UNIVERSITÉ LUMIÈRE LYON2
Année 2003

THÈSE
pour obtenir le grade de
DOCTEUR
en
INFORMATIQUE

présentée et soutenue publiquement par

Radwan JALAM
le 4 juin 2003

Apprentissage automatique et catégorisation de textes multilingues

---

préparée au sein du laboratoire ERIC
Equipe de Recherche en Ingénierie des Connaissances

sous la direction de
Jean-Hugues CHAUCHAT

DEVANT LE JURY, COMPOSÉ DE :

Annie MORIN, Rapporteur	Maître de conférences habilitée, IRISA, Rennes
Yves KODRATOFF, Rapporteur	Directeur de recherche, CNRS, LRI Orsay
Martin RAJMAN, Rapporteur	Professeur, Ecole Polytechnique Fédérale, Lausanne
Geneviève BOIDIN-LALLICH, Examinateur	Professeur, Université Claude Bernard-Lyon 1
Ludovic LEBART, Examinateur	Directeur de recherche, CNRS, ENST Paris
Jean-Hugues CHAUCHAT, Directeur de thèse	Professeur, Université Lumière-Lyon 2

Table des matières

Introduction **1**

I. Catégorisation de textes monolingues **5**

1. Catégorisation de textes **7**
1.1. Introduction 8
1.2. Définition de la catégorisation de texte 8
1.3. Comment catégoriser un texte ? 9
1.3.1. Représentation, le codage, des textes 10
1.3.2. Choix de classifieurs 11
1.3.3. Évaluation de la qualité des classifieurs 12
1.4. Applications de la catégorisation de texte 12
1.4.1. Catégorisation de textes : une fin en soi 13
1.4.2. Catégorisation de textes : un support pour différentes applications 13
1.5. Difficultés particulières de la catégorisation de textes 13
1.5.1. Grandes dimensions 14
1.5.2. Imprécision des fréquences 15
1.5.3. Déséquilibrage 15
1.5.4. Ambiguïté 15
1.5.5. Synonymie 15
1.5.6. Subjectivité de la décision 16
1.6. Lien avec la recherche documentaire 16
1.7. Jeu de données utilisé pour l'évaluation 18
1.8. Conclusion 19

2. Approches pour la représentation de textes **21**
2.1. Introduction 22

2.2. Choix de termes . . . 22
2.2.1. Représentation en « sac de mots » . . . 22
2.2.2. Représentation des textes par des phrases . . . 23
2.2.3. Représentation des textes avec des racines lexicales et des lemmes . . . 24
2.2.4. Méthodes basées sur les n-grammes . . . 24
2.3. Codage des termes . . . 26
2.3.1. Codage TF $\times$ IDF . . . 27
2.3.2. Codage TFC . . . 27
2.4. Réduction de la dimension . . . 28
2.4.1. Réduction locale de dimension . . . 29
2.4.2. Réduction globale de dimension . . . 29
2.4.3. Sélection de termes . . . 29
2.4.4. Extraction de termes . . . 30
2.5. Conclusion . . . 30

3. Sélection multivariée de termes 33
3.1. Introduction . . . 34
3.2. Méthode du χ^2 univariée . . . 34
3.3. Méthode du χ^2 multivarié . . . 36
3.4. Expérimentation . . . 36
3.5. Conclusion . . . 38

4. Pourquoi les n-grammes fonctionnent 39
4.1. Introduction . . . 40
4.2. Intérêt du codage en n-grammes . . . 40
4.3. Étapes de la recherche des mots caractéristiques . . . 41
4.3.1. Recherche des n-grammes caractéristiques et des mots qui les contiennent . . . 41
4.3.2. Filtrage des mots « parasites » . . . 42
4.3.3. Algorithme complet . . . 42
4.4. Exemple d'application . . . 42
4.4.1. Données indexées de Reuters . . . 42
4.4.2. Quelques résultats . . . 44
4.4.3. Discussion des résultats sur la collection Reuters . . . 44
4.5. Conclusion . . . 46

5. Techniques pour la construction de classifieurs 51
5.1. Introduction . . . 52
5.1.1. Manière de construction du classifieur . . . 52
5.1.2. Caractéristique du modèle . . . 53
5.2. Méthode de Rocchio . . . 53
5.3. Arbres de décision . . . 55

5.3.1. Phase d'apprentissage 56
5.3.2. Phase de classification 61
5.3.3. Critiques de la méthode 61
5.4. Classifieurs à base d'exemples 62
5.4.1. K-plus proches voisins 63
5.5. Fonctions à bases radiales 67
5.6. Machine à Vecteurs de Support 68
5.6.1. Cas des classes linéairement séparables 69
5.6.2. Cas des classes non séparables 70
5.7. Évaluation de classifieurs de textes 71
5.7.1. Évaluation des classifieurs, l'approche « binaire » 72
5.7.2. Évaluation des classifieurs, l'approche « multi-classes » . . 76
5.8. Contributions personnelles 77
5.8.1. Nouvelle utilisation des SVM 77
5.8.2. Nouvelle utilisation des réseaux RBF 77
5.8.3. Nos expérimentations 77
5.9. Conclusion : quel est le meilleur classifieur ? 79

II. Catégorisation de textes multilingues 83

6. Catégorisation multilingue : les solutions proposées 85
6.1. Introduction 86
6.2. Intérêt accru aux traitements multilingues 86
6.2.1. Davantage de collections numériques 86
6.2.2. Plus de personnes connectées en ligne 87
6.2.3. Plus de globalisation et de pays unifiés 87
6.2.4. Réseau plus rapide et plus souple 88
6.3. Recherche documentaire multilingues 88
6.3.1. Approches basées sur la traduction automatique 89
6.3.2. Thésaurus multilingues 90
6.3.3. Utilisation de dictionnaires 90
6.4. Nos solutions pour catégoriser des textes multilingues 91
6.4.1. Premier schéma : le schéma trivial 92
6.4.2. Deuxième schéma : choisir une seule langue d'apprentissage 93
6.4.3. Troisième schéma : mélanger les ensembles d'apprentissage 93
6.5. Conclusion 94

7. Identification de la langue 97
7.1. Introduction 98
7.2. Approches linguistiques 99
7.2.1. Présence de certains chaînes de caractères spécifiques 99
7.2.2. Présence de certains mots 100

7.2.3. Approche lexicale . . . 101
7.2.4. Approche plus linguistique . . . 101
7.3. Approches statistiques et probabilistes . . . 102
7.3.1. Mots les plus fréquents . . . 102
7.3.2. Méthodes basées sur les *n*-grammes . . . 103
7.4. Expériences pour la reconnaissance de langue . . . 107
7.5. Conclusion . . . 109

8. Traduction automatique **111**
8.1. Introduction . . . 112
8.2. Premières approches historiques de la traduction automatique . . . 112
8.2.1. Décryptage . . . 112
8.2.2. Analyse par micro-contexte . . . 112
8.2.3. Imiter la traduction humaine . . . 113
8.3. Nouvelles approches, plus modestes . . . 113
8.3.1. Mémoire de traduction . . . 113
8.3.2. Sous-langages et langages contrôlés . . . 114
8.4. Évaluer la traduction automatique . . . 114
8.5. Conclusion . . . 115

9. Cadre pour la catégorisation de textes multilingues **117**
9.1. Introduction . . . 118
9.2. Méthodes pour la catégorisation de textes multilingues . . . 118
9.2.1. Nouveau cadre pour la catégorisation multilingue . . . 118
9.2.2. Détection de la langue du texte à classer . . . 119
9.2.3. Traduction du texte à classer . . . 119
9.3. Application sur les corpus CLEF . . . 120
9.3.1. Constitution du corpus . . . 120
9.3.2. Représentation des textes . . . 122
9.3.3. Algorithmes d'apprentissage . . . 123
9.3.4. Reconnaissance de la langue . . . 123
9.3.5. Catégorisation des articles . . . 123
9.4. Discussion . . . 128
9.5. Conclusion . . . 129

Conclusion et perspectives **133**

Index des auteurs cités **137**

Bibliographie **141**

Première partie .

Catégorisation de textes monolingues

Chapitre 1

Catégorisation de textes

Sommaire

1.1. Introduction 8
1.2. Définition de la catégorisation de texte 8
1.3. Comment catégoriser un texte ? 9
1.3.1. Représentation, le codage, des textes 10
1.3.2. Choix de classifieurs 11
1.3.3. Évaluation de la qualité des classifieurs 12
1.4. Applications de la catégorisation de texte 12
1.4.1. Catégorisation de textes : une fin en soi 13
1.4.2. Catégorisation de textes : un support pour différentes applications 13
1.5. Difficultés particulières de la catégorisation de textes 13
1.5.1. Grandes dimensions 14
1.5.2. Imprécision des fréquences 15
1.5.3. Déséquilibrage 15
1.5.4. Ambiguïté 15
1.5.5. Synonymie 15
1.5.6. Subjectivité de la décision 16
1.6. Lien avec la recherche documentaire 16
1.7. Jeu de données utilisé pour l'évaluation 18
1.8. Conclusion 19

1.1. Introduction

La quantité d'information disponible sous format électronique sur Internet ou dans l'intranet des entreprises croît de façon extrêmement rapide. Par exemple, le 22 juillet 2002, Google avait recensés 2.073.418.204 pages ; le 16 mars 2003, ce nombre est passé à 3.083.324.652 pages, soit moitié plus en moins d'un an (voir `http://www.google.fr`).

L'utilisateur, submergé par cette masse d'informations, ne se pose plus la question d'*accéder à l'information*. Son problème devient : *comment trouver l'information* dont il a besoin, parmi toutes celles qui est accessible.

Dans ce contexte, *la catégorisation de texte*, définie comme le processus permettant d'associer une catégorie (ou classe) à un texte libre, en fonction des informations qu'il contient, est un élément important des systèmes de gestion de l'information.

Associer une classe à un texte libre est une opération coûteuse et longue, par conséquent, *l'automatisation* de cette opération est devenue un enjeu pour la communauté scientifique.

Dans ce chapitre, nous définissons la *catégorisation de textes*, nous décrivons le processus général de la catégorisation de textes et nous montrons ses *application*s et les *problèmes spécifiques* aux textes lors de l'apprentissage automatique. Enfin, nous montrons les relations entre la catégorisation de textes et la *recherche documentaire* et nous terminons ce chapitre en présentant le jeu de données habituellement utilisé dans la littérature.

1.2. Définition de la catégorisation de texte

La catégorisation de texte consiste à chercher une liaison fonctionnelle entre *un ensemble de textes* et *un ensemble de catégories* (étiquettes, classes). Cette liaison fonctionnelle, que l'on appelle également *modèle de prédiction*, est estimée par un apprentissage automatique (traduction de *machine learning method*). Pour ce faire, il est nécessaire de disposer d'un ensemble de textes préalablement étiquetés, dit *ensemble d'apprentissage*, à partir duquel nous estimons les paramètres du modèle de prédiction le plus performant possible, c'est-à-dire le modèle qui produit le moins d'*erreur* en prédiction.

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

Les catégories c_i sont choisies parmi un ensemble prédéfini, par exemple les mots clés autorisés par un journal scientifique, ou encore la classification générale utilisée par les bibliothécaires pour ranger les livres. Pour nous, les catégories sont des labels symboliques et aucune connaissance supplémentaire n'est disponible concernant leur signification. Théoriquement, afin d'avoir des techniques de catégorisation généralisables et indépendantes de toute donnée exogène, seules les informations endogènes peuvent intervenir dans la décision ; ceci suppose que les « metadonnées » comme la date de publication, le type de document, la source de publication n'interviennent pas dans la décision. Dans les applications réelles, ces exigences académiques sont rarement respectées car on cherche évidement à utiliser toute information disponible qui aide à la décision.

Notons que nous allons utiliser, dans ce mémoire, les deux termes classement et classification pour désigner le même concept à savoir la « catégorisation ».

1.3. Comment catégoriser un texte ?

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

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

Le processus de catégorisation, intégrant la phase de classement de nouveaux textes, est résumé dans la figure 1.1. Il comporte deux phases que l'on peut distinguer comme suit :

1. **l'apprentissage**, qui comprend plusieurs étapes et aboutit à un modèle de prédiction :
   a) nous disposons d'un ensemble de textes étiquetés (pour chaque texte nous connaissons sa catégorie) ;
   b) à partir de ce corpus, nous extrayons les k descripteurs (ou mots, ou termes) $(t_1; ..; t_k)$ les plus pertinents au sens du problème à résoudre ;
   c) nous disposons alors d'un tableau « descripteurs $\times$ individus », et pour chaque texte nous connaissons la valeur de ses descripteurs et son étiquette ;
   d) nous appliquons un algorithme d'apprentissage sur ce tableau afin d'obtenir un modèle de prédiction Φ.
2. **le classement** d'un nouveau texte d_x, qui comprend deux étapes :
   a) recherche puis pondération des occurrences $(t_1; ...; t_k)$ des termes dans le texte d_x à classer ;

b) application du modèle Φ sur ces occurrences afin de prédire l'étiquette de ce texte d_x.

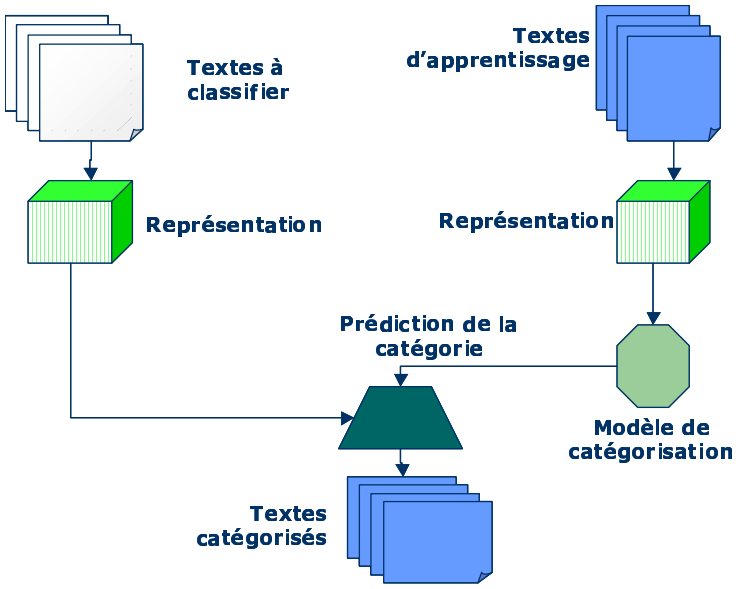

FIG. 1.1.: Processus de la catégorisation de textes

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

Dans la suite nous présentons brièvement ces étapes qui seront développées dans les chapitres 2, 5.

1.3.1. Représentation, le codage, des textes

Un codage préalable du texte est nécessaire, comme pour l'image, le son, etc., [Sebastiani, 2002], car il n'existe pas actuellement de méthode d'apprentissage capable de traiter directement des données non-structurées, ni dans la phase de construction du modèle, ni lors de son utilisation en classement.

Pour la majorité des méthodes d'apprentissage, il faut transformer l'ensemble des textes en un tableau croisé "individus-variables" :

- L'**individu** est un texte (un document) d_j, étiqueté lors de la phase d'apprentissage, et à classer dans la phase de prédiction.
- Les **variables** sont les descripteurs (les termes) t_k qui sont extraits des données textuelles.

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

Le principal enjeu de la catégorisation de texte, par rapport à un processus d'apprentissage classique, réside dans la recherche des descripteurs (ou termes) les plus pertinents pour le problème à traiter [Aas and Eikvil, 1999]. Différentes méthodes sont proposées pour le choix des descripteurs et des poids associés à ces descripteurs [Yang, 1999]. [Salton and McGill, 1983, Aas and Eikvil, 1999] utilisent, à titre d'exemples, les mots comme descripteurs, tandis que d'autres préfèrent utiliser les lemmes (racines lexicales) [Sahami, 1999] ; ou encore des *stemmes* (la suppression d'affixes)[de Loupy, 2001].

Il existe une autre approche de la représentation des textes : les **n-grammes** : les séquences de *n* caractères [Cavnar and Trenkle, 1994]. Les sections 2.2.1, 2.2.3 et 2.2.4 du chapitre 2 traiteront ces différentes approches et détailleront leurs avantages et limites.

L'exploitation des documents textuels dans l'espace des termes (mots, lemmes, stemmes, n-grammes, etc.) n'est pas possible sans réduction préalable de la dimension de cet espace car celle-ci est en générale trop grande (de l'ordre du millier). L'objectif des techniques de réduction est de déterminer un espace restreint de descripteurs conservant au mieux l'information originelle pour différencier les étiquettes de classement. La sélection de termes peut être effectuée soit localement : pour chaque catégorie c_i, un ensemble de termes $\mathcal{T}_i'$ avec $|\mathcal{T}_i'| \ll |\mathcal{T}_i|$ est choisi pour représenter c_i [Apté et al., 1994, Lewis and Ringuette, 1994], soit globalement : un ensemble de termes $\mathcal{T}'$ avec $|\mathcal{T}'| \ll |\mathcal{T}|$ est choisi pour représenter la totalité des classes $\mathcal{C}$ [Yang and Pedersen, 1997, Mladenić, 1998, Caropreso et al., 2001]. Dans la section 2.4 page 28 du chapitre 2, nous présentons en détail ces techniques pour ensuite, dans le chapitre 4 page 39, proposer et justifier notre choix de représentation qui est basée sur les n-grammes. Le chapitre 3 page 33 propose une nouvelle approche basée sur la statistique de χ^2 pour la sélection multi-variée de termes.

