A. (2008). Automatic arabic text classification. 60, 66

[Al-Kabi et al., 2015] Al-Kabi, M. N., Kazakzeh, S. A., Ata, B. M. A., Al-Rababah, S. A., and Alsmadi, I. M. (2015). A novel root based Arabic stemmer . *Journal of King Saud University - Computer and Information Sciences*, 27(2) :94 – 103. 55

[Al-Nashashibi et al., 2010] Al-Nashashibi, M., Neagu, D., and Yaghi, A. (2010). An improved root extraction technique for Arabic words. In *2010 2nd Interna-*

tional Conference on Computer Technology and Development, pages 264–269. 55

[Al-Radaideh et al., 2011] Al-Radaideh, Q. A., Al-Shawakfa, E. M., Ghareb, A. S., and Abu-Salem, H. (2011). An approach for Arabic text categorization using association rule mining. *International Journal of Computer Processing of Languages*, 23(01) :81–106. 66

[Al-Salemi and Aziz, 2011] Al-Salemi, B. and Aziz, M. J. A. (2011). Statistical bayesian learning for automatic arabic text categorization. *Journal of Computer Science*, 7(1) :39. 62

[Al-Serhan et al., 2003] Al-Serhan, H., Shalabi, R. A., and Kannan, G. (2003). New Approach For Extracting Arabic Roots. In *Proceedings of The 2003 Arab Conf. on Infor. Technology*, pages 42–59, Alexandria, Egypt. 5, 55, 76

[Al-Shalabi et al., 2006] Al-Shalabi, R., Kanaan, G., and Gharaibeh, M. (2006). Arabic Text Categorization using KNN Algorithm. In *Proceedings of The 4th International Multiconference on Computer Science and Information Technology*, volume 4, pages 5–7. 60, 65, 66

[Al-Tahrawi,] Al-Tahrawi, M. M. Arabic Text Categorization Using Logistic Regression. *I.J. Intelligent Systems and Applications*, 7(6) :71–78. 67

[Al-Tahrawi and Al-Khatib, 2015] Al-Tahrawi, M. M. and Al-Khatib, S. N. (2015). Arabic text classification using polynomial networks. *Journal of King Saud University - Computer and Information Sciences*, 27(4) :437 – 449. 62, 67

[Al-Thubaity et al., 2015] Al-Thubaity, A., Alhoshan, M., and Hazzaa, I. (2015). Using word n-grams as features in arabic text classification. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 35–43. Springer. 3, 67, 68

[Alahmadi et al., 2014a] Alahmadi, A., Joorabchi, A., and Mahdi, A. (2014a). Combining Bag-of-Words and Bag-of-Concepts representations for Arabic text classification. In *Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014). 25th IET*, pages 343–348. 67

[Alahmadi et al., 2014b] Alahmadi, A., Joorabchi, A., and Mahdi, A. E. (2014b). Arabic Text Classification using Bag-of-Concepts Representation. In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval (IC3K 2014)*, pages 374–380. 67

[Alajmi et al., 2011] Alajmi, A., Saad, E., and Awadalla, M. (2011). Hidden markov model based Arabic morphological analyzer. *International Journal of Computer Engineering Research, IJCER*, 2(2) :28–33. 59

[Alhutaish and Omar, 2015] Alhutaish, R. and Omar, N. (2015). Arabic text classification using k-nearest neighbour algorithm. *International Arab Journal of Information Technology*, 12(2) :190–195. 67

[Aljlayl and Frieder, 2002] Aljlayl, M. and Frieder, O. (2002). On Arabic Search : Improving the Retrieval Effectiveness Via Light Stemming Approach. In *ACM Eleventh Conference on Infor. and Knowledge Management*, pages 340–347. 2, 53, 58, 68

[Allauzen et al., 2007] Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., and Mohri, M. (2007). OpenFst : A General and Efficient Weighted Finite-State Transducer Library. In *Proceedings of the Ninth International Conference on Implementation and Application of Automata, (CIAA 2007)*, volume 4783 of *LNCS*, pages 11–23. Springer. `http://www.openfst.org`. 76

[Allauzen, Cyril and Mohri, Mehryar, 2008] Allauzen, Cyril and Mohri, Mehryar (2008). 3-Way Composition of Weighted Finite-State Transducers. In *Proceedings of the 13th International Conference on Implementation and Applications of Automata*, CIAA '08, pages 262–273, Berlin, Heidelberg. Springer-Verlag. 46