1.3.2. Choix de classifieurs

La catégorisation de textes comporte un choix de technique d'apprentissage (ou classifieur) disponibles. Parmi les méthodes d'apprentissage les plus souvent utilisées figurent l'analyse factorielle discriminante [Lebart and Salem, 1994], la régression logistique [Hull, 1994], les réseaux de neurones [Wiener et al., 1995, Schütze et al., 1995, Stricker, 2000], les plus proches voisins [Yang and Chute, 1994, Yang and Liu, 1999], les arbres de décision [Lewis and Ringuette, 1994, Apté et al., 1994], les réseaux bayesiens [Borko and Bernick, 1964, Lewis, 1998, Androutsopoulos et al., 2000, Chai et al., 2002, Adam et al., 2002], les machines à vecteurs supports [Joachims, 1998, Joachims, 1999, Joachims, 2000, Dumais et al., 1998, He et al., 2000] et, plus récemment, les méthodes dites de *boos-*

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

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

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

1.3.3. Évaluation de la qualité des classifieurs

Afin de pouvoir comparer les résultats produits par différents modèles, et afin de s'assurer que le modèle est généralisable à d'autres textes, nous devons appliquer une méthode d'évaluation. Toutes les mesures usuelles se basent sur le tableau de contingence 5.3 page 72. La performance d'un classifieur dans la catégorisation de textes est souvent mesurée via la précision (traduction de *precision*) et le rappel (traduction de *recall*). La précision π_i pour la classe c_i est définie comme la probabilité conditionnelle $P(\breve{\Phi}(d_x, c_i) = Vrai \mid \Phi(d_x, c_i) = Vrai)$, ce qu'on peut interpréter comme la probabilité conditionnelle qu'un exemple choisi aléatoirement dans la classe soit bien classé par le système. Le rappel ρ_i mesure la largeur de l'apprentissage et correspond à la fraction des documents pertinents, parmi ceux proposés par le classifieur. [Sebastiani, 2002] le présente comme $p(\Phi(d_x, c_i) = Vrai / \breve{\Phi}(d_x, c_i) = Vrai)$. Notons que les taux de succès et d'erreur sont rarement utilisés pour mesurer les performances d'un classifieur, nous détaillons les raisons de cela ainsi que les autres mesures de performances utilisées dans la littérature dans les sections 5.7.1 page 74 et 5.7.1 page 74.

1.4. Applications de la catégorisation de texte

Depuis les travaux de [Maron, 1961], la catégorisation de textes est utilisée dans de nombreuses applications. Parmi ces domaines figurent : l'identification de la langue [Cavnar and Trenkle, 1994], la reconnaissance d'écrivains [Forsyth, 1999, Teytaud and Jalam, 2001] et la catégorisation de documents multimédia [Sable and Hatzivassiloglou, 2000], et bien d'autres. Dans cette section, nous allons présenter un cadre général qui résume la manière dont la catégorisation de textes est utilisée.

La catégorisation de textes peut être une fin en soi, par exemple lors de l'étiquetage de documents, ou bien représenter une étape dans la représentation et le traitement de l'information contenues dans les textes [Moulinier, 1996].

1.4.1. Catégorisation de textes : une fin en soi

L'indexation automatique de textes consiste à associer à chaque texte d'une collection un ou plusieurs termes parmi un ensemble prédéfini. L'objectif est de décrire le contenu de ces textes par des mots ou des phrases clés qui font partie d'un ensemble de vocabulaire contrôlé. Dans un tel contexte, si nous regardons ce vocabulaire contrôlé comme des catégories, l'indexation de textes peut être alors vue comme une forme de catégorisation de textes [Fuhr and Knorz, 1984, Robertson and Harding, 1984, Hayes and Weinstein, 1990, Fuhr et al., 1991, Tzeras and Hartmann, 1993].

1.4.2. Catégorisation de textes : un support pour différentes applications

La catégorisation de textes peut être un support pour différentes applications parmi lesquelles le filtrage, consistant à déterminer si un document est pertinent ou non (décision binaire), et le routage, consistant à affecter un document à une ou plusieurs catégories parmi n.

Un exemple de l'utilisation de la catégorisation de texte pour le filtrage est la détection de *spams* (les courriers indésirables) pour ensuite les supprimer [Androutsopoulos et al., 2000, Cohen, 1996]. Un exemple de routage est la diffusion sélective d'information. Lors de la réception d'un document l'outil choisit à quelles personnes le faire parvenir en fonction de leurs centres d'intérêt. Ces centres d'intérêt correspondent à des profils individuels [Liddy et al., 1994].

Notons que si la sélection est faite au niveau du producteur de l'information (*e.g.*, l'agence de presse), le système doit « diriger » l'information au consommateur intéressé (*e.g.*, le journal) [Liddy et al., 1994] et on parle alors de routage, tandis que si la sélection est faite au niveau de consommateur de l'information, comme c'est le cas lors de la sélection de l'information pour un même utilisateur, on parle alors de filtrage. [Sebastiani, 2002] observe qu'une confusion entre filtrage et routage existe chez certains auteurs.

1.5. Difficultés particulières de la catégorisation de textes

L'utilisation des méthodes d'apprentissage automatique afin de traiter les données textuelles est plus difficile que le traitement de données numériques. Le langage naturel (par opposition aux langages informatiques) n'est pas univoque : « *un langage*

univoque (ou plus précisément bi-univoque) est un langage dans lequel chaque mot ou expression a un seul sens, une seule interprétation possible et il n'existe qu'une seule manière d'exprimer un concept donné » [Lefèvre, 2000, page 21]. Le langage naturel est équivoque : il y a plusieurs façons d'exprimer la même idée (la redondance), ce qui est exprimé possède souvent plusieurs interprétations (l'ambiguïté) et tout n'est pas exprimé dans le discours (l'implicite). Ajoutant à ces particularités la grande dimensionalité des descripteurs, et la subjectivité de la décision prise par les experts qui déterminent la catégorie dans laquelle classer un document.

1.5.1. Grandes dimensions

Le modèle vectoriel proposé par [Salton and McGill, 1983] consiste en la représentation de chaque texte par un vecteur dont les composants sont les termes constituant le vocabulaire du corpus. Or, pour un corpus de taille raisonnable, le tableau « termes $\times$ textes » peut avoir des centaines de milliers de lignes (textes) et des milliers de colonnes (termes) ; mais ce tableau est souvent creux, c'est-à-dire que la majorité de ses cellules sont vides. Ceci affecte le processus d'apprentissage en rendant certains algorithmes inopérants et en augmentant le risque de sur-apprentissage.

Complexité de l'algorithme Dans la catégorisation de textes, la grande dimensionalité peut réduire l'efficacité des algorithmes d'apprentissage. En effet, la plupart des algorithmes d'apprentissage sophistiqués, comme les arbres de décision (voir section 4 page 59), le k-PPV (voir section 5 page 63) et la LLSF (pour *Linear Least Squares Fit*) [Yang and Chute, 1992], sont sensibles au $|\mathcal{T}|$, le nombres des variables utilisées pour coder les textes, car $|\mathcal{T}|$ est un paramètre de la complexité de l'algorithme. C'est pourquoi une méthode de réduction de dimension doit être utilisée avant d'estimer les paramètres d'un classifieur.

Sur-apprentissage Le sur-apprentissage se produit lorsqu'un classifieur classe correctement les exemples d'apprentissage (les exemples ayant servi à estimer les paramètres du classifieur), mais classe mal de nouveaux exemples. Expérimentalement, pour éviter le sur-apprentissage, on doit limiter le nombre de descripteurs en fonction du nombre d'exemples dans l'échantillon d'apprentissage. [Fuhr and Buckley, 1991] conseillent d'utiliser au moins 50 à 100 fois plus de textes que de termes. Dans la pratique, comme on dispose d'un nombre limité d'exemples d'apprentissage, on tend à réduire le nombre des termes utilisé pour éviter ce sur-apprentissage.

La réduction de dimension doit cependant être utilisée avec précaution pour ne pas supprimer des termes pertinents [Sebastiani, 2002].

Les techniques utilisées pour la réduction de dimension sont issues de la théorie de l'information et de l'algèbre linéaire. Ces techniques réagissent soit localement, soit globalement. Elles seront présentées à la section 2.4 page 28.

1.5.2. Imprécision des fréquences

Les descripteurs d'un texte, ou d'une classe, étant nombreux, chacun se rencontre rarement dans chaque texte, ou chaque classe. Par conséquent, les cellules du tableau croisé contiennent des nombres petits et aléatoires. On peut modéliser cet aléatoire comme suit : ces nombres suivent approximativement des lois de Poisson ; le coefficient de variation (CV= écart-type/moyenne) donne une indication sur la précision relative de l'estimation de la fréquence dans une cellule du tableau croisé ; pour une loi de Poisson de moyenne *m*, la variance est aussi égale à *m* ; le coefficient de variation est donc : $CV = \sqrt{m}/m = 1/\sqrt{m}$; si *m* est petit, le *CV* est grand et la fréquence est donc imprécise.

1.5.3. Déséquilibrage

Dans la pratique, les effectifs des classes sont souvent déséquilibrés et, pour certaines classes, le nombre d'exemples positifs est faible comparé à celui des exemples négatifs. Ceci crée une difficulté supplémentaire car les classes peu nombreuses sont mal représentées. Une solution proposée pour pallier ce problème est d'utiliser les techniques de redressement.

1.5.4. Ambiguïté

Un même mot ou une même expression peuvent avoir plusieurs sens différents. Les ambiguïtés intrinsèques des mots ou des phrases apparaissent à deux niveaux : lexical et syntaxique. Il faut y rajouter les ambiguïtés de rapport au contexte (voir [Lefèvre, 2000] pour une discussion détaillée). Au niveau lexical, on emploie le terme générique d'homonymie. Des homonymes sont des mots qui ont la même forme, la même graphie, mais des sens différents ; exemples : nous *portions* les *portions* de gâteau, ou bien : le *président* et le directeur *président* la séance.

1.5.5. Synonymie

Il existe de multiples manière d'exprimer la même réalité, avec des nuances diverses. Deux mots ou expressions seront dits synonymes s'ils ont le même sens ; exemples : *mon chat mange un oiseau*, *mon gros matou croque un piaf* et *mon félin préféré dévore une petite bête à plumes* [Lefèvre, 2000, page 24]. On voit bien qu'il s'agit d'un chat qui mange un oiseau mais pourtant les trois textes ne partagent aucun mot autre que des mots-outils (mon, un). La table 1.5.5 montre l'hétérogénéité de l'usage de termes et présente un exemple de quatre dépêches tirées de la collection Reuters-21578 pour la classe « *corporate acquisitions* » où nous pouvons remarquer que les seuls termes communs sont les mots-outils tels que « *it, and, of, for, an, not* » qui sont généralement éliminés lors du prétraitement [Joachims, 2001].

MODULAIRE BUYS BOISE HOMES PROPERTY Modulaire Industries said it acquired the design library and manufacturing rights of privately-owned Boise Homes for an undisclosed amount of cash. Boise Homes sold commercial and residential prefabricated structures, Modulaire said.	USX, CONSOLIDATED NATURAL END TALKS USX Corp's Texas Oil and Gas Corp subsidiary and Consolidated Natural Gas Co have mutually agreed not to pursue further their talks on Consolidated's possible purchase of Apollo Gas Co from Texas Oil. No details were given.
JUSTICE ASKS U.S. DISMISSAL OF TWA FILING The Justice Department told the Transportation Department it supported a request by USAir Group that the DOT dismiss an application by Trans World Airlines Inc for approval to take control of USAir. "Our rationale is that we reviewed the application for controlled by TWA with the DOT and ascertained that it did not contain sufficient information upon which to base a competitive review," James Weiss, an official in Justice's Antitrust Division, told Reuters.	E.D. And F. MAN TO BUY INTO HONG KONG FIRM The U.K. Based commodity house E.D. And F. Man Ltd and Singapore's Yeo Hiap Seng Ltd jointly announced that Man will buy a substantial stake in Yeo's 71.1 pct held unit, Yeo Hiap Seng Enterprises Ltd. Man will develop the locally listed soft drinks manufacturer into a securities and commodities brokerage arm and will rename the firm Man Pacific (Holdings) Ltd.

TAB. 1.1.: Quatre dépêches extraites de Reuters-21578 pour la catégorie « *corporate acquisitions* » qui ne partagent aucun mot commun [Joachims, 2001]

1.5.6. Subjectivité de la décision

Contrairement, à d'autres situations où l'appartenance à une classe est objective (un client achète, ou non, tel produit ; un patient a, ou n'a pas, tel microbe), l'attribution d'une catégorie à un texte est subjective [Uren, 2000]. En effet, la catégorie est attribuée en fonction du contenu sémantique de ce texte, qui est une notion subjective, et dépend du jugement d'un expert. Souvent les experts ne sont pas d'accord sur la classe d'appartenance d'un document [Sebastiani, 1999] ; on parle de « *inter-indexer inconsistency* » [Cleverdon, 1984].

1.6. Lien avec la recherche documentaire

La catégorisation de texte est proche de le recherche documentaire (*Information Retrieval*). En recherche documentaire, on doit retrouver les documents qui correspondent à une requête, ce qui revient à classer tout le corpus en deux classes : les textes correspondant à la requête d'une part, les autres d'autre part. En catégorisation, il s'agit d'attribuer les documents à un ou plusieurs groupes, en fonction des informations qu'ils contiennent.

La recherche documentaire est à l'origine de nombreux modèles de catégorisation de textes. La distinction entre les deux domaines n'est pas facile à établir car ils

peuvent être vus tout deux comme des problèmes de *classement* [Moulinier, 1996].

Le principe de toute recherche documentaire repose sur l'appariement d'une question (requête) avec des documents ou des informations contenue dans une base [Lefèvre, 2000].

[Lewis, 1992b, voir page 3] résume les étapes de la recherche documentaire comme suit :

1. L'indexation de textes : l'opération qui permet de représenter le texte afin qu'il soit exploitable par les système de recherche documentaire.
2. La formulation d'une question, sous diverses formes de requêtes[1] :
   a) un thème ou un descripteur : ex. neutronique ;
   b) une requête, construite avec des mots du langage courant, et utilisant des opérateurs booléens, de proximité, de troncature : ex. (physique and plasma*) or (très near haute* near température*) ;
   c) une expression en langage naturel : ex. vitesse de corrosion des métaux non ferreux en ambiance marine ;
   d) un document entier, utilisé comme exemple du sujet sur lequel on veut obtenir d'autres informations ;
   e) un graphe de concepts ; les concepts, représentés par des termes, peuvent être liés par des relations sémantiques de natures diverses.
3. Comparaison entre requête et documents ; la comparaison se fait habituellement en utilisant une fonction de similarité.
4. Le *Feedback* : les résultats fournis par le système correspondent rarement aux besoins exacts de l'utilisateur. L'utilisateur doit donc revoir la requête et la reformuler. Si le système de recherche modifie ou reformule la requête on parle alors du *relevance feedback*.

Le résultat est souvent imparfait à cause de l'ambiguïté et de la redondance de la langue naturelle [Lefèvre, 2000]. L'ambiguïté se produit car un mot peut posséder plusieurs sens selon le contexte, et la redondance car un même concept peut être exprimé par différents mots.

La catégorisation de texte est étroitement liée à la recherche documentaire. La recherche documentaire intervient en trois phases du cycle de vie d'un classifieur [Sebastiani, 2002] ; voir section 1.3 page 9 :

1. Lors de *l'indexation* des textes.
2. Lors du choix d'une *méthode d'appariement* entre un texte étiqueté et un autre texte, à étiqueter.
3. Lors de l'évaluation de classifieur.

[1] Les exemples présentés sont pris de [Lefèvre, 2000]

1.7. Jeu de données utilisé pour l'évaluation

Les chercheurs en catégorisation de textes, comme dans d'autres domaines expérimentaux, utilisent un ensemble de jeux de données qui aident à valider et à comparer les performances des classifieurs proposés. Ainsi, la collection de Reuters proposée en 1989 a aidé les chercheurs à valider leurs modèles et à comparer les performances avec d'autres modèles.

L'agence Reuters a proposé en 1987 un corpus de dépêches en langue anglaise, disponible gratuitement sur le Web ; ce corpus initial, nommé Reuters-22173 a été étudié notamment par [Lewis, 1992b, Moulinier, 1997]. Depuis, plusieurs versions ont été diffusées. Ces versions se différencient entre elles par les nombres de textes des ensembles d'apprentissage et de test, ainsi que par le nombre des catégories à apprendre. La table 1.2 montre les 5 versions proposées, avec les statistiques concernant chacune d'elles. Les versions Yang, Apte et PARC sont accessibles sur le Web à l'adresse : `http://www-2.cs.cmu.edu/~yiming/`.

Version	Préparée par	# Catég	# Ens. Apprent	# Ens. Test	% Doc étiqutés
Vers 1	CGI	182	21 450	723	80%
Vers 2	Lewis	113	14 704	6 746	42%
Vers 2.2	Yang	113	7 789	3 309	100%
Vers 3	Apte	93	7 789	3 309	100%
Vers 4	PARC	93	9 610	3 662	100%

TAB. 1.2.: Différentes versions de la *collection Reuters*

Le corpus Reuters est souvent utilisé pour les évaluation dans les publications : [Schapire et al., 1998] l'utilise pour comparer l'algorithme AdaBoost avec la formule de Rocchio, et [Joachims, 1998] et [Dumais et al., 1998] pour évaluer les performances des machines à vecteurs supports (SVM).

[Yang and Liu, 1999] ont également utilisé ce corpus pour comparer différents algorithmes (machines à vecteurs supports, réseaux de neurones, arbres de décision, réseaux bayesiens).

Le fait que plusieurs versions de cette collection soient proposées rend difficile la comparaison de modèles entre eux, sauf si l'on utilise une évaluation dite « contrôlée ». Par exemple, une telle évaluation contrôlée a été utilisée par [Yang, 1999] pour évaluer les performances de 13 méthodes d'apprentissages sur les 5 versions de Reuters. D'autres auteurs ont pu évaluer diverses méthodes en les testant sur les mêmes données, avec les mêmes mesures de performances, voir section 5.9 page 79.

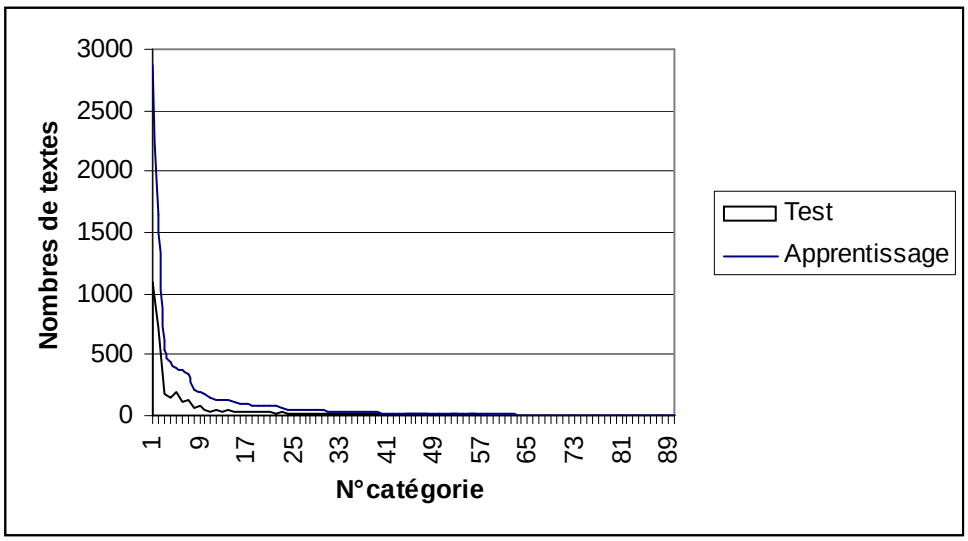

FIG. 1.2.: Nombre de textes par catégories de la collection Reuters. Seules les 20 premières catégories contiennent plus de 100 textes

1.8. Conclusion

Ce chapitre introductif a fixé les objectifs de la catégorisation et a introduit les notions nécessaires à la compréhension des chapitres suivants. Nous avons présenté la notion de catégorisation de textes. Nous avons vu que le processus de catégorisation comporte généralement trois étapes : la représentation, le choix de classifieurs et la méthode d'évaluation du modèle.

La phase de représentation est importante et comporte deux choix qui affectent souvent les performances : le choix de termes (mot, lemme, stemme ou n-grammes) et le choix des poids associés à ces termes (absence/présence, nombre d'occurrences, fréquence, ... *etc.*). Le choix de la méthode d'apprentissage est également primordial ; de nombreuses méthodes sont proposées, chacune possédant des avantages, et des inconvénients. L'évaluation des modèles permet de s'assurer de bonnes performances en généralisation, c'est à dire sur d'autres données, qui n'ont pas été utilisées pour l'apprentissage.

Dans ce chapitre, nous avons montré les applications de la catégorisation de texte. La catégorisation peut être une fin en soi, comme par exemple lors de l'indexation de textes en utilisant les vocabulaires contrôlées ; elle aussi peut être une étape intermédiaire, pour de filtrage ou le routage par exemple.

L'application des algorithmes d'apprentissage aux données textuelles introduit des difficultés supplémentaires. Nous avons évoqué : la grande dimensionalité, la synonymie, la polysémie, la subjectivité de l'attribution d'un texte à une telle ou telle

Num. Catégorie	Catégorie	# Apprentissage	# Test
1	earn	2877	1087
2	acq	1650	719
3	money-fx	538	179
4	grain	433	149
5	crude	389	189
6	trade	369	118
7	interest	347	131
8	wheat	212	71
9	ship	197	89
10	corn	182	56
11	money-supply	140	34
12	dlr	131	44
13	sugar	126	36
14	oilseed	124	47
15	coffee	111	28
16	gnp	101	35
17	gold	94	30
18	veg-oil	87	37
19	soybean	78	33
20	nat-gas	75	30
21	bop	75	30

TAB. 1.3.: Dictionnaire des classes Reuters collection 21578

catégorie.

La catégorisation de textes est liée à la recherche documentaire. En effet, les techniques utilisées dans la recherche documentaire interviennent dans les trois étapes de construction d'un classifieur.

Les chapitres suivants vont détailler les trois étapes de la classification de textes : représentions des textes pour qu'ils soient traitables par les algorithmes d'apprentissage, choix de classifieurs, évaluation de ces classifieurs.

Chapitre 2

Approches pour la représentation de textes

Sommaire

2.1. Introduction **22**
2.2. Choix de termes **22**
2.2.1. Représentation en « sac de mots » 22
2.2.2. Représentation des textes par des phrases 23
2.2.3. Représentation des textes avec des racines lexicales et des lemmes 24
2.2.4. Méthodes basées sur les n-grammes 24
2.3. Codage des termes **26**
2.3.1. Codage TF $\times$ IDF 27
2.3.2. Codage TFC 27
2.4. Réduction de la dimension **28**
2.4.1. Réduction locale de dimension 29
2.4.2. Réduction globale de dimension 29
2.4.3. Sélection de termes 29
2.4.4. Extraction de termes 30
2.5. Conclusion **30**

2.1. Introduction

Les algorithmes d'apprentissage ne sont pas capables de traiter directement les textes ni, plus généralement, les données non structurées comme les images, les sons et les séquences vidéos. C'est pourquoi une étape préliminaire dite de représentation est nécessaire. Cette étape consiste généralement en la représentation de chaque document par un vecteur, dont les composantes sont par exemple les mots contenus dans le texte, afin de le rendre exploitable par les algorithmes d'apprentissage [Salton and McGill, 1983]. Une collection de textes peut être ainsi représentée par une matrice dont les lignes sont les termes qui apparaissent au moins une fois et les colonnes sont les documents de cette collection. L'entrée w_{kj} est le poids du terme t_k dans le document d_j.

Dans ce chapitre nous allons discuter les méthodes proposées (i) pour le choix de termes, (ii) pour associer des poids à ces termes, et (iii) pour la réduction de dimension.

2.2. Choix de termes

Dans la catégorisation de textes, comme dans la recherche documentaire, on transforme le document d_j en un vecteur $\mathbf{d}_j = (w_{1j}, w_{2j}, ..., w_{|\mathcal{T}|j})$, où $\mathcal{T}$ est l'ensemble de termes (ou descripteurs) qui apparaissent au moins une fois dans le corpus (ou la collection) d'apprentissage. Le poids w_{kj} correspond à la contribution du termes t_k à la sémantique du texte d_j. Notons que la représentation par un vecteur entraîne une perte d'information notamment celle relative à la position de mots dans la phrase[1].

2.2.1. Représentation en « sac de mots »

La représentation de textes la plus simple a été introduite dans le cadre du modèle vectoriel présenté ci-dessus, et porte le nom de « sac de mots ». L'idée est de transformer les textes en vecteurs dont chaque composante représente un mot. Les mots ont l'avantage de posséder un sens explicite. Cependant, plusieurs problèmes se posent. Il faut tout d'abord définir ce qu'est « un mot » pour pouvoir le traiter automatiquement. On peut le considérer comme étant une suite de caractères appartenant à un dictionnaire, ou bien, de façon plus pratique, comme étant une séquence de caractères non-délimiteurs encadrés par des caractères délimiteurs (caractères de ponctuation) [Gilli, 1988] ; il faut alors gérer les sigles, ainsi que les mots composés ; ceci nécessite un pré-traitement linguistique. On peut choisir de conserver les majuscules pour aider, par exemple, à la reconnaissance de noms propres, mais il faut alors résoudre le problème des débuts de phrases.

[1] D'autres méthodes d'apprentissage utilisent cette information ; par exemple les modèles de Markov cachés.

Les composantes du vecteur sont une fonction de l'occurrence des mots dans le texte. Cette représentation des textes exclut toute analyse grammaticale et toute notion de distance entre les mots : c'est pourquoi cette représentation est appelée « sac de mots » (voir la figure 2.1) ; d'autres auteurs parlent d'« ensemble de mots » lorsque les poids associés sont binaires.

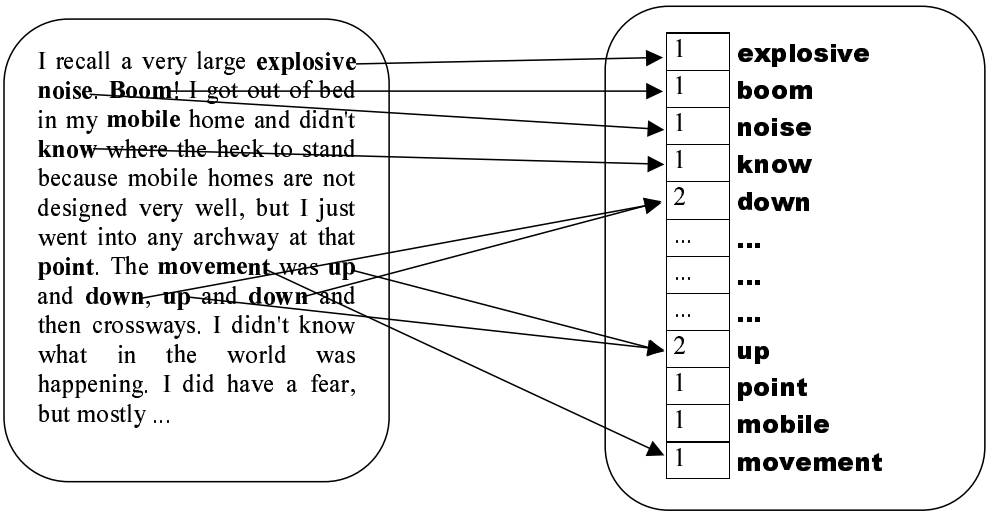

FIG. 2.1.: Exemple de la représentation d'un texte extrait de la collection CLEF en sac de mots (voir la section 9.3 page 120 pour une représentation de cette collection)

Un grand nombre d'auteurs comme [Lewis, 1992b, Apté et al., 1994, Dumais et al., 1998, Aas and Eikvil, 1999] utilisent les mots comme termes (c'est à dire comme composantes du vecteur) pour représenter les textes.

2.2.2. Représentation des textes par des phrases

Malgré la simplicité de l'utilisation de mots comme unité de représentation, certains auteurs proposent plutôt d'utiliser les phrases comme unité [Fuhr and Buckley, 1991, Schütze et al., 1995, Tzeras and Hartmann, 1993]. Les phrases sont plus informatives que les mots seuls, par exemple : « *machine learning* » ou « *world wide web* » car les phrases ont l'avantage de conserver l'information relative à la position du mot dans la phrase (*the problem of compositional semantics*). Logiquement, une telle représentation doit obtenir de meilleurs résultats que ceux obtenus via les mots. Mais les expériences présentées ne sont pas concluantes car, si les qualités sémantiques sont conservées, les qualités statistiques sont largement dégradées : le grand nombre de combinaisons possibles entraîne des fréquences faibles et trop aléatoires [Lewis, 1992b]. Néanmoins, [Caropreso et al., 2001] proposent d'uti-

liser les phrases statistiques comme unités de représentation en opposition aux phrases « grammaticales » et obtiennent de bons résultats. Une phrase statistique est un ensemble de mots contigus (mais pas nécessairement ordonnés) qui apparaissent ensembles mais qui ne respectent pas forcement les règles grammaticales. Afin de déterminer les phrases statistiques, [Caropreso et al., 2001] utilisent des prétraitements tels que l'élimination des mots outils (*stop words*) et le *stemming*.

2.2.3. Représentation des textes avec des racines lexicales et des lemmes

Dans le modèle précédent (représentation en « sac de mots »), chaque flexion d'un mot est considérée comme un descripteur différent et donc une dimension de plus ; ainsi, les différentes formes d'un verbe constituent autant de mots. Par exemple : les mots « déménageur, déménageurs, déménagement, déménagements, déménager, déménage, déménagera, etc. » sont considérés comme des descripteurs différents alors qu'il s'agit de la même racine « déménage » ; les techniques de *désuffixation (*ou *stemming*), qui consistent à rechercher les racines lexicales, et de *lemmatisation* cherchent à résoudre cette difficulté.

Pour la recherche des racines lexicales, plusieurs algorithmes ont été proposés ; l'un des plus connus pour la langue anglaise est l'algorithme de Porter [Porter, 1980] ; une comparaison entre différents algorithmes de recherche de racines lexicales a été menée dans [Hull, 1996].

La lemmatisation consiste à remplacer les verbes par leur forme infinitive, et les noms par leur forme au singulier. Un algorithme efficace, nommé TreeTagger [Schmid, 1994], a été développé pour les langues anglaise, française, allemande et italienne.

L'extraction des *stemmes* repose, quant à elle, sur des contraintes linguistiques bien moins fortes ; elle se base sur la morphologie flexionnelle mais aussi dérivationnelle [de Loupy, 2001]. De ce fait, les algorithmes sont beaucoup plus simplistes et mécaniques que ceux permettant l'extraction des lemmes ; ils sont donc plus rapides ; mais leur précision et leur qualité sont naturellement inférieures.

2.2.4. Méthodes basées sur les n-grammes

Principe

Un n-gramme est une séquence de n caractères. Dans ce manuscrit un *n-gramme* désignera une chaîne de n caractères consécutifs. Dans la littérature, ce terme désigne quelquefois des séquences qui ne sont ni ordonnées ni consécutives ; par exemple un 2-gramme peut être composé de la première lettre et de la troisième lettre d'un mot [Cavnar and Trenkle, 1994] ; [Caropreso et al., 2001] considèrent un n-grammes comme un ensemble non ordonné de *n* mots après avoir effectué la désuffixation (ou *stemming*) et la suppression de mots vides. Ce n'est pas l'acceptation utilisée ici.

Pour un document quelconque, l'ensemble des n-grammes est le résultat obtenu en déplaçant une fenêtre de n cases sur le texte ; ce déplacement se fait par étapes de un caractère et à chaque étape on prend une photo ; l'ensemble de ces photos donne l'ensemble de tous les n-grammes du document. Par exemple, pour générer tous les 5-grammes dans la phrase "*Je suis un génie*", on obtient : *je_su*, *e_sui*, *suis_*, *_suis*, *uis_u*, etc. [Miller et al., 1999]. Le profil n-grammes d'un document est la liste des n caractères les plus fréquents, par ordre décroissant de leur fréquence d'apparition dans le document, ainsi que leurs fréquences elles-mêmes ; un document est caractérisé par son profil n-grammes. Les profils s'obtiennent en temps linéaire grâce à des tables de hachage.

La notion de n-grammes a été introduite par [Shannon, 1948] en 1948 ; il s'intéressait à la prédiction d'apparition de certains caractères en fonction des autres caractères. Depuis cette date, les n-grammes sont utilisés dans plusieurs domaines comme l'identification de la parole, la recherche documentaire, *etc.*

Avantages

Il y a plusieurs avantages à l'utilisation des n-grammes :

– Premièrement, les n-grammes capturent les connaissances des mots les plus fréquents ; le tableau 2.1 montre quelques suffixes et mots grammaticaux extraits en utilisant ces techniques.