[Alsaad and Abbod, 2014] Alsaad, A. and Abbod, M. (2014). Arabic text root extraction via morphological analysis and linguistic constraints. In *Computer Modelling and Simulation (UKSim), 2014 UKSim-AMSS 16th International Conference on*, pages 125–130. IEEE. 55

[Alsaleem, 2011] Alsaleem, S. (2011). Automated Arabic Text Categorization Using SVM and NB. *Int. Arab J. e-Technol.*, 2(2) :124–128. 62, 66, 67

[Alserhan and Ayesh, 2006] Alserhan, H. M. and Ayesh, A. S. (2006). An Application of Neural Network for Extracting Arabic Word Roots. *WSEAS Transactions on Computers*, 5(11) :2623–2627. 55, 56

[Alshalabi, 2005] Alshalabi, R. (2005). Pattern-based stemmer for finding arabic roots. *Information Technology Journal*, 4(1) :38–43. 55

[Althubaity et al., 2008] Althubaity, A., Almuhareb, A., Alharbi, S., Al-Rajeh, A., and Khorsheed, M. (2008). KACST Arabic Text Classification Project : Overview and Preliminary Results. In *Proceedings of The 9th IBIMA conference on Information Management in Modern Organizations*. 67, 87

[Bawaneh et al., 2008] Bawaneh, M., S. Alkoffash, M., and I. Al Rabea, A. (2008). Arabic Text Classification using K-NN and Naive Bayes. *Journal of Computer Science*, 4(7) :600–605. 61, 66

[Belkebir and Guessoum, 2013] Belkebir, R. and Guessoum, A. (2013). A hybrid BSO-Chi2-SVM approach to Arabic text categorization. In *Computer Systems and Applications (AICCSA), 2013 ACS International Conference on*, pages 1–7. IEEE. 62, 67, 68

[Berstel, 1979] Berstel, J. (1979). *Transductions and Context-Free Languages.* Teubner Studienbücher, Stuttgart. 42

[Boudlal et al., 2011] Boudlal, A., Bebah, M. O. A. O., Lakhouaja, A., Mazroui, A., and Meziane, A. (2011). A Markovian Approach for Arabic Root Extraction. *iajit*, 8(1) :91–98. 55, 56

[Buckwalter, 2004] Buckwalter, T. (2004). Buckwalter Arabic Morphological Analyzer Version 2.0. Linguistic Data Consortium, University of Pennsylvania, 2002. LDC Cat alog No. : LDC2004L02. Technical report, ISBN 1-58563-324-0. 54, 55

[Cavnar and Trenkle, 1994] Cavnar, W. B. and Trenkle, J. M. (1994). N-gram based text categorization. In *In Proc. of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175. 19

[Chang and Lin, 2011] Chang, C.-C. and Lin, C.-J. (2011). LIBSVM : A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2 :27 :1–27 :27. 88, 89

[Cherif et al., 2015] Cherif, W., Madani, A., and Kissi, M. (2015). New rules-based algorithm to improve arabic stemming accuracy. *International Journal of Knowledge Engineering and Data Mining*, 3(3-4) :315–336. 55

[Cortes et al., 2004] Cortes, C., Haffner, P., and Mohri, M. (2004). Rational Kernels : Theory and Algorithms. *J. Mach. Learn. Res.*, 5 :1035–1062. 41, 46, 47, 84

[Cortes et al., 2007] Cortes, C., Kontorovich, L., and Mohri, M. (2007). Learning languages with rational kernels. In *Proceedings of the 20th annual conference on Learning Theory*, COLT'07, pages 349–364. 42, 84

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-Vector Networks. In *Machine Learning*, pages 273–297. 25, 35

[Dietterich, 2000] Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. In *Multiple Classifier Systems, LNCS-1857*, pages 1–15. Springer. 25

[Drucker et al., 1999] Drucker, H., Wu, D., and Vapnik, V. N. (1999). Support vector machines for spam categorization. *Neural Networks, IEEE Transactions on*, 10(5) :1048–1054. 25

[Dumais and Chen, 2000] Dumais, S. and Chen, H. (2000). Hierarchical classification of Web content. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 256–263. ACM. 25