Dan	Dut	Eng	Fre	Ger	Ita	Nor	Por	Spa	Swe
er_	en_	_th	de_	en_	_di	et_	_de	_de	en_
en_	de_	he_	es_	er_	to_	_._	de_	de_	_._
for	_de	the	de_	_de	_de	en_	os_	os_	er_
et_	et_	_,_	ent	der	di_	er_	do_	_la	et_
ing	an_	nd_	nt_	ie_	_co	_de	que	el_	tt_
_fo	n_d	ed_	_le	ich	la_	_ha	_qu	la_	_de
_af	_he	_an	e_d	sch	re_	an_	_co	que_	ar_
_de	er_	and	le_	ein	ion	de_	as_	as_	_,_
nde	_va	_._	ion	che	ent	_,_	ent	ue_	fr
els	van	_to	s_d	die	e_d	det	o_	_qu	om_
lse	een	ing	e_l	ch_	le_	ar_	ue_	_co	_oc
ret	ver	to_	_la	den	o_d	_og	_a_	_en	ch_
_sa	aar	ng_	la_	nd_	ne_	og_	o_d	en_	de_
der	_ee	er_	re_	_di	no_	te_	_se	ent	och
_i_	het	_of	on_	ung	_in	han	_o_	es_	an_

TAB. 2.1.: Trigrammes les plus fréquents dans chacune de dix langues européennes, d'après le « ECI Multilingual Corpus »

– Deuxièmement, les n-grammes opèrent indépendamment des langues, alors que les systèmes basés sur les mots sont dépendants des langues ; par exemple, les traitements d'élimination des "mots vides", de "recherche de racine" et de lemmatisation (voir [Sahami, 1999] pour une description de ces techniques) sont spécifiques à chaque langue. De même, la plupart des techniques de n-grammes n'exigent pas une segmentation préalable du texte en mots ; ceci est intéressant pour le traitement des langues où les frontières entre mots ne sont pas fortement marquées comme le chinois, l'arabe et l'allemand, ou encore les séquences ADN (qui peuvent être considérées comme de textes d'un alphabet de quatre lettres). Par exemple, pour l'allemand « lebensversicherungsgesellschaftsangestellter » ("employé d'une compagnie d'assurance vie"), ou pour la langue arabe, dans laquelle les pronoms sujets et compléments sont dans certains cas attachés aux verbes, une seule chaîne de caractères représentant une phrase comme, par exemple, kathabthouhou ("je l'ai écrit") ; dans ces deux cas la notion de « tokens » devient carrément inadéquate [Manning and Schütze, 1999, Biskri and Delisle, 2001].
– Troisièmement, elles sont tolérantes aux déformations liées à l'utilisation de systèmes de reconnaissance optique de caractères (OCR) et tolérantes aux fautes d'orthographe. Lorsqu'un document est scanné en utilisant un OCR, la reconnaissance optique est souvent imparfaite. Par exemple, le mot "character" peut être lu comme "claracter". Un système fondé sur les mots reconnaîtra difficilement le mot "character", ou sa même racine. Par contre, un système basé sur les n-grammes sera capable de prendre en compte les autres n-grammes (ici, n = 5) comme "aract", "racte", *etc*. [Miller et al., 1999] montre que certains systèmes de recherches documentaires fondés sur les n-grammes ont conservé leurs performances avec des taux de déformations de 30%, un taux avec lequel aucun système fondé sur les mots ne peut fonctionner correctement.
– Finalement, ces techniques n'ont pas besoin de procéder à l'élimination ni des "mots outils" (ou "mots vides"), ni au "Stemming", ni à la lemmatisation, qui améliorent la performance des systèmes basés sur les mots. Pour les systèmes n-grammes, de nombreuses études ont montré que la performance ne s'améliore pas lorsque on applique des traitements d'éliminations des "mots vides", de "Stemming" ou de lemmatisation. A titre d'exemple, si un document contient plusieurs mots de même racine, les fréquences des n-grammes correspondants augmenteront sans avoir besoin d'aucun traitement linguistique préalable.

2.3. Codage des termes

Une fois choisies les composantes du vecteur représentant un texte *j*, il faut décider comment coder chaque coordonnée de son vecteur $\mathbf{d}_j$.

Il existe différentes méthodes pour calculer le poids w_{kj}. Ces méthodes sont basées sur les deux observations suivantes :

1. Plus le terme t_k est fréquent dans un *document* d_j, plus il est *en rapport* avec le sujet de ce document.
2. Plus le terme t_k est fréquent dans une *collection*, moins il sera utilisé comme *discriminant* entre documents.

Soient $\#(t_k, d_j)$ le nombre d'occurrences du terme t_k dans le texte d_j, $|Tr|$ le nombre de documents du corpus d'apprentissage et $\#Tr(t_k)$ le nombre de documents de cet ensemble dans lesquels apparaît au moins une fois le terme t_k. Selon les deux observations précédentes, un terme t_k se voit donc attribuer un poids d'autant plus fort qu'il apparaît souvent dans le document et rarement dans le corpus complet. La composante du vecteur est codée $f(\#(t_k, d_j))$, où la fonction f reste à déterminer. Deux approches triviales peuvent être utilisées. La première consiste à attribuer un poids égal à la fréquence du terme dans le document :

$$w_{kj} = \#(t_k, d_j) \tag{2.1}$$

et la deuxième approche consiste à associer une valeur booléenne :

$$w_{kj} = \begin{cases} 1 & \text{Si } \#(t_k, d_j) > 1 \\ 0 & \text{Sinon} \end{cases} \tag{2.2}$$

2.3.1. Codage TF $\times$ IDF

Les deux fonctions 2.1 et 2.2 précédentes sont rarement utilisées car ces codages appauvrissent l'information : la fonction 2.2 ne prend pas en compte la fréquence d'apparition du terme dans le texte, fréquence qui peut constituer un élément de décision important ; la fonction 2.1 ne prend pas en compte la fréquence du terme dans les autres textes.

Le codage TF $\times$ IDF a été introduit dans le cadre du modèle vectoriel et utilise une fonction de l'occurrence multipliée par une fonction de l'inverse du nombre de documents différents dans lequel un terme apparaît. Ce sigle provient de l'anglais et signifie « *'term frequency'*$\times$*'inverse document frequency*' ».

Les termes caractérisant une classe apparaissent plusieurs fois dans les document de cette classe, et moins, ou pas du tout, dans les autres. C'est pourquoi le codage TF $\times$ IDF [Sebastiani, 2002] est défini comme suit :

$$\text{TF} \times \text{IDF}(t_k, d_j) = \#(t_k, d_j) * \log \frac{|Tr|}{\#Tr(t_k)} \tag{2.3}$$

2.3.2. Codage TFC

Le codage TF $\times$ IDF ne corrige pas la longueur des documents. Pour ce faire, le codage TFC est similaire à celui de TF $\times$ IDF mais il corrige les longueurs des textes

par la normalisation en cosinus, afin de ne pas favoriser les documents les plus longs.

$$\text{TFC}(t_k, d_j) = \frac{\text{TF} \times \text{IDF}(t_k, d_j)}{\sqrt{\sum_{s=1}^{|\tau|} (\text{TF} \times \text{IDF}(t_s d_j))^2}}$$

D'autres codage sont également utilisés, comme par exemple le codage LTC [Buckley et al., 1994] qui tente de réduire les effets des différences de fréquences, ou encore le codage à base d'entropie. [Dumais, 1991] affirme obtenir de meilleurs résultats avec un codage basé sur l'entropie [Aas and Eikvil, 1999].

2.4. Réduction de la dimension

Un problème central pour l'approche statistique de la catégorisation de textes est la grande dimension de l'espace de représentation. Avec la représentation en sac de mots, chacun des mots d'un corpus est un descripteur potentiel ; or pour un corpus de taille raisonnable, ce nombre peut être de plusieurs dizaines de milliers. Pour beaucoup d'algorithmes d'apprentissage, il faut sélectionner un sous-ensemble de ces descripteurs. Sinon, deux problèmes se posent :

- le coût du traitement car le nombre des termes intervient dans l'expression de la complexité de l'algorithme ; plus ce nombre est élevé, plus le volume de calcul est important ;
- la faible fréquence de certains termes : on ne peut pas construire des règles fiables à partir de quelques occurrences dans l'ensemble d'apprentissage.

On a observé que les mots les plus fréquents peuvent être supprimés : ils n'apportent pas d'information sur la catégorie d'un texte puisqu'ils sont présents partout. De même, les mots très rares, qui n'apparaissent qu'une ou deux fois sur un corpus, sont supprimés, car leurs faibles fréquences ne permettent pas de construire de règles stables.

Cependant, même après la suppression de ces deux catégories de mots, le nombre de candidats reste encore élevé, et il faut utiliser une méthode statistique pour choisir les mots utiles pour discriminer entre documents pertinents et documents non pertinents, ou, plus généralement, entre les classes de documents.

Les techniques utilisées pour la réduction de dimension sont issues de la théorie de l'information et de l'algèbre linéaire. [Sebastiani, 2002] classe ces techniques de deux façons : *i*) selon qu'elles agissent localement ou globalement, et *ii*) selon la nature des résultats de la sélection (s'agit-il d'une sélection de termes ou d'une extraction de termes).

2.4.1. Réduction locale de dimension

Il s'agit de proposer, pour chaque catégorie c_i un nouvel ensemble de terme $\mathcal{T}_i'$ avec $|\mathcal{T}_i'| \ll |\mathcal{T}_i|$ [Apté et al., 1994, Lewis and Ringuette, 1994, Schütze et al., 1995, Wiener et al., 1995, Ng et al., 1997, Li and Jain, 1998, Sable and Hatzivassiloglou, 2000]. Ainsi, chaque catégorie c_i possède son propre ensemble de termes et chaque document d_j sera représenté par un ensemble de vecteurs $\mathbf{d}_j$ différents selon la catégorie. Habituellement, $10 \leq \mathcal{T}_i' \leq 50$.

2.4.2. Réduction globale de dimension

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

Notons que toutes les techniques de réduction de termes peuvent être appliquées soit localement soit globalement.

2.4.3. Sélection de termes

Les techniques de réduction de dimensions par sélection visent à proposer un nouvel ensemble $\mathcal{T}'$ avec $|\mathcal{T}'| \ll |\mathcal{T}|$. Parmi ces techniques figurent le calcul de l'information mutuelle [Lewis, 1992a, Moulinier, 1997, Dumais et al., 1998], la méthode du χ^2_{max} [Schütze et al., 1995], ou des méthodes plus simples utilisant uniquement les fréquences d'apparitions [Wiener et al., 1995, Yang and Pedersen, 1997] ; d'autres méthodes ont également été testées [Moulinier, 1996, Sahami, 1999] (voir la table 2.2).

TR	Représentée par	Le forme mathématique
le gain d'information	$IG(t_k, c_i)$	$\sum_{c\in\{c_i,\bar{c}_i\}} \sum_{t\in\{t_k,\bar{t}_k\}} P(t,c).\log\frac{P(t,c)}{P(t).P(c)}$
l'information mutuelle	$MI(t_k, c_i)$	$\log\frac{P(t_k,c_i)}{P(t_k).P(c_i)}$
χ^2_{uni}	$\chi^2_{\text{uni}}(t_k, c_i)$	$\frac{\lvert Tr\rvert[P(t_k,c_i)P(\bar{t}_k,\bar{c}_i)-P(t_k,\bar{c}_i).P(\bar{t}_k,c_i)]^2}{P(t_k).P(\bar{t}_k).P(c_i).P(\bar{c}_i)}$
Odds ratio	$OR(t_k, c_i)$	$\frac{P(t_k\lvert c_i).(1-P(t_k\lvert\bar{c}_i))}{(1-P(t_k\lvert c_i)).P(t_k\lvert\bar{c}_i)}$

TAB. 2.2.: Techniques de réductions (TR) de dimensions souvent utilisées dans le domaine de la catégorisation de textes

Une comparaison de l'information mutuelle et de la méthode du χ^2_{uni} avec d'autres méthodes est présentée dans [Yang and Pedersen, 1997][2] ; il semble que l'informa-

[2] [Yang and Pedersen, 1997], dans leur article, utilisent le $\chi^2_{\text{max}} = \max_i \chi^2_{\text{uni}}(t_k, c_i)$.

tion mutuelle soit légèrement supérieure aux autres[3], mais ces méthodes de réduction n'augmentent la performance que de l'ordre de 5%, selon le classifieur et selon le degré d'agressivité de réduction $\frac{|\mathcal{T}|}{|\mathcal{T}'|}$.

2.4.4. Extraction de termes

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

L'une des approches est appelée le « Latent Semantic Indexing (LSI) », proposée par [Deerwester et al., 1990]. Le LSI est fondé sur l'hypothèse d'une structure latente des termes, identifiable par les techniques factorielles. Il consiste en une décomposition en valeurs singulières de la matrice dans laquelle chaque document est représenté par la colonne des occurrences des termes qui le composent. D'autres approches sont également proposées et utilisées pour la réduction de dimensions comme « *term clustering* » testé par [Lewis, 1992a] (voir [Sebastiani, 2002] pour une présentation détaillée de ces approches). Le LSI est très proche de l'Analyse des Correspondances [Morin, 2002], introduite par [Benzecri, 1976] et [Escofier, 1965] pour traiter les données textuelles ; une utilisation systématique de l'analyse factorielle des correspondances pour les données textuelles se trouve dans [Lebart and Salem, 1994].

2.5. Conclusion

L'objectif des méthodes de réduction de termes est de fournir une liste de termes plus courte mais porteuse d'information. Les termes sont en général ordonnée du terme le plus important au moins important selon un certain critère. La question se pose du nombre de termes à conserver dans cette liste [Stricker, 2000].

Ce nombre dépend souvent du modèle, puisque, par exemple, les machines à vecteurs supports sont capables de manipuler des vecteurs de grandes dimensions alors que, pour les réseaux de neurones, il est préférable de limiter la dimension des vecteurs d'entrées. Pour choisir le bon nombre de descripteurs, il faut déterminer si l'information apportée par les descripteurs en fin de liste est utile, ou si elle est redondante avec l'information apportée par les descripteurs du début de la liste. Dans son utilisation des machines à vecteurs supports, [Joachims, 1998] considère l'ensemble des termes du corpus Reuters, après suppression des mots les plus fréquents et l'utilisation de racines lexicales (les stemmes). Il reste alors 9.962 termes distincts qui sont

[3] L'équivalence approximative entre les deux mesures a été prouvée dans les années 60 par Benzecri, voir [Chauchat, 1975].

utilisés pour représenter les textes en entrée de son modèle. Il considère que chacun de ces termes apporte de l'information, et qu'il est indispensable de les inclure tous.

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

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

Chapitre 3

Sélection multivariée de termes

Sommaire

3.1. Introduction **34**
3.2. Méthode du χ^2 univariée **34**
3.3. Méthode du χ^2 multivarié **36**
3.4. Expérimentation **36**
3.5. Conclusion **38**

3.1. Introduction

Comme il n'existe pas à l'heure actuelle de méthodes statistiques capables de traiter directement des données non-structurées, il est nécessaire de transformer les données en une représentation tabulaire individus-variables propice à l'apprentissage ; l'individu est un texte étiqueté, et les variables sont des termes extraits du corpus textuel ; un terme étant un mot, ou une séquence de n caractères consécutifs (n-grammes). Un des principaux enjeux de la catégorisation de textes est de sélectionner, parmi ces termes un sous ensemble optimal, c'est-à-dire qui assurera les meilleures performances au modèle de prédiction.

Cette sélection s'inscrit dans un cadre particulier, car le nombre de termes, après le pré-traitement du corpus, est potentiellement très élevé ($\mathcal{T}$ variables, avec très souvent $|\mathcal{T}| > 10000$). Les méthodes couramment utilisées pour la sélection de variables en Data Mining sont, certe, performantes mais de complexité élevée [Liu and Motoda, 1998] ; elles sont donc peu appropriées ici. De fait, les méthodes de sélection de termes les plus souvent mises en oeuvre en catégorisation de textes sont généralement univariées afin de rester de complexité $O(|\mathcal{T}|)$. Ces méthodes, qui semblent donner satisfaction dans certains contextes, notamment lorsque l'on cherche la présence ou absence d'un sujet, présentent néanmoins deux défauts : elles ne tiennent pas compte des corrélations entre les termes, car le rôle de chaque terme est évalué indépendamment des autres ; et, dans le cas où l'étiquette à prédire peut prendre plusieurs modalités, elles ne permettent pas de déterminer, lorsqu'un terme est significatif, à quelle association "catégorie"-"terme" est dû sa sélection, et par là, à la reconnaissance de quelle catégorie il contribue le plus.

Dans ce chapitre, nous présentons une nouvelle méthode pour la sélection de termes lors de la catégorisation de textes. Cette méthode s'appuie sur la contribution du χ^2 d'indépendance et se démarque des méthodes univariées par la nature de l'information utilisée : elle est fondée sur la fréquence des termes, et non leur présence/absence ; elle tient compte des interactions entre les termes et des interactions termes/catégories. Malgré sa sophistication, elle reste de complexité linéaire en $|\mathcal{T}|$ (nombre de termes).

Dans la section 3.2, nous rappelons la méthode de sélection de termes univariée χ^2_{uni}(usuelle en catégorisation de textes). Dans la section 3.3, nous décrivons notre méthode de sélection multivariée. Puis, dans la section 3.4, nous présentons une expérimentation qui permet d'évaluer la pertinence de notre approche par rapport à une méthode univariée de référence. Nous conclurons dans la section 3.5, en indiquant les améliorations possibles.

3.2. Méthode du χ^2 univariée

La statistique du χ^2_{uni} mesure l'écart à l'indépendance entre un descripteur t_k (présent ou absent) et un thème c_i (présent ou absent) ; elle est donc calculée sur un

	terme t_k présent	**terme t_k absent**	
Thème c_i présent	a	c	$a+c$
Thème c_i absent	b	d	$b+d$
	$a+b$	$c+d$	$N=a+b+c+d$

TAB. 3.1.: Tableau de contingences pour l'absence ou la présence d'un descripteur dans les documents d'une classe

tableau 2×2. Cette mesure a été utilisée pour la sélection des descripteurs dans [Schütze et al., 1995, Wiener et al., 1995, Yang and Pedersen, 1997, He et al., 2000]. Le calcul nécessite de construire le tableau de contingence (2×2) pour chaque descripteur t_k du corpus et pour chaque classe c_i (voir table 3.1). Dans ce tableau, on compte les documents ; par exemple, dans la première cellule, a est le nombre de documents de la classe c_i dans lesquels le terme t_k est présent.

Dans le cas d'un tableau de contingence (2×2), la statistique du χ^2 peut se mettre sous la forme

$$\chi^2_{\text{uni}}(t_k, c_i) = \frac{N(ad - cb)^2}{(a+c)(b+d)(a+b)(c+d)} \tag{3.1}$$

Si un descripteur t_k et le thème c_i sont totalement indépendants dans le corpus disponible, alors t_k apparaît avec la même fréquence relative dans le sous-ensemble des textes pertinents et dans celui des textes non pertinents, ce qui se traduit par $(ad = bc)$ et la valeur $\chi^2(t_k, c_i)$ est nulle. A l'inverse, si le descripteur t_k apparaît dans tous les textes pertinents et jamais dans l'ensemble des textes non pertinents, on a $c = b = 0$ et $\chi^2(t_k, c_i)$ vaut N, ce qui est sa valeur maximale. Cette valeur est également atteinte si un descripteur apparaît dans tous les textes non pertinents et jamais dans l'ensemble des textes pertinents.

Entre ces deux valeurs extrêmes, plus la valeur de $\chi^2(t_k, c_i)$ est grande, plus t_k et c_i sont liés. Les descripteurs du corpus sont ensuite classés par ordre décroissant de $\chi^2(t_k, c_i)$, les plus discriminants figurant en tête de liste [Yang and Pedersen, 1997].

La formule 3.1 est calculée pour tous les couples (t_k, c_i). On a proposé de résumer la valeur discriminante globale de chaque terme (t_k) par deux mesures :

$$\chi^2_{\max}(t_k) = \max_i \chi^2_{\text{uni}}(t_k, c_i) \tag{3.2}$$

$$\chi^2(t_k) = \sum_{i=1}^{|C|} \chi^2_{\text{uni}}(t_k, c_i) \tag{3.3}$$

Pour les comparaisons ultérieures, à l'instar [Yang and Pedersen, 1997] et [Wiener et al., 1995], nous utiliserons le $\chi^2_{\max}(t_k)$ qui maximise le lien entre le terme

t_k et un attribut à prédire c_i ; on sélectionne ensuite les termes qui sont les plus liés à au moins une classe.

Algorithme 1 Méthode du χ^2 multivarié

pour chaque classe $c_i \in \mathcal{C}$ **faire**
  rechercher tous les termes dans tous les textes d'apprentissage
fin pour
constituer le tableau des N_{ki} nombre d'occurrences du terme k dans la classe i
calculer les fréquences f_{ki} relatives : $f_{ki} = \frac{N_{ki}}{N}$
calculer les contributions des cellules (ki) à la statistique du χ^2_{multi} : $\chi^2_{ki} = \frac{\left(N_{ki} - \frac{N_{k.} \times N_{.i}}{N}\right)^2}{\frac{N_{k.} \times N_{.i}}{N}} = N \times \frac{(f_{ki} - f_{k.} \times f_{.i})^2}{f_{k.} \times f_{.i}}$
calculer le $\chi^2_{ki} \times signe(f_{ki} - f_{k.} \times f_{.i})$
trier le tableau des χ^2_{ki} dans l'ordre décroisant
pour chaque classe i **faire**
  déterminer la liste $\{terme_{ki}\}$ des K premiers termes de la classe (K est un paramètre de l'algorithme).
fin pour

3.3. Méthode du χ^2 multivarié

Nous présentons ici une nouvelle méthode de sélection de termes pour la catégorisation de textes : le χ^2 multivarié (noté χ^2_{multi}). C'est une méthode supervisée permettant la sélection de termes en prenant en compte, non seulement leurs fréquences dans chaque classe, mais aussi l'interaction des termes entre-eux et les interactions termes/classes. L'idée consiste à utiliser les contributions des cellules (t_k, c_i) au χ^2 associé au tableau croisé global, où N_{ki} est le nombre de fois où le terme t_k est présent dans les documents de la classe c_i. Les étapes sont décrites dans l'algorithme 1.

Les principales caractéristiques de cette méthode sont les suivantes : elle est supervisée car elle s'appuie sur l'information apportée par les catégories $\mathcal{C}$; elle est multivariée car elle évalue globalement le rôle d'un terme par rapport aux autres ; elle tient compte de l'interaction termes/classes car elle permet de choisir, pour chaque catégorie, les termes qui contribuent le plus à leur discrimination.

3.4. Expérimentation

Afin d'illustrer la pertinence de notre approche, nous comparons les résultats des sélections par χ^2_{uni}(avec χ^2_{max}) et par χ^2_{multi}. Il est à noter que la nature de l'information

		100 mots	**200 3-grammes**	**200 4-grammes**	**200 5-grammes**
Taux de succès	χ^2_{multi}	95%	94%	95%	94%
	χ^2_{max}	93%	92%	91%	89%
Rappel moyen	χ^2_{multi}	95%	93%	96%	95%
	χ^2_{max}	94%	91%	92%	90%
Précision moyenne	χ^2_{multi}	94%	94%	95%	93%
	χ^2_{max}	94%	92%	91%	88%

TAB. 3.2.: Qualité du modèle (en 10 validation croisées) selon la nature des termes utilisés et la méthode de sélection des termes

utilisée est très différente dans les deux cas. En effet, dans le cadre multivarié, l'information utilisée est le nombre d'occurrences des termes (la marge totale équivaut à la somme du nombre d'occurrences des termes sur tout le corpus), tandis que dans le cadre univarié, l'information est le nombre de documents où un terme apparaît (la marge totale est le nombre de documents).

L'expérience a été réalisée sur la catégorisation des dépêches du journal *Le Monde* de 1994. Le corpus *Le Monde* est constitué de 419 dépêches, divisées en 10 catégories. Les dépêches sont inégalement réparties entre les catégories ; leur nombre varie de 19 à 95. Les catégories sont des sujets définis par des experts (par exemple Téléphone Portable, Conflit en Palestine). Une dépêche n'est affectée qu'à une et à une seule catégorie.

La méthode d'apprentissage utilisée est usuelle dans le cadre de la catégorisation : le 3 Plus Proches Voisins (3-PVV) [Mitchell, 1997]. Ce choix est également motivé par la sensibilité de cette méthode à la qualité de l'espace de représentation, c'est-à-dire aux termes sélectionnés pour l'apprentissage. Nous avons paramétré cette méthode en utilisant les votes pondérés par l'inverse de leur distance et la métrique cosinus [Amini, 2001]. Les valeurs des termes sélectionnés sont pondérés par le TF$\times$IDF ($Term\ Frequency \times Inverse\ Document\ Frequency$), là encore largement utilisé dans les problèmes de catégorisation (voir la section 2.3.1 page 27).

Notre chaîne de traitements consiste donc en trois étapes : 1) sélection de termes (multivariée ou univariée), 2) pondération des termes par le TF$\times$IDF, 3) apprentissage par la méthode des 3-PPV. Par validation croisée, nous mesurons alors plusieurs indicateurs d'évaluation de la catégorisation de textes (tableau 3.2) : le taux de succès, le rappel macro-moyen et la précision macro-moyen (moyenne des rappels (respectivement des précisions) des catégories) [Sebastiani, 2002].

Les résultats (tableau 3.2) montrent que la méthode du χ^2_{multi} a une meilleure qualité de prédiction, quel que soit l'indicateur de qualité utilisé pour évaluer l'apprentissage, et ceci pour chaque type de termes (mot, $3-4-5$ grammes).

3.5. Conclusion

Dans ce chapitre, nous avons proposé une nouvelle méthode (χ^2_{multi}) pour la sélection de termes lors de la catégorisation de textes. Cette méthode, de complexité linéaire, s'appuie sur la contribution des cellules (t_k, c_i) au χ^2 associé au tableau croisé global, où N_{ki} est le nombre de fois où le terme t_k est présent dans les documents de la classe c_i. Elle se démarque des méthodes univariées usuelles par la nature de l'information utilisée, elle est fondée sur la fréquence des termes, et non leur présence/absence ; elle tient compte, de plus, des interactions entre les termes et des interactions termes/catégories.

Nos premières évaluations indiquent que l'approche est efficace ; d'autres expérimentations permettront de mieux discerner les avantages/inconvénients de l'algorithme. Cela nous permettra par la suite de caractériser les situations où la méthode de sélection χ^2_{multi} est la plus efficace.

Enfin, l'algorithme présenté ici repose sur un paramètre, le nombre de termes à sélectionner pour chaque catégorie, qui doit être fixé par l'utilisateur. On pourrait chercher à déterminer automatiquement les limites de cette liste de termes (les plus pertinents pour la discrimination) par des considérations statistiques sur la "significativité" des contributions.

Chapitre 4

Pourquoi les n-grammes fonctionnent

Sommaire

4.1. Introduction 40
4.2. Intérêt du codage en n-grammes 40
4.3. Étapes de la recherche des mots caractéristiques 41
4.3.1. Recherche des n-grammes caractéristiques et des mots qui les contiennent 41
4.3.2. Filtrage des mots « parasites » 42
4.3.3. Algorithme complet 42
4.4. Exemple d'application 42
4.4.1. Données indexées de Reuters 42
4.4.2. Quelques résultats 44
4.4.3. Discussion des résultats sur la collection Reuters 44
4.5. Conclusion 46

4.1. Introduction

De nombreux travaux ont montré l'efficacité des n-grammes comme méthode de représentation des textes pour leur catégorisation : attribution d'un texte à une, ou plusieurs, catégorie(s) parmi une liste pré-déterminée [Dunning, 1994, Cavnar and Trenkle, 1994, Damashek, 1995, Miller et al., 1999, Biskri and Delisle, 2001, Teytaud and Jalam, 2001].

Un premier objectif de ce chapitre est de montrer pourquoi la représentation en n-grammes est efficace. On rétablit le lien entre l'aspect purement formel du texte et son sens ; on passe des n-grammes, caractéristiques d'une classe de textes, aux mots contenant ces n-grammes dans ces textes.

Un deuxième objectif est la recherche automatique de nouveaux candidats "mots-clés", c'est-à-dire une liste de mots statistiquement caractéristiques d'une classe de textes, parmi lesquels l'utilisateur pourra choisir ceux qu'il retiendra. Notre travail se situe donc en amont de celui de [Lelu and Hallab, 2000] qui ont proposé, pour le même objectif, une méthode interactive pour sélectionner des mots ou des groupes de mots.

La section 4.3 décrit les étapes de la recherche des mots statistiquement caractéristiques ; la section suivante décrit deux applications sur de grands corpus de données réelles ; la dernière section conclue et propose des pistes pour la poursuite de ce travail.

Tout d'abord, nous rappelons le principe du codage en n-grammes, puis ses qualités.

4.2. Intérêt du codage en n-grammes

Un n-gramme est une séquence de *n* caractères consécutifs. Dans un texte, on repère tous les n-grammes présents, puis on compte leurs fréquences. Par exemple la phrase "*La nourrice nourrit le nourrisson*" se représente par [la_=1, a_n=1, _no=3, nou=3, our=3, urr=3, rri=3, ric=1, ice=1, _ce=1, e_n=2, rit=1, it_=1, t_l=1, _le=1, le_=1, ris=1, iss=1, sso=1, son=1]. Dans ce mémoire, nous représentons les n-grammes en utilisant le caractère "_" à la place des blancs, pour faciliter la lecture.

Dans la section 2.2.4 page 24 du chapitre 2, nous avons détaillé les avantages de l'utilisation de n-grammes comme technique de représentation. Nous avons montré que cette technique, purement statistique, n'exige aucune connaissance linguistique. Cette indépendance linguistique est une propriété importante dans le cadre de notre sujet (la catégorisation de textes multilingues) car nous souhaitons proposer un cadre général pour la catégorisation de textes multilingues indépendant des langues et n'exigeant donc pas de connaissances linguistiques. Un autre avantage des n-grammes est la capture automatique des racines les plus fréquentes [Grefenstette, 1995] : dans l'exemple précédent, grâce aux techniques basées sur les n-grammes nous trouvons la racine commune de : nourrir, nourri,

nourrit, nourrissez, nourrissant, ... , nourriture, ... , nourrice, ... La tolérance aux fautes d'orthographes et aux déformations est également une propriété importante [Miller et al., 1999]. Enfin, ces techniques n'ont pas besoin d'éliminer les mots-outils (Stop Words) ni de procéder à la lemmatisation, ni au Stemming [Fürnkranz, 1998, Sahami, 1999, Zhou and Guan, 2002].

4.3. Étapes de la recherche des mots caractéristiques

Pourquoi les n-grammes constituent-ils un outil efficace pour le classement de textes ? Comment passe-t-on de la forme au sens ?

Pour répondre à ces questions, nous recherchons les n-grammes spécifiques à un sous-ensemble de textes, puis les mots qui contiennent ces n-grammes spécifiques. On peut ainsi découvrir de nouveaux mots-clés pour cette classe de textes.

L'idée consiste à extraire les n-grammes caractérisant chaque classe puis à extraire les mots contenant ces n-grammes. Nous avons développé un programme en Java qui recherche et compte les n-grammes de chaque classe de textes, sélectionne les plus caractéristiques de chaque classe, puis recherche dans ces textes les mots contenants ces n-grammes caractéristiques et finalement élimine les mots « parasites ».

4.3.1. Recherche des n-grammes caractéristiques et des mots qui les contiennent

Avant de présenter l'algorithme complet, nous expliquons le principe de la démarche.

Les étapes principales sont les suivantes :

- recherche de tous les n-grammes de tous les textes de l'ensemble d'apprentissage ;
- constitution du tableau croisé des effectifs (classe de textes $\times$ n-grammes) ;
- calcul des contributions de chaque cellule de ce tableau au χ^2 d'indépendance ;
- pour chaque classe : recherche des n-grammes caractéristiques, c'est-à-dire ceux qui sont significativement plus fréquents dans les textes de cette classe que dans les autres) ;
- recherche des mots contenants ces n-grammes.

Pour caractériser une classe de texte, nous utilisons donc la statistique du χ^2. De nombreux autres indices sont disponibles, à partir de la matrice (N_{ij}) des occurrences des n-grammes i dans les classes de textes j [Yang, 1999, Aas and Eikvil, 1999] ; la statistique du χ^2 est souvent citée parmi les plus efficaces lors des comparaisons empiriques.

En pratique, la méthode proposée fournit une longue liste de mots parmi lesquels certains sont des "parasites", c'est à dire des mots contenants par hasard un des n-grammes caractéristiques de la classe, sans que le mot lui-même soit intéressant. L'objectif suivant est d'affiner la liste des « *candidats mots-clefs* ».

4.3.2. Filtrage des mots « parasites »

Pour éviter les mots « parasites » nous proposons de reprendre le traitement à l'inverse : pour chaque mot extrait précédemment, nous examinons l'ensemble des n-grammes qu'il contient et vérifions si ces n-grammes sont suffisamment nombreux à faire partie des n-grammes caractéristiques de la classe. Nous regardons également le nombre d'occurrences de ces mots dans le corpus afin d'éviter la sélection de mots très rares. Ainsi, si :

1. la proportion des n-grammes de ce mot présents dans la liste des n-grammes caractéristiques dépasse un certain seuil (*seuil1*), et
2. la fréquence de ce mot dans le texte dépasse également un certain seuil (*seuil2*),

alors le mot sera considéré comme mot clé candidat. Si un mot se répète plusieurs fois dans le texte, c'est qu'il est intéressant. Si un mot se répète rarement, et qu'il a été sélectionné car il contient seulement un ou deux n-grammes en commun avec un autre mot qui se répète souvent, c'est que ce mot est parasite.

Exemple : un 3-gramme comme *acq* dans la classe "*Acquisition*" va donner des mots caractérisant la classe, tels "*acquisition*" ou "*acquire*" ; mais il peut également être inclus dans des mots qui n'ont rien de caractéristiques, comme "*Jacques*" *ou "racquets"* qui seront considérés comme parasites car ils sont rares dans cette classe.

4.3.3. Algorithme complet

Nous avons développé un programme en Java qui recherche les n-grammes, les compte, sélectionne les plus caractéristiques de chaque texte et recherche les mots correspondants comme indiqué dans l'algorithme 2.

4.4. Exemple d'application

Notre méthode vise à aider un utilisateur à sélectionner des documents qui l'intéressent pour une tâche donnée, à partir d'un ensemble de documents non structurés, comme le Web. Dans un premier temps, l'utilisateur doit constituer son ensemble d'apprentissage, c'est à dire fournir deux sous-ensembles de documents : un sous-ensemble de textes qui l'intéressent, et un sous-ensemble d'autres textes, de même(s) origine(s), mais qui ne l'intéressent pas pour cette tâche.

Comme ce travail est, par nature, spécifique à chaque utilisateur, nous présentons ici un exemple d'application que chacun peut plus facilement comprendre et contrôler, à savoir la sélection de dépêches d'agence sur tel ou tel sujet.