[Duwairi et al., 2009] Duwairi, R., Al-Refai, M. N., and Khasawneh, N. (2009). Feature reduction techniques for Arabic text categorization. *Journal of the American Society for Information Science and Technology*, 60(11) :2347–2352. 66

[Duwairi, 2006] Duwairi, R. M. (2006). Machine learning for arabic text categorization. *Journal of the American Society for Information Science and Technology*, 57(8) :1005–1010. 60, 65

[Duwairi, 2007] Duwairi, R. M. (2007). Arabic Text Categorization. *Int. Arab J. Inf. Technol.*, 4(2) :125–132. 60, 66

[El-Halees, 2007] El-Halees, A. (2007). Arabic text classification using maximum entropy. *The Islamic University Journal (Series of Natural Studies and Engineering)*, 15(1) :157–167. 60

[El Kourdi et al., 2004] El Kourdi, M., Bensaid, A., and Rachidi, T.-e. (2004). Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, Semitic '04, pages 51–58, Stroudsburg, PA, USA. Association for Computational Linguistics. 60, 65

[Feldman and Sanger, 2006] Feldman, R. and Sanger, J. (2006). *Text Mining Handbook : Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, New York, NY, USA. 13, 17, 20, 25

[Gadri and Moussaoui, 2015] Gadri, S. and Moussaoui, A. (2015). Arabic texts categorization : Features selection based on the extraction of words' roots. *IFIP Advances in Information and Communication Technology*, 456 :167–180. 67

[Gartner, 2009] Gartner, T. (2009). *Kernels For Structured Data*. World Scientific Publishing Co., Inc., River Edge, NJ, USA. 30

[Gharib et al., 2009] Gharib, T., Habib, M., and Fayed, Z. (2009). Arabic Text Classification Using Support Vector Machines. *International Journal of Computers and Their Applications*, 16(4) :192–199. 61, 66

[Ghwanmeh et al., 2009] Ghwanmeh, S., Kanaan, G., Al-Shalabi, R., and Rabab'ah, S. (2009). Enhanced Algorithm for Extracting the Root of Arabic Words. In *Sixth International Conference on Computer Graphics, Imaging and Visualization, 2009. CGIV'09.*, pages 388–391. IEEE. 55

[Habash, 2010] Habash, N. (2010). *Introduction to Arabic Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers. 52

[Hadni et al., 2013] Hadni, M., Ouatik, S. A., and Lachkar, A. (2013). Effective Arabic Stemmer Based Hybrid Approach for Arabic Text Categorization. *International Journal of Data Mining & Knowledge Management Process (IJDKP)* , 3. 55

[Han et al., 2011] Han, J., Kamber, M., and Pei, J. (2011). *Data Mining : Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition. 17, 28

[Haralambous et al., 2014] Haralambous, Y., Elidrissi, Y., and Lenca, P. (2014). Arabic language text classification using dependency syntax-based feature selection. *arXiv preprint arXiv :1410.4863*. 67

[Harmanani et al., 2006] Harmanani, H. M., Keirouz, W., and Raheel, S. (2006). A Rule-Based Extensible Stemmer for Information Retrieval with Application to Arabic. *Int. Arab J. Inf. Technol.*, 3(3) :265–272. 55, 58

[Harrag and El-Qawasmah, 2009] Harrag, F. and El-Qawasmah, E. (2009). Neural network for arabic text classification. In *Applications of Digital Information and Web Technologies, 2009. ICADIWT'09. Second International Conference on the*, pages 778–783. IEEE. 61, 66

[Harrag et al., 2011] Harrag, F., El-Qawasmah, E., and Al-Salman, A. (2011). Stemming as a feature reduction technique for Arabic Text Categorization. In *10th International Symposium on Programming and Systems (ISPS), 2011*, pages 128–133. 66

[Harrag et al., 2010] Harrag, F., El-Qawasmeh, E., and Al-Salman, A. M. S. (2010). A comparative study of statistical feature reduction methods for arabic text categorization. In *Networked Digital Technologies*, pages 676–682. Springer. 66

[Harrag et al., 2009] Harrag, F., El-Qawasmeh, E., and Pichappan, P. (2009). Improving Arabic text categorization using decision trees. In *Networked Digital Technologies, 2009. NDT'09. First International Conference on*, pages 110–115. IEEE. 61, 66