4.4.1. Données indexées de Reuters

Pour cet exemple, nous utilisons un jeu d'essai classique : les recueils de dépêches de l'agence de presse Reuters (voir la section 1.7 page 18 pour une présentation com-

Algorithme 2 Méthode proposée pour la sélection de candidats mots clés

pour chaque classe j **faire**
  rechercher tous les n-grammes dans tous les textes d'apprentissage
fin pour
constituer le tableau N_{ij} des occurrences des n-grammes i dans la classe j
calculer les fréquences f_{ij} correspondantes : $f_{ij} = \frac{N_{ij}}{N}$
calculer les contributions de (ij) à la statistique du χ^2 : $\chi^2_{ij} = \frac{\left(N_{ij} - \frac{N_{i.} \times N_{.j}}{N}\right)^2}{\frac{N_{i.} \times N_{.j}}{N}} = N \times \frac{(f_{ij} - (f_{i.} \times f_{.j}))^2}{f_{i.} \times f_{.j}}$
calculer le $\chi^2_{ij} \times signe(f_{ij} - (f_{i.} \times f_{.j}))$
trier le tableau des χ^2_{ij} dans l'ordre décroisant
pour chaque classe j **faire**
  déterminer la liste $\{gram_{ij}\}$ des K premiers n-grammes de la classe
fin pour
pour chaque $gram_{ij}$ **faire**
  chercher tous les mots (mot_{jk}) tels que $gram_{ij} \subseteq mot_{jk}$
  calculer le nombre $nb_{mots_{jk}}$ des répétitions de mot_{jk} dans la classe
fin pour
pour chaque mot_{jk} **faire**
  extraire les grammes $gram_{mot_{jk}}$ de mot_{jk}, leur total est noté $nbGram_{mot_{jk}}$
fin pour
pour chaque gramme $gram_{mot_{jk}}$ **faire**
  si ($gram_{mot_{jk}} \in \{gram_{ij}\}$) **alors**
    $presenceGram_{mot_{jk}} ++$
  fin si
fin pour
si $\frac{presenceGram_{mot_{jk}}}{nbGram_{mot_{jk}}} > seuil_1$ et $nb_{mots_{jk}} > seuil_2$ **alors**
  $mot_{jk} \in \{mots\, candidat\, de\, la\, classe_j\}$
fin si

plète de ce corpus). Le tableau 4.1 présente les différentes versions de cette collection.

Version	Prép par	# Catég	# Ens. Apprent	# Ens. Test	% Doc étiqutés
Vers 1	CGI	182	21 450	723	80%
Vers 2	Lewis	113	14 704	6 746	42%
Vers 2.2	Yang	113	7 789	3 309	100%
Vers 3	Apte	93	7 789	3 309	100%
Vers 4	PARC	93	9 610	3 662	100%

TAB. 4.1.: Différentes versions de la *collection Reuters*

Nous utilisons ici un sous-ensemble de l'ensemble d'apprentissage de la version "Apte" qui contient 7 789 dépêches [Yang, 1999] : les 6 709 dépêches, correspondant aux 10 classes les plus représentées dans la collection d'apprentissage. Le tableau 4.2 montre la répartition de ces 6 709 dépêches sur les 10 classes.

4.4.2. Quelques résultats

Le tableau 4.3 montre le résultat que notre méthode propose pour les classes *Acquisition* et *Crude*. Dans cette expérimentation, la valeur de *seuil 1* est égale à 2/3 et la valeur du *seuil 2* est égale à 30. La méthode propose une liste d'une centaine de mots clés candidats pour chaque classe mais, faute de place, nous ne présentons que les premiers 3-grammes significatifs avec les mots correspondants.

4.4.3. Discussion des résultats sur la collection Reuters

Pour chaque classe, la méthode nous propose une liste de candidats-mots-clefs, dont la plupart des mots parasites ont été éliminés, et ces mots sont spécifiques à chaque classe. On voit sur les tableaux que les mots proposés sont raisonnables. Mais, évidemment, ces résultats sont en partie liés aux événements de l'époque où les textes sont sélectionnés. La règle du "*toutes choses égales par ailleurs*" s'applique ici, comme ailleurs : les dépêches d'une autre période aboutiraient à une liste un peu différente.

La classe	**Acquisition**	**Earn**	**Money-fx**	**Wheat**	**Trade**
Nb textes	1629	2841	528	209	362
La classe	**Crude**	**Corn**	**Grain**	**Interest**	**Ship**
Nb textes	383	173	427	346	194

TAB. 4.2.: Répartition de l'ensemble des textes sur les 10 classes les plus représentées

La classe Acquisition		La classe Crude	
Les 3-grammes les plus significatifs	**Les mots clés extraits**	**Les 3-grammes les plus significatif**	**Les mots clés extraits**
acq cqu qui uis iti sit	acquisition	oil _oi il_ il,	oil oil,
acq cqu qui uir	acquire acquired acquiring acquiring	rud cru ude	crude
sha har are	share	bar arr rel els	barrels
sha har are reh hol lde eho old der	shareholder holders holding hold	cua uad dor ado	ecuador ecuadorean
com omp any pan	companies company ;	bpd _bp pd_ pd,	bpd (baril par jour)
tak ake eov keo	takeover stake take	gas	gas
sto toc ock lde	stockholders	ene erg rgy nergy_	energy
mer erg	merger ; merge	pet etr leu eum ole	petroleum
off ffe fer	offer offers offering offered	plo xpl lor ora	exploration
has pur has	purchase	sau aud udi di_	saudi
usa sai air	USAir	zue ezu nez uel	venezuela
buy	buy buys	bbl	bbl (barrel)
inv nve sto	investment investment investor	pip ipe pel	pipeline
scl	disclosed undisclosed	xxo exx	exxon
cyc ycl	cyclops	ref ner efi	refinery
sac	transaction		
com omp ple let	complete	ara rab iea	arabian arabia
fil	filing	cub bic ubi	cubic
oup	group	_ku kuw uwa wai	kuwait kuwaiti
tst	outstanding	ric ice ces	prices
twa	twa (Trans World Airline)	ope pec	opec (non-opec)

TAB. 4.3.: Premiers 3-grammes significatifs avec les mots correspondants

La méthode est complètement indépendante de la langue, dans la mesure où on peut ne retirer aucun séparateur : ni blanc, ni signe de ponctuation.

Les essais réalisés en utilisant - soit les 1+2+3-grammes, - soit les 4-grammes n'ont pas apporté d'amélioration ; les résultats sont quasi semblables. Sur ce point, nous retrouvons les conclusions de nombreux auteurs notamment [Cavnar and Trenkle, 1994, Lelu and Hallab, 2000, Fürnkranz, 1998].

Deux tableaux complètent cette présentation :

1. Le tableau 4.4 contient les listes complètes des n-grammes spécifiques des classes *corn* et *crude*, chacune contre les 9 autres classes de dépêches ;
2. Le tableau 4.5 présente les résultats comparés de notre méthode sur quatre codages possibles des textes :
   a) calculs des χ^2_{ij} sur le tableau (**mots** i $\times$ classes j) **avec un pré-traitement** : élimination des ponctuations et espaces,
   b) calculs des χ^2_{ij} sur le tableau (**mots** i $\times$ classes j) **sans pré-traitement** : on laisse les ponctuations et les espaces,
   c) calculs des χ^2_{ij} sur le tableau (**n-grammes** i $\times$ classes j) **avec un pré-traitement** : élimination des ponctuations et espaces,
   d) calculs des χ^2_{ij} sur le tableau (**n-grammes** i $\times$ classes j) **sans pré-traitement** : on laisse les ponctuations et les espaces.

On voit que notre méthode, fondée sur les n-grammes et sans aucun pré-traitement donne d'excellents résultats. Cela laisse à penser qu'elle est complètement indépendante de la langue des textes.

4.5. Conclusion

Dans ce travail nous exposons une méthode qui aide à comprendre pourquoi les n-grammes donnent de bons résultats. Nous proposons pour cela un algorithme qui extrait des candidats-mots-clés spécifiques à un sous-ensemble de textes. Une application est réalisée sur 6 709 dépêches classées en 10 classes (les classes les plus représentées dans la collection Reuters). La méthode donne des résultats encourageants ; les mots qui ont en commun des n-grammes significatifs sont sélectionnés. Nous proposons ensuite une méthode pour réduire les mots parasites, fondée sur la fréquence des mots et la proportion de n-grammes significatifs qu'ils contiennent. Cette méthode s'avère efficace, bien qu'elle travaille sur les fichiers de textes bruts, sans aucune analyse linguistique préalable.

En outre, les résultats de cette approche montrent que le passage des n-grammes sélectionnés vers les mots apporte non seulement les mots statistiquement significatifs (au sens du χ^2) mais également les flexions présentes dans le corpus (et supérieurs au seuil minimal pré-requis des fréquences) de ces mots. Nous projetons ainsi d'utiliser

cette représentation plus robuste, de part les propriétés des n-grammes, et plus riche, de part les flexions, pour l'aide à la création d'ontologie et pour la catégorisation automatique.

Les 3-grammes	Les mots extraits
[orn] [onn] [rn_] [nne] [aiz] [acr] [ton] [0_t] [gra] [usd] [soy] [oyb] [ybe] [_–] [sda] [ze_]	[corn] [corn.] [corn,] [tonnes,] [tonnes] [tonne] [tonnes.] [tonne,] [maize] [maize,] [acreage]
[nes] [arv] [–_] [ghu] [bea] [rai] [cor] [ize] [_us] [hum] [da_] [-_-] [sov] [far] [_so] [arm]	[acres.] [acres] [washington,] [grain] [program] [program.] [grains] [program.] [usda] [usda's]
[ltu] [ush] [rgh] [gri] [u.s] [.s.] [iet] [_to] [ssr] [ram] [mai] [uga] [por] [ean] [nro] [._c]	[usda.] [soybeans] [soybean] [soybeans,] [harvest] [sorghum] [rains] [record] [ussr] [soviet] [so-
[mm_] [_mm] [ogr] [vie] [rog] [icu] [huc] [s._] [gnu] [ort] [_u.] [enr] [rtm] [bue] [86/]	viets] [farmers] [farm] [sources] [agriculture] [agricultural] [bushel] [bushels] [u.s.] [u.s.-ussr] [to]
[xpo] [wee] [_ep] [cul] [6/8] [/87] [cre] [ain] [cer] [nkn] [eat] [ovi] [nup] [aby] [rod] [odu]	[total] [sugar] [reported] [export] [report] [imports] [exports] [exporters] [import] [reports] [im-
[pik] [rn,] [kab] [_gr] [rva] [whe] [ric] [ze,] [sug] [epa] [hea] [cro] [sr_] [duc] [liv] [dob]	ports.] [enrollment] [u.s. corn] [huckaby] [u.s. agriculture] [department] [1986/87] [week] [bet-
[rn.] [mpo] [rme] [eag] [llm] [fob] [unk] [she] [fre] [-55] [rts] [tot]	ween] [certificates] [producers] [unknown] [wheat] [wheat,] [products] [production] [growers]
	[conservation] [price] [prices] [crop] [french]
[oil] [_oi] [bpd] [rud] [_bp] [il_] [cru] [pd_] [bar] [rel] [cua] [arr] [etr] [uad] [gas] [...]	[oil.] [oil,] [oil] [oilfield] [bpd] [(bpd)] [bpd.] [bpd,] [crude] [crude,] [crudes] [crude.] [oil prices]
[rgy] [eum] [leu] [xpl] [els] [sau] [dor] [pet] [ado] [zue] [ezu] [0_b] [nez] [uel] [ira] [aud]	[oil industry] [oil companies] [oil prices,] [oil prices.] [oil price] [oil and] [oil production] [bpd
[ole] [di_] [bbl] [ec_] [pip] [lor] [cks] [l_p] [pd,] [tpu] [plo] [utp] [gy_] [..1] [odu] [cub]	in] [barrel.] [barrel] [barrels] [barrels.] [barrel,] [barrels,] [reliance] [ecuador,] [ecuador] [ecua-
[rod] [ude] [kuw] [uwa] [pd.] [n_b] [i_a] [_cr] [pel] [iea] [rre] [bic] [_ir] [ice] [xxo] [exx]	dor's] [ecuadorean] [petroleum] [petrobras] [petroleos] [petroleum,] [gasoline] [gas] [energy] [ex-
[raq] [bia] [ner] [udi] [/bb] [ara] [ipe] [rab] [ene] [pd)] [mex] [obr] [thq] [hqu] [s/b] [uak]	ploration] [exploratory] [exploration.] [saudi] [saudis] [venezuela] [venezuela,] [venezuelan] [ve-
[_op] [duc] [al-] [ref] [(bp] [e_o] [ubi] [fie] [ait] [pec] [tro] [fue] [uot] [_ie] [ora] [ls_]	nezuela's] [fuel] [iraqi] [iraq] [iranian] [iran] [iran's] [saudi arabia] [dlrs/bbl.] [opec] [non-opec]
[dri] [quo] [fsh] [efi] [eia] [mob] [as_] [exa] [pdv] [vsa] [dvs] [wai] [_ku] [_ga] [f_o] [ric]	[pipeline] [pipeline,] [stocks] [output] [products] [product] [production] [production.] [producing]
[um_] [_bb] [ces] [ukm] [dez] [iel] [urk] [aqi] [try] [xic] [uct] [abi] [naz] [rol] [xac] [a_b]	[producer] [produce] [producer] [producers] [produced] [products.] [products,] [production,] [cubic] [kuwait]
[erg] [eik] [_dr] [put] [prt] [qi_] [c_m] [kh_] [ian] [tex] [ikh] [aeg] [ia_] [ia'] [_pd] [aq_]	[kuwait,] [kuwaiti] [mln barrels] [mln bpd] [iea] [current] [prices] [prices.] [price] [prices,] [exxon]
[rs/] [wti] [_km] [noc] [ope] [ubr] [il,]	[arabia] [arabian] [arabia's] [arabia,] [refinery] [general] [refineries] [refiners] [including] [arab]
	[mexico] [earthquake,] [earthquake] [operating] [opec's] [operations] [open] [reduction] [refining]
	[crude oil] [the oil] [because of] [price of] [fields] [field] [expected] [quota] [quoted] [barrels
	per] [barrels of] [barrels a] [drill] [drilling] [offshore] [mobil] [was] [has been] [as] [texas] [as
	a] [texaco] [pdvsa] [of oil] [american] [sources] [industry] [ministry] [a barrel,] [a barrel] [sheikh]
	[drop] [canadian]

TAB. 4.4.: Listes complètes de n-grammes spécifiques et de "candidats-mots-clefs" pour les classes *Corn* puis *Crude*

La technique	Les mots extraits
extraction à partir de mots complets avec élimination des ponctuations et espaces	[oil] [bpd] [crude] [opec] [barrels] [barrel] [ecuador] [energy] [exploration] [petroleum] [prices] [gasoline] [gas] [refinery] [saudi] [saudis] [pipeline] [production]
extraction à partir de mots complets sans élimination de ponctuation et espaces	[oil.] [oil,] [oil] [crude] [opec] [opec's] [non-opec] [barrels] [barrels.] [barrels,] [bpd] ([bpd]) [bpd.] [bpd,] [energy] [petroleum] [ecuador,] [ecuador] [ecuador's] [exploration] [gasoline] [gas] [refinery] [saudi] [saudis] [prices] [prices.] [prices,] [barrel.] [barrel] [barrel,] [cubic] [production] [production,] [output] [stocks] [drilling] [pipeline,] [today] [day] [days] [yesterday] [iea] [arabia] [arabian] [natural] [venezuela] [venezuelan] [texaco] [petrobras] [api] [herrington] [mobil] [exxon] [offshore] [iranian] [feet] [15.8] [quota] [refining] [reserves] [kuwait] [wells] [fuel] [fields] [industry] [field] [iraqi] [minister] [spot] [demand] [price] [lukman] [santos] [producing] [iraq] [shell] [sources] [texas] [rigs] [research] [sea] [iran] [greece] [gulf]
extraction à partir de n-grams complets avec élimination de ponctuation et espaces	[oil] [bpd] [bp] [crude] [crudes] [oil prices] [oil industry] [oil stocks] [oil companies] [oil minister] [oil company] [oil price] [oil and] [oil production] [bpd in] [barrel] [barrels] [ecuador] [ecuadorean] [petroleum] [petrobras] [petroleos] [petro-canada] [gasoline] [gas] [energy] [exploration] [exploratory] [levels] [saudi] [saudis] [venezuela] [venezuelan] [fuel] [iraq] [iranian] [iran] [000 barrels] [000 bpd] [saudi arabia] [bbl] [pipeline] [stocks] [output] [products] [product] [production] [producing] [producer] [produce] [producers] [produced] [cubic] [kuwait] [kuwaiti] [iea] [mln barrels] [mln bpd] [current]
extraction à partir de n-grams complets sans élimination de ponctuation et espaces	[oil.] [oil,] [oil] [oilfield] [bpd] [(bpd)] [bpd.] [bpd,] [crude] [crude,] [crudes] [crude.] [oil prices] [oil industry] [oil companies] [oil prices,] [oil prices.] [oil price] [oil and] [oil production] [bpd in] [barrel.] [barrel] [barrels] [barrels.] [barrel,] [barrels,] [reliance] [ecuador,] [ecuador] [ecuador's] [ecuadorean] [petroleum] [petrobras] [petroleos] [petroleum,] [gasoline] [gas] [energy] [exploration] [exploratory] [exploration.] [saudi] [saudis] [venezuela] [venezuela,] [venezuelan] [venezuela's] [fuel] [iraqi] [iraq] [iranian] [iran] [iran's] [saudi arabia] [dlrs/bbl.] [opec] [non-opec] [pipeline] [pipeline,] [stocks] [output] [products] [product] [production] [production.] [producing] [producer] [produce] [producers] [produced] [products.] [products,] [production,] [cubic] [kuwait] [kuwait,] [kuwaiti] [mln barrels] [mln bpd] [iea] [current] [prices] [prices.] [price] [prices,] [exxon] [arabia] [arabian] [arabia's] [arabia,] [refinery] [general] [refineries] [refiners] [including] [arab] [mexico] [earthquake] [earthquake,] [operating] [opec's] [operations] [open] [reduction] [refining] [crude oil] [the oil] [because of] [price of] [fields] [field] [expected] [quota] [quoted] [barrels per] [barrels of] [barrels a] [drill] [drilling] [offshore] [mobil] [was] [has been] [as] [texas] [as a] [texaco] [pdvsa] [of oil] [american] [sources] [industry] [ministry] [a barrel.] [a barrel] [sheikh] [drop] [canadian]

TAB. 4.5.: Comparaisons de quatre techniques sur la classe *"Crude"*

Chapitre 5

Techniques pour la construction de classifieurs

Sommaire

5.1. Introduction 52
5.1.1. Manière de construction du classifieur 52
5.1.2. Caractéristique du modèle 53
5.2. Méthode de Rocchio 53
5.3. Arbres de décision 55
5.3.1. Phase d'apprentissage 56
5.3.2. Phase de classification 61
5.3.3. Critiques de la méthode 61
5.4. Classifieurs à base d'exemples 62
5.4.1. K-plus proches voisins 63
5.5. Fonctions à bases radiales 67
5.6. Machine à Vecteurs de Support 68
5.6.1. Cas des classes linéairement séparables 69
5.6.2. Cas des classes non séparables 70
5.7. Évaluation de classifieurs de textes 71
5.7.1. Évaluation des classifieurs, l'approche « binaire » . . . 72
5.7.2. Évaluation des classifieurs, l'approche « multi-classes » 76
5.8. Contributions personnelles 77
5.8.1. Nouvelle utilisation des SVM 77
5.8.2. Nouvelle utilisation des réseaux RBF 77
5.8.3. Nos expérimentations 77
5.9. Conclusion : quel est le meilleur classifieur ? 79

5.1. Introduction

Les algorithmes d'apprentissage supervisés tentent d'ajuster un modèle qui explique le lien entre des documents d'entrée et les classes de sortie. En catégorisation de textes, on fournit à la machine des exemples sous la forme (Document, Classe). Cette méthode de raisonnement est appelée inductive car on induit de la connaissance (le modèle) à partir des données d'entrée (l'échantillon de Documents) et des sorties (leurs Classes). Grâce à ce modèle, on peut alors estimer les classes de nouveaux documents : le modèle est utilisé pour « prédire ». Le modèle est bon s'il permet de bien prédire.