[Hawas and Emmert, 2014] Hawas, F. and Emmert, K. (2014). Rule-based approach for arabic root extraction : New rules to directly extract roots of arabic words. *Journal of Computing and Information Technology*, 22(1) :57–68. cited By 0. 55

[Hmeidi et al., 2014] Hmeidi, I., Al-Ayyoub, M., Abdulla, N. A., Almodawar, A. A., Abooraig, R., and Mahyoub, N. A. (2014). Automatic arabic text categorization : A comprehensive comparative study. *Journal of Information Science*. 2

[Hmeidi et al., 2010] Hmeidi, I., Al-Shalabi, R., Al-Taani, A., Najadat, H., and Al-Hazaimeh, S. (2010). A novel approach to the extraction of roots from arabic words using bigrams. *Journal of the American Society for Information Science and Technology*, 61(3) :583–591. 55, 56

[Huffman, 1995] Huffman, S. (1995). Acquaintance : Language-independent document categorization by n-grams. Technical report, DTIC Document. 19

[Joachims, 1998] Joachims, T. (1998). Text Categorization with Support Vector Machines : Learning with Many Relevant Features. 25

[Kadri and Nie, 2006] Kadri, Y. and Nie, J.-Y. (2006). Effective stemming for Arabic information retrieval. In *The Challenge of Arabic for NLP/MT, Intl Conf. at the BCS*, pages 68–74. 56

[Kanaan et al., 2008] Kanaan, G., Al-Shalabi, R., Ababneh, M., and Al-Nobani, A. (2008). Building an Effective Rule-based Light Stemmer for Arabic Language to Improve Search Effectiveness. In *Innovations in Information Technology, 2008. IIT 2008. International Conference on*, pages 312–316. IEEE. 58

[Khaliq and Carroll, 2013] Khaliq, B. and Carroll, J. (2013). Unsupervised induction of arabic root and pattern lexicons using machine learning. pages 350–356. cited By 0. 55

[Khoja and Garside, 1999] Khoja, S. and Garside, R. (1999). Stemming arabic text. Technical report, Computing Department, Lancaster University. 5, 55, 56, 76

[Khorsi, 2012] Khorsi, A. (2012). Effective unsupervised Arabic word stemming : Towards an unsupervised radicals extraction. *International Arab Journal of Information Technology*, 9(6). cited By 1. 59

[Khreisat, 2009] Khreisat, L. (2009). A machine learning approach for Arabic text classification using N-gram frequency statistics. *Journal of Informatrics*, 3(1) :72–77. 3, 61, 66

[Klinkenberg and Joachims, 2000] Klinkenberg, R. and Joachims, T. (2000). Detecting Concept Drift with Support Vector Machines. In *ICML*, pages 487–494. 25

[Larkey et al., 2002] Larkey, L. S., Ballesteros, L., and Connell, M. E. (2002). Improving Stemming for Arabic Information Retrieval : Light Stemming and Co-occurrence Analysis. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 275–282, New York, NY, USA. ACM. 2, 58

[Larkey et al., 2007] Larkey, L. S., Ballesteros, L., and Connell, M. E. (2007). Light Stemming for Arabic Information Retrieval. In *Arabic computational morphology*, pages 221–243. Springer. 58

[Manning et al., 2008] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA. 20, 23

[Maron, M. E., 1961] Maron, M. E. (1961). Automatic indexing : An experimental inquiry. *J. ACM*, 8(3) :404–417. 13

[Mehrotra et al., 1997] Mehrotra, K., Mohan, C. K., and Ranka, S. (1997). *Elements of artificial neural networks*. MIT press. 23

[Mesleh, 2008] Mesleh, A. (2008). Support Vector Machines based Arabic Language Text Classification System : Feature Selection Comparative Study. In Sobh, T., editor, *Advances in Computer and Information Sciences and Engineering*, pages 11–16. Springer Netherlands. 66

[Mesleh, 2011] Mesleh, A. M. (2011). Feature sub-set selection metrics for arabic text classification. *Pattern Recognition Letters*, 32(14) :1922 – 1929. 66