Le nombre très important de conférences et de publications relatives à la catégorisation de textes rend impossible une présentation exhaustive des algorithmes. Dans ce chapitre nous nous contentons de présenter les méthodes les plus utilisées dans la littérature en insistant sur les caractéristiques, les avantages et les limites de chaque méthode, puis nous présentons comment ces méthodes sont utilisées dans le cadre de la catégorisation de textes. Nous montrons tout d'abord comment les classifieurs sont construits.

Les techniques utilisées peuvent être différenciées selon le mode de la construction du classifieur ou selon ses caractéristiques.

5.1.1. Manière de construction du classifieur

Un classifieur peut être construit manuellement ou bien automatiquement, par induction à partir des données. Les systèmes construits manuellement, comme le système à base de connaissances *Mycin* [Shortliffe, 1976], pour le diagnostic, et le système *CONSTRUE* [Hayes and Weinstein, 1990], développé et testé par le Groupe Carnegie sur la collection de dépêches Reuters, ont fait l'objet de critiques concernant a) le coût associé à la mise au point de classifieurs complexes et b) la difficulté de les faire évoluer dans le temps. L'apprentissage automatique répond en partie à ces critiques, en *automatisant* l'acquisition des connaissances nécessaires pour construire un système à base de connaissances. On dispose d'un ensemble d'objets (textes, documents, ...) divisé en plusieurs classes ; chaque objet est décrit par un certain nombre d'attributs. On connaît pour certains objets la classe à laquelle ils appartiennent. On généralise ces exemples sous la forme d'une fonction f pour la classe c_i. Cette fonction f (appelée aussi le classifieur) sert ensuite à prédire la classe d'appartenance de tout objet non classé.

On parle d'une construction automatique « *hard* » et d'une construction semi-automatique « *ranking* » selon la valeur envoyée par la fonction f.

La construction inductive d'un classifieur « *ranking* » pour la classe $c_i \in C$ consiste en la définition d'une fonction $f : D \rightarrow [0, 1]$ qui retourne une valeur comprise entre 0 et 1 pour chaque document à classer. Cette valeur est ensuite interprétée selon la méthode d'apprentissage utilisée : pour le classifieur « Bayésien Naïf », c'est

une estimation de la probabilité que le document appartienne à la classe c_i ; pour la méthode de Rocchio [voir section 5.2], c'est la proximité du document à classer au profil « prototypique » de la classe c_i dans l'espace de représentation.

La construction inductive d'un classifieur « *hard* » correspond à la définition d'une fonction $f : D \rightarrow [V, F]$: pour chaque document à classer la fonction f retourne une valeur parmi deux valeurs possibles : V (vrai) si le document appartient à la classe c_i ou F (faux) sinon. Remarquons qu'un classifieur « *ranking* » peut être transformé en classifieur « *hard* » en fixant une seuil τ_i tel que si $f(d_j) \geq \tau_i$ alors le document est affecté à la classe c_i .

5.1.2. Caractéristique du modèle

Le modèle est-il *compréhensible,* ou bien s'agit-il d'une *fonction numérique* calculée à partir de données servant d'exemples ? La distinction principale entre *induction (apprentissage) numérique* et *apprentissage symbolique inductif* réside dans l'expression de la fonction f ; l'apprentissage symbolique produit des expressions compréhensibles, telles que des règles de production ou des arbres de décision.

De bons exemples d'apprentissage symbolique et d'apprentissage numérique pour la catégorisation de texte peuvent être trouvés dans [Moulinier, 1996] et [Stricker, 2000, Amini, 2001] respectivement.

Parmi les méthodes d'apprentissage les plus souvent utilisées figurent la régression logistique [Hull, 1994], les réseaux de neurones [Wiener et al., 1995, Wiener, 1995] et [Schütze et al., 1995], l'algorithme du perceptron [Ng et al., 1997], les plus proches voisins [Yang and Chute, 1994], les arbres de décision [Lewis and Ringuette, 1994, Apté et al., 1994], les réseaux bayesiens [Lewis, 1992a, Lewis and Ringuette, 1994, Joachims, 1997, Sahami, 1998], les machines à vecteurs supports [Joachims, 1998, Dumais et al., 1998] et, plus récemment, les méthodes basées sur la méthode dite de *boosting* [Schapire et al., 1998, Iyer et al., 2000].

5.2. Méthode de Rocchio

Certains classifieurs linéaires représentent les catégories par des profils « prototypiques ». Un profil de la classe c_i est une liste de termes pondérés, dont la présence et l'absence discriminent au mieux cette classe c_i. Les avantages de ce type de classifieurs sont la simplicité et l'interpretabilité, car, pour un expert, ce profil prototype est plus compréhensible qu'un réseau de neurones par exemple. L'apprentissage de ce type de classifieur est souvent précédé par une sélection et une réduction de termes.

La méthode de Rocchio est un classifieur linéaire proposé dans [Rocchio, 1971] pour améliorer les systèmes de recherche documentaires. Ce classifieur s'appuie sur une représentation vectorielle des documents [Salton and McGill, 1983] : chaque document d_j est représenté par un vecteur $\mathbf{d}_\mathrm{j}$ de $\mathbb{R}^n$ (n est le nombre de termes après

sélection et réduction) ; chaque coordonnée t_{kj} se déduit du nombre d'occurrences $\#(t_k, d_j)$ du terme t_k dans d_j, par :

$$\text{TF} \times \text{IDF}(t_k, d_j) = \#(t_k, d_j) * \log \frac{|Tr|}{\#Tr(t_k)} \tag{5.1}$$

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

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

Selon la méthode de Rocchio, pour chaque catégorie c_i, les coordonnées t_{ki} du profil prototypique $\mathbf{c}_i = (t_{1i}, \cdots, t_{|\tau|i})$ sont calculé ainsi :

$$t_{ki} = \beta . \sum_{\{d_j \in \text{POS}_i\}} \frac{t_{kj}}{|\text{POS}_i|} - \gamma . \sum_{\{d_j \in \text{NEG}_i\}} \frac{t_{kj}}{|\text{NEG}_i|} \tag{5.2}$$

avec $\text{POS}_i = \{d_j \in Tr \mid \Phi(d_j, c_i) = T\}$, $\text{NEG}_i = \{d_j \in Tr \mid \Phi(d_j, c_i) = F\}$. β et γ sont deux paramètres choisis selon l'importance que l'on accorde aux deux ensembles POS_i et NEG_i. [Hull, 1994, Schütze et al., 1995, Dumais et al., 1998, Joachims, 1998] fixent, par exemple, la valeur β à 1 et celle de γ à 0. En règle générale, l'on peut réduire le rôle des exemples négatifs dans la construction de classifieur en choisissant une valeur élevée pour β et une valeur faible pour γ. [Cohen and Singer, 1999] et [Joachims, 1997] utilisent $\beta = 16$ et $\gamma = 4$.

Les profils prototypes $\mathbf{c}_i$ correspondent donc aux barycentres des exemples (avec un coefficient positif pour les exemples de la classe, et négatif pour les autres). Le classement de nouveaux documents s'opère en calculant la distance euclidienne entre la représentation vectorielle du document et celle de chacune des classes ; le document est assigné à la classe la plus proche.

La méthode de Rocchio présente deux caractéristiques importantes [Vinot and Yvon, 2002] :

– elle implémente une règle de décision qui dessine des séparations linéaires (hyperplans) dans l'espace de représentation des textes. [Lewis et al., 1996] montre que les performances de Rocchio avec feedback dynamique, sur des tâches de filtrage, sont comparables à celles d'un réseau de neurones entraîné par descente de gradient (Widrow-Hoff). La méthode de Rocchio devrait donc être peu adaptée quand la séparation des classes n'est pas linéaire ;

– chaque exemple contribue identiquement à la construction du centroïde de sa classe. Rocchio s'oppose ainsi d'une part aux algorithmes dirigés par les erreurs (réseaux de neurones, SVM), qui donnent plus d'importance aux exemples mal classés, et d'autre part aux algorithmes locaux, qui n'utilisent qu'une faible partie des exemples à chaque classification (par exemple les K-PPV (K plus proches voisins)).

[Vinot and Yvon, 2002] testent la sensibilité de l'algorithme de Rocchio au *bruit*. Ils ont réalisé des expériences en bruitant[1] peu à peu les étiquettes des classes ; ils concluent que la méthode de Rocchio est exceptionnellement robuste au bruit : même avec 50% des exemples bruités, ses performances sont presque inchangées.

Diverses améliorations récentes de ce modèle se sont avérées fructueuses : de nouvelles méthodes de calcul de $\mathbf{d_j}$; un choix plus raisonné des exemples négatifs intervenant dans l'équation 5.2 (Query Zoning et feedback dynamique [Singhal et al., 1997, Buckley and Salton, 1995]) ont ainsi permis d'améliorer sensiblement les performances, le hissant, dans certaines conditions expérimentales, au niveau des meilleurs algorithmes. [Schapire et al., 1998] concluent que la méthode de Rocchio obtient une performance comparable à celles obtenues par les méthode les plus sophistiquées, comme celle de boosting, avec un temps d'apprentissage 60 fois plus rapide. Ces résultats vont sans doute renouveler l'intérêt porté à cette méthode ; mais d'autres auteurs tels [Lewis et al., 1996, Joachims, 1998, Cohen and Singer, 1999, Yang and Liu, 1999] considèrent qu'elle est surclassée.

La méthode de Rocchio dans [Vinot and Yvon, 2002] est appliquée aux corpus dont les classes à discriminer sont dotées d'une structure interne, par exemple lorsqu'il existe différents *sous-groupes thématiquement homogènes* au sein d'une même classe. Cette situation se rencontre dans les corpus réels : pour une tâche de filtrage de courrier électronique, les « catégories courrier valide » et « courrier non-sollicité » recoupent en fait des sous-groupes thématiquement très disparates. [Vinot and Yvon, 2002] montrent que la méthode de Rocchio est particulièrement performante sur les tâches de routage où l'algorithme peut proposer plusieurs classes. Sa simplicité, et la faiblesse d'expressivité qui en découle, ne semblent nuire aux performances, même en présence de sous-classes thématiquement homogènes, sauf si ces thèmes sont trop éparpillés dans les différentes classes.

5.3. Arbres de décision

La méthode des arbres de décision est une méthode d'apprentissage supervisée dont le but est de calculer automatiquement les valeurs de la variable endogène (à

[1] Dans les applications réelles, ce bruit peut résulter d'assignations d'étiquettes de classes incohérentes, erronées, ou non directement corrélées aux profils lexicaux ; ou bien encore de bruit dans les documents eux-mêmes, fautes d'orthographe ou de syntaxe par exemple lorsque les textes sont des courriers électroniques.

prédire), fixée à priori, à partir d'autres informations (variables exogènes ou prédictives). Le principe des arbres de décision repose sur un partitionnement récursif des données. Le but du partitionnement est d'obtenir des groupes homogènes du point de vue de la variable à prédire. Le résultat est un enchaînement hiérarchique de règles. Un chemin, partant de la racine jusqu'à une feuille de l'arbre, constitue une règle d'affectation du type « *Si* condition *Alors* conclusion ». L'ensemble de ces règles constitue le modèle de prédiction [Zighed and Rakotomalala, 2000].

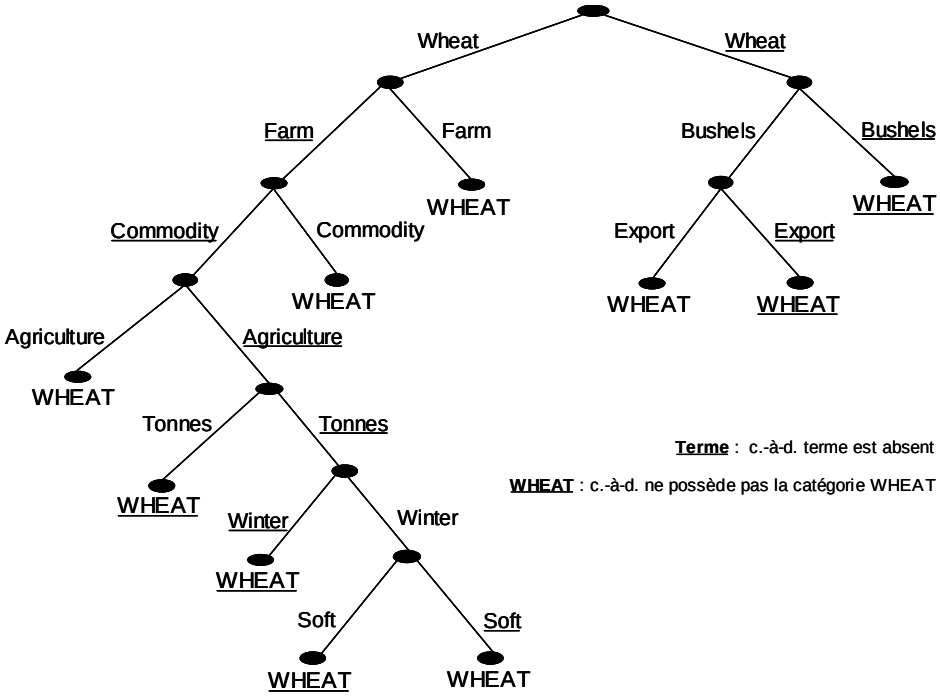

FIG. 5.1.: Exemple d'arbre de décision appliquée sur le corpus Reuters. 'WHEAT' correspond au concept *le document possède la catégorie* 'WHEAT'

La figure 5.1 présente un exemple d'un arbre de décision appliqué sur le corpus Reuters pour la classe WHEAT.

5.3.1. Phase d'apprentissage

Soit un ensemble d'individus ou d'objets concernés par le problème d'apprentissage. A cette population est associé un ensemble d'attributs particuliers appelés « attributs endogènes » noté $\mathcal{C} = \{c_1, ..., c_i, ..., c_{|\mathcal{C}|}\}$ et un ensemble d'attributs exogènes noté $\mathcal{T}$.

L'algorithme d'apprentissage prend en entrée un échantillon Ω, comprenant N enregistrements (textes) classés (d_j, c_i), et fournit en sortie un arbre de décision. L'al-

gorithme procède de façon descendante : il part de la racine puis, récursivement, choisit l'étiquette des fils. L'algorithme 3 est l'algorithme générique pour les arbres de décision.

Algorithme 3 Algorithme général d'apprentissage par arbres de décision

Contexte : un échantillon Ω de S textes classés (d_j, c_i)
Vérifier : arbre vide ; nœud courant : racine ; échantillon courant : Ω

1: **répéter**
2: **si** le nœud courant est terminal **alors**
3: étiqueter le nœud courant par une feuille portant le nom de cette classe
4: **sinon**
5: Choisir le meilleur attribut (terme) pour créer le sous-arbre
6: **fin si** {nœud courant : un nœud non encore étudié et l'échantillon courant : échantillon atteignant le nœud courant}
7: **jusqu'à** production d'un arbre de décision
8: élaguer l'arbre de décision obtenu

Sortie : arbre de décision élagué

Nous allons préciser les différents points de cet algorithme :

– L'algorithme d'arbres de décision construit une succession de partitions sur l'échantillon de données d'apprentissage. Les partitions sont de plus en plus fines. Le premier nœud contient toutes les données de l'échantillon avec leurs classes.
– On cherche, parmi les variables prédictives, celle qui donne la meilleure partition selon un critère de sélection des variables et on répète le processus de segmentation pour chaque nœud obtenu sans se préoccuper des autres nœuds. Si les variables prédictives sont discrètes, chaque variable peut engendrer une partition dont le nombre d'éléments dépend du nombre de valeurs que la variable peut prendre. Si les variables sont continues, elles doivent être discrétisées, soit a priori, soit sur chaque noeud.
– Un nœud est saturé s'il n'existe aucune variable (prédictive) qui permet de créer localement une partition qui améliore le critère utilisé.
– Le processus s'arrête quand tous les nœuds sont saturés.