[Mitchell, 1997] Mitchell, T. (1997). *Machine Learning*. McGraw Hill edition. 8

[Moh'd A Mesleh, 2007] Moh'd A Mesleh, A. (2007). Chi square feature extraction based SVMs Arabic language text categorization system. *Journal of Computer Science*, 3(6) :430–435. 60, 66

[Momani and Faraj, 2007] Momani, M. and Faraj, J. (2007). A Novel Algorithm to Extract Tri-Literal Arabic Roots. In *Computer Systems and Applications, 2007. AICCSA'07. IEEE/ACS International Conference on*, pages 309–315. IEEE. 55

[Nehar et al., 2014] Nehar, A., Benmessaoud, A., Cherroun, H., and Ziadi, D. (2014). Subsequence kernels-based Arabic text classification. In *11th IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 2014, Doha, Qatar, November 10-13, 2014*, pages 206–213. IEEE. 5

[Nehar et al., 2013] Nehar, A., Ziadi, D., and Cherroun, H. (2013). Rational Kernels for Arabic Text Classification. In Dediu, A. H., Martín-Vide, C., Mitkov, R., and Truthe, B., editors, *Statistical Language and Speech Processing - First International Conference, SLSP 2013, Tarragona, Spain, July 29-31, 2013. Proceedings*, volume 7978 of *Lecture Notes in Computer Science*, pages 176–187. Springer. 5

[Nehar et al., 2016] Nehar, A., Ziadi, D., and Cherroun, H. (2016). Rational Kernels for Arabic Root Extraction and Text Classification. *J. King Saud Univ. Comput. Inf. Sci.*, 28(2) :157–169. 5

[Nehar et al., 2012] Nehar, A., Ziadi, D., Cherroun, H., and Guellouma, Y. (2012). An Efficient Stemming for Arabic Text Classification. In *Innovations in Information Technology (IIT), 2012 International Conference on*. 5

[Odeh et al.,] Odeh, A., Abu-Errub, A., Shambour, Q., and Turab, N. Arabic Text Categorization Algorithm using Vector Evaluation Method. *ijcsit*, 6(6) :83–92. 67

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of Decision Trees. 1 :81–106. 23

[Raheel and Dichy, 2010] Raheel, S. and Dichy, J. (2010). An empirical study on the feature's type effect on the automatic classification of arabic documents. In *Computational Linguistics and Intelligent Text Processing*, pages 673–686. Springer. 66

[علي حلمي موسى, 1978] علي حلمي موسى (1978). دراسة احصائية لجذور معجم الصحاح . الدار المصرية للكتاب. 74

[Rogati et al., 2003] Rogati, M., McCarley, S., and Yang, Y. (2003). Unsupervised learning of arabic stemming using a parallel corpus. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 391–398. Association for Computational Linguistics. 58

[Saad and Ashour, 2010] Saad, M. K. and Ashour, W. (2010). Arabic text classification using decision trees. In *Proceedings of the 12th international workshop*

on computer science and information technologies CSIT, pages 75–79. 61, 66, 67

[Samuel, 2000] Samuel, A. L. (2000). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 44(1) :206–227. 7

[Sawaf et al., 2001] Sawaf, H., Zaplo, J., and Ney, H. (2001). Statistical classification methods for Arabic news articles. *Natural Language Processing in ACL2001, Toulouse, France*. 60, 65

[Sawalha, 2011] Sawalha, M. (2011). *Open-source Resources and Standards for Arabic Word Structure Analysis*. PhD, University of Leeds, Leeds. 76

[Scholkopf and Smola, 2001] Scholkopf, B. and Smola, A. J. (2001). *Learning with kernels : support vector machines, regularization, optimization, and beyond*. MIT Press. 32

[Sebastiani, 2006] Sebastiani, F. (2006). Classification of text, automatic. In *The Encyclopedia of Language and Linguistics*, pages 457–463. Elsevier Science Publishers. 1

[Sebastiani and Ricerche, 2002] Sebastiani, F. and Ricerche, C. N. D. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34 :1–47. 1, 13, 24

[Sharef et al., 2014] Sharef, B. T., Omar, N., and Sharef, Z. T. (2014). An automated arabic text categorization based on the frequency ratio accumulation. *Int. Arab J. Inf. Technol.*, 11(2) :213–221. 62

[Shawe-Taylor and Cristianini, 2004] Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA. iv, 30, 33, 34, 35

[Sokolova and Lapalme, 2009] Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4) :427–437. 11

[Syiam et al., 2006] Syiam, M., Fayed, Z., and Habib, M. (2006). An Intelligent System For Arabic Text Categorization. *International Journal of Intelligent Computing and Information Sciences*, 6(1) :1–19. 60, 65, 66

[Taghva et al., 2005] Taghva, K., Elkhoury, R., and Coombs, J. S. (2005). Arabic Stemming Without A Root Dictionary. In *ITCC (1)*, pages 152–157. 55, 56

[Taira and Haruno, 1999] Taira, H. and Haruno, M. (1999). Feature selection in SVM text categorization. In *AAAI/IAAI*, pages 480–486. 25

[Thabtah et al., 2009] Thabtah, F., Eljinini, M., Zamzeer, M., and Hadi, W. (2009). Naïve Bayesian based on Chi Square to categorize Arabic data. In *proceedings of The 11th International Business Information Management Association Conference (IBIMA) Conference on Innovation and Knowledge Management in Twin Track Economies, Cairo, Egypt*, pages 4–6. Citeseer. 61, 66, 67

[Xu and Croft, 1998] Xu, J. and Croft, W. B. (1998). Corpus-based stemming using cooccurrence of word variants. *ACM Transactions on Information Systems (TOIS)*, 16(1) :61–81. 58

[Yahya and Salhi, 2014] Yahya, A. and Salhi, A. (2014). Arabic Text Categorization Based on Arabic Wikipedia. *ACM Transactions on Asian Language Information Processing (TALIP)*, 13(1) :4. 67

[Yang, 1994] Yang, Y. (1994). Expert network : Effective and efficient learning from human decisions in text categorization and retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 13–22. Springer-Verlag New York, Inc. 13

[Yang and Liu, 1999] Yang, Y. and Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49. ACM. 25

[Yaseen and Hmeidi, 2014] Yaseen, Q. and Hmeidi, I. (2014). Extracting the Roots of Arabic Words without Removing Affixes. *Journal of Information Science*, 40(3) :376–385. 55

[Yousef et al., 2014] Yousef, N., Abu-Errub, A., Odeh, A., and Khafajeh, H. (2014). An Improved Arabic Word's Roots Extraction Method Using N-Gram Technique. *Journal of Computer Science*, 10(4) :716. 55, 56

[Yousif et al., 2015] Yousif, S. A., Samawi, V. W., Elkaban, I., and Zantout, R. (2015). Enhancement of Arabic Text Classification Using Semantic Relations of Arabic WordNet. *Journal of Computer Science*, 11(3) :498–509. 67

Annexe A

Liste des motifs utilisés

Tableau A.1: Liste des motifs de noms utilisés

Le motif	Exemple
فعل	علم
فاعل	عاقل
مفعل	مسجد
فعال	كلام
فعول	سرور
فوعل	كوكب
فعيل	رقيب
فعال	سعال
افعل	اخضر
فعلى	ذكرى
فاعول	قانون
فواعل	صواعق
مفاعل	مجالس
مفتعل	منتصر
مفعلل	مدحرج
مفعال	مصباح
مفعول	مضروب
مفعيل	مسكين
أفعال	أسباب
إفعيل	إكليل
أفعول	أسلوب
تفعال	تجفاف
تفعيل	تنظيف
فعوال	عنوان
فعالة	عمامة
فعولة	عمومة
أفاعيل	أعاصير
مفاعيل	مناديل
متفاعل	متطابق
مستفعل	مستشرق
مفعوعل	معشوشب
متفعلل	متدحرج
انفعال	انطلاق
افتعال	اجتماع
افعلال	اصفرار
فاعولة	نافورة
استفعال	استخراج
افعيلال	اخضيرار

Tableau A.2: Liste des motifs de verbes utilisés

Le motif	Exemple
فعل	ذهب
فاعل	صافح
فعلل	دحرج
افعلّ	اصفرّ
انفعل	انطلق
افتعل	اختبر
تفعّل	تبتّل
تمفعل	تمسكن
تفعلل	تدحرج
افعنلل	احرنجم
استفعل	استخدم
افعوعل	اعشوشب
افعالّ	اصفارّ