Soient $S = \{s_1, s_2, ..., s_i, ..., s_k\}$ une partition de k éléments (sommets terminaux ou feuilles) engendrée par l'ensemble des attributs $\mathcal{T}$ sur l'échantillon d'apprentissage Ω, et $I(S)$ un indicateur de l'incertitude liée à cette partition, défini par la fonction

$$\begin{array}{rccc} I: & \mathrm{P}(\Omega, |\mathcal{T}|) & \longrightarrow & \mathbb{R}^+ \\ & S & \longrightarrow & I(S) \end{array}$$

où $\mathrm{P}(\Omega, |\mathcal{T}|)$ est l'ensemble des partitions qui peuvent être engendrées par les attributs. A toute partition S de Ω, on peut associer un tableau T (voir le tableau 5.1)

de k lignes ($k \geq 1$) et $|\mathcal{C}|$ colonnes ($|\mathcal{C}| \geq 2$) des effectifs N_{ij} pour chaque sommet s_i et classe c_j.

	c_1	$\cdots$	c_j	$\cdots$	$c_{\|\mathcal{C}\|}$	Total
s_1	N_{11}	$\cdots$	N_{1j}	$\cdots$	$N_{1\|\mathcal{C}\|}$	$N_{1.}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
s_i	N_{i1}	$\cdots$	N_{ij}	$\cdots$	$N_{i\|\mathcal{C}\|}$	$N_{i.}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
s_k	N_{k1}	$\cdots$	N_{kj}	$\cdots$	$N_{k\|\mathcal{C}\|}$	$N_{k.}$
Total	$N_{.1}$	$\cdots$	$N_{.j}$	$\cdots$	$N_{.\|\mathcal{C}\|}$	N

TAB. 5.1.: Tableau T des effectifs, associé à une partition

Pour que l'indicateur d'incertitude $I(S)$ soit utilisable lors de la construction d'un arbre, il doit posséder plusieurs « bonnes » propriétés. Ces propriétés concernent notamment la minimalité par répartition unimodale, la maximalité par équirépartition, la symétrie et l'indépendance ; voir [Zighed et al., 1992] pour une présentation détaillée de ces propriétés.

Le principe de construction d'un arbre peut être alors décrit par les étapes suivantes (voir l'algorithme 4 page ci-contre) :

1. Calculer l'incertitude $I(S)$ de la partition S.
2. Pour chaque attribut et sommet candidats à la segmentation, calculer $I_t(S)$ où S_t représente la partition issue de S après la segmentation d'un sommet selon l'attribut t.
3. Sélectionner l'attribut qui maximise la réduction d'incertitude $\Delta I(S) = I(S) - I_t(S)$ et effectuer la segmentation selon cet attribut.
4. $S \leftarrow S_t$.
5. Si S est une partition homogène alors affecter à chacune des feuilles une classe, sinon aller en 1.

Le calcul du gain informationnel diffère d'une méthode à l'autre, mais le principe est le même. Le passage d'une partition S à une partition S_t se fait exclusivement par segmentation au moyen d'une variable ; on peut interpréter $\Delta I(S)$ comme étant la quantité d'information apportée par la variable t utilisée. Le Gain d'Information est une expression de l'information conjointe existant entre les classes et l'attribut. Plus $I_t(S)$ est faible (c'est-à-dire moins d'information pour classer les exemples avec l'attribut t est nécessaire), plus le Gain d'Information apporté par l'attribut t est important.

Si nous adoptons comme critère la mesure de l'entropie de Shannon dont l'expression générale est donnée par la formule

Algorithme 4 Réduction de l'incertitude dans les arbres de décision

Contexte : un échantillon Ω de S textes classés (d_j, c_i)
Vérifier : arbre vide ; nœud courant : racine ; échantillon courant : Ω

répéter
  si le nœud courant est terminal **alors**
    étiqueter le nœud courant par une feuille portant le nom de cette classe
  sinon
    Calculer l'incertitude $I(S)$ de la partition S.
    pour chaque attribut et sommet candidats à la segmentation, **faire**
      calculer $I_t(S)$ où S_t représente la partition issue de S après la segmentation d'un sommet selon l'attribut t.
    fin pour
    Sélectionner l'attribut qui maximise la réduction d'incertitude $\Delta I(S) = I(S) - I_t(S)$ {la réduction d'incertitude peut être aussi appelée un gain}
    effectuer la segmentation selon cet attribut.
    $S \leftarrow S_t$
  fin si {nœud courant : un nœud non encore étudié et l'échantillon courant : échantillon atteignant le nœud courant}
jusqu'à production d'un arbre de décision
élaguer l'arbre de décision obtenu

Sortie : arbre de décision élagué

$$I(S) = -\sum_{j=1}^{|\mathcal{C}|} \frac{N_{.j}}{N} \log_2 \frac{N_{.j}}{N} \tag{5.3}$$

alors l'incertitude $I_t(S)$ du sommet S, après segmentation selon les valeurs de t, peut être calculée comme suit :

$$I_t(S) = \sum_{i=1}^{k} \frac{N_{i.}}{N} * \left(\sum_{j=1}^{|\mathcal{C}|} \frac{N_{ij}}{N_{i.}} \log_2 \frac{N_{ij}}{N_{i.}} \right) \tag{5.4}$$

et le Gain d'Information, noté $gain(t)$, apporté par la segmentation du sommet S selon les valeurs de l'attribut t, est défini de la façon suivante :

$$gain(t) = \Delta I(S) = I(S) - I(S_t) \tag{5.5}$$

Lors de l'apprentissage de modèle de prédiction il est important de prévoir des dispositifs d'arrêts. En effet, le risque de sur-apprentissage devient important au fur et à mesure que l'arbre grandit. En ce qui concerne la condition d'arrêt, il existe plu-

sieurs techniques pour considérer si une nouvelle partition améliore ou pas le critère de gain d'information :

- Fixation d'un seuil en dessous duquel le nœud n'est pas divisé ;
- Ajout d'une contrainte d'admissibilité, du type : si, dans la nouvelle partition, un nombre n de nœuds ont un effectif inférieur à une valeur v fixée par l'utilisateur, la nouvelle partition n'est pas créée. Cela évite d'avoir des nœuds de très faible taille.
- Utilisation d'un test statistique, par exemple un test d'indépendance du χ^2. Les variables candidates à la segmentation sont celles qui permettent de construire un tableau de contingence dont le χ^2 est supérieur à une valeur fixée par l'utilisateur.

[Zighed, 1985] propose une autre méthode pour limiter le sur-apprentissage. Dans les expressions 5.4 et 5.5, on remplace les rapports $N_{.j}/N$ et $N_{ij}/N_{i.}$ par, respectivement, $(N_{.j}+\lambda)/(N+m\lambda)$ et $(N_{ij}+\lambda)/(N_{i.}+m\lambda)$ où m est le nombre de classes et λ est un paramètre positif qui pénalise les noeuds de faibles effectifs.

Concernant la discrétisation des variables, le choix de la méthode de discrétisation a des conséquences importantes sur la qualité du modèle de prédiction. La discrétisation consiste à découper le domaine de la variable en un nombre fini d'intervalles, chacun identifié par un code différent. Il existe plusieurs techniques de discrétisation :

- La discrétisation non supervisée, qui ne se préoccupe pas de savoir si les intervalles résultants sont avantageux ou non par rapport au problème de détermination des classes. Par exemple deux méthodes très simples :
  - Fixer un nombre K d'intervalles et construire K intervalles d'amplitudes égales ;
  - Fixer un nombre K d'intervalles et construire K intervalles d'effectifs égaux.
- La discrétisation supervisée, où on prend en compte les classes à prédire. Ce sont des méthodes gloutonnes, très simples, consistant à ranger les individus par valeurs croissantes, formant ainsi une liste ordonnée de points identifiés par leur classe. Au début, chaque séquence de points appartenant à une même classe se situe dans un sous-intervalle I_j, délimité par deux points frontière d_{j-1} et d_j qui sont des points de discrétisation potentiels. Si la même valeur appartient à plusieurs classes, le sous-intervalle associé contiendra un mélange de classes. Ensuite il existe deux méthodes :
  - Algorithme descendant : la stratégie consiste à découper l'intervalle en deux par l'optimisation d'un critère local, puis chacune des parties en deux et ainsi de suite. Parmi tous les points frontières possibles on gardera celui qui optimise le critère. On divise l'intervalle initial dans deux partitions déterminées par ce point frontière optimal et on recommence le processus pour chaque partition.
  - Algorithme ascendant : on part de la discrétisation la plus fine, engendrée par tous les points frontières identifiés au début. L'idée est de fusionner, en optimisant un critère, des paires d'intervalles adjacents (donc d'éliminer des

points de discrétisation) pour obtenir une meilleure partition.
Le critère à optimiser, dans notre cas est la maximisation du gain informationnel apporté par la bipartition.

5.3.2. Phase de classification

Le résultat de l'apprentissage d'un arbre est un enchaînement hiérarchique de règles. Comme on l'a dit, une règle est un chemin partant de la racine et descendant jusqu'à une feuille de l'arbre. Ces règles sont de type `si condition alors conclusion`. L'ensemble de ces règles constitue le modèle de prédiction.

Pour classer un nouveau individu (par exemple un texte), il suffit de descendre dans l'arbre selon les réponses aux différents tests pour l'individu considéré. A titre d'exemple, la table 5.2 montre la transformation immédiate du modèle de prédiction de la figure 5.1 en système de règles.

Si	`(wheat && frame) == oui`	alors	WHEAT
Si	`(wheat && commodity) == oui`	alors	WHEAT
Si	`(bushels && export) == oui`	alors	WHEAT
Si	`(wheat && winter &&!soft) == oui`	alors	WHEAT

TAB. 5.2.: Modèle de prédiction produit par un arbre de décision sous la forme de règles

5.3.3. Critiques de la méthode

Les arbres de décision sont des instruments privilégiés d'exploration de données, que ce soit en termes de description ou de classement. [Gilleron and Tommasi, 2000, Zighed and Rakotomalala, 2000] comparent les arbres de décision avec d'autres techniques d'apprentissage et en rapportent les caractéristiques suivantes :

lisibilité du résultat : un arbre de décision est facile à interpréter car il est la représentation graphique d'un ensemble de règles.

tout type de données : l'algorithme peut prendre en compte tous les types d'attributs et les valeurs manquantes. Il est robuste au bruit.

sélection des variables : l'algorithmes intègre une procedure de sélection de variables, ainsi les variables contenues dans l'arbre sont utiles pour la classification.

classification efficace : l'attribution d'une classe à un exemple à l'aide d'un arbre de décision est un processus très efficace et rapide (parcours d'un chemin dans un arbre).

outil disponible : les algorithmes de génération d'arbres de décision sont disponibles dans tous les environnements de fouille de données.

extensions et modifications : la méthode peut être adaptée pour résoudre des tâches d'estimation et de prédiction. Des améliorations des performances des algorithmes de base sont possibles grâce aux techniques de bagging et de boosting : on génère un ensemble d'arbres qui votent pour attribuer la classe.

sensible au nombre de classes : les performances tendent à se dégrader lorsque le nombre de classes devient trop important.

évolutivité dans le temps : l'algorithme n'est pas incrémental, c'est-à-dire, que si les données évoluent avec le temps, il est nécessaire de relancer une phase d'apprentissage sur l'échantillon complet (anciens exemples et nouveaux exemples).

Les arbres de décision sont utilisé fréquemment dans la catégorisation de textes [Mitchell, 1997]. [Fuhr et al., 1991] utilise la méthode ID3 (pour « *Induction Decision Tree* ») [Quinlan, 1986], la méthode C4.5 de [Quinlan, 1993] est testée par [Lewis and Catlett, 1994, Cohen and Hirsh, 1998, Joachims, 1998, Cohen and Singer, 1999] et la méthode C5 (voir `http://www.rulequest.com/see5-info.html`) est utilisé par [Li and Jain, 1998]. [Lewis and Ringuette, 1994, Lewis and Catlett, 1994] utilisent les arbres de décision en tant que classifieur principal tandis que [Li and Jain, 1998, Weiss et al., 1999, Schapire and Singer, 2000] les utilisent comme membre d'un « comité » dans le cadre d'un apprentissage par combinaison de décisions.

5.4. Classifieurs à base d'exemples

À la différence de la plupart des algorithmes d'apprentissage, les approches à base d'exemples (ou raisonnement à partir de cas) ou d'« instances », (traduction de *instance-based learning)* ne construisent pas une représentation abstraite mais réalisent un classement des exemples nouveaux à partir de leur similarité avec des exemples d'apprentissage [Mitchell, 1997, Sebastiani, 2002]. La phase d'apprentissage est donc particulièrement simple puisqu'elle se réduit au seul stockage des exemples d'apprentissage. Les principaux traitements ne sont ainsi réalisés qu'en phase de généralisation, les exemples du modèle étant sollicités au moment où un nouvel exemple à besoin d'être prédit [Muhlenbach, 2002].

Ces méthodes sont, partfois, appelées « systèmes d'apprentissage paresseux » (*lazy learner*) à la différence des systèmes d'apprentissage « gloutons » (*eager learners*) tels que les arbres de décision [Aha, 1997] puisque « *they defer the decision on how to generalize beyond the training data until each new query instance is encountered* » [Mitchell, 1997, page 244].

Au cours de l'apprentissage à base d'exemples, les objets constituant le modèle sont des points projetés dans un espace multidimensionnel. Étant donné un nouvel exemple, sa relation avec les exemples stockés est examinée selon la valeur d'une fonction cible de ce nouvel exemple. Dans le pire des cas, la vérification nécessite de comparer l'exemple test à chacun des exemples d'apprentissage.

[Aha et al., 1991] montre qu'un algorithme d'apprentissage à base d'exemples est caractérisé par

1. une fonction de similarité,
2. une fonction de sélection des exemples typiques,
3. une fonction de classement qui détermine de quelle manière un nouvel exemple est lié aux exemples appris.

La première application des méthodes à base d'exemples à la catégorisation de textes est attribuée à [Creecy et al., 1992]. Depuis, plusieurs auteurs tels que [Joachims, 1998, Lam et al., 1999, Larkey, 1998, Larkey, 1999, Li and Jain, 1998, Yang and Pedersen, 1997, Yang and Liu, 1999] l'ont utilisé dans leurs expérimentations. Nous allons maintenant présenter un exemple de classifieur à base d'exemples.

5.4.1. K-plus proches voisins

L'idée de base de l'algorithme des *k-plus proches voisins* (PPV), traduction de *nearest neighbor (kNN)* en anglais, est de prédire la classe d'un texte t en fonction des k textes les plus proches voisins déjà étiquetés en mémoire [Fix and Hodges, 1951, Fix and Hodges, 1952] [Cover and Hart, 1967].

La méthode ne nécessite pas de phase d'apprentissage ; c'est l'échantillon d'apprentissage, associé à une fonction de distance et à une fonction de choix de la classe en fonction des classes des voisins les plus proches, qui constitue le modèle. L'algorithme 5 montre comment classer un nouvel exemple par la méthode PPV [Gilleron and Tommasi, 2000].

Algorithme 5 Algorithme de classification par k-PPV

Paramètre : le nombre k de voisins

Contexte : un échantillon de l textes classés en $C = c_1, c_2, ..., c_n$ classes

1: **pour** chaque texte t **faire**
2: transformer le texte t en vecteur $\mathbf{t} = (x_1, x_2, ..., x_m)$
3: déterminer les k plus proches textes du texte t selon une métrique de distance
4: combiner les classes de ces k exemples en une classe c
5: **fin pour**

Sortie : le texte t associé à la classe c.

Les choix de la distance et du paramètre k sont primordiaux pour le bon fonctionnement de cette méthode. Nous présentons maintenant les différents choix possibles pour la *distance* et pour le *mode de sélection* de la classe du cas présenté.

Définition de la distance

Une distance est une application de $E \times E$ dans $\mathbb{R}^+$ telle que les propriétés suivantes soient vérifiées [Saporta, 1990, page 241] :

$$\begin{aligned} d(\mathbf{a},\mathbf{b}) &\geq 0 \\ d(\mathbf{a},\mathbf{b}) &= 0 \Leftrightarrow \mathbf{a}=\mathbf{b} \\ d(\mathbf{a},\mathbf{b}) &= d(\mathbf{b},\mathbf{a}) \\ d(\mathbf{a},\mathbf{b}) &\leq d(\mathbf{a},\mathbf{c}) + d(\mathbf{c},\mathbf{b}) \end{aligned}$$

pour tous points $\mathbf{a}, \mathbf{b}, \mathbf{c}$ de l'espace.

Dans notre cadre, les points sont des textes. On peut noter qu'un point $\mathbf{a}$ peut avoir un plus proche voisin $\mathbf{b}$ qui, lui-même, possède de nombreux voisins plus proches que $\mathbf{a}$ (voir figure 5.2).

FIG. 5.2.: A a un plus proche voisin B, B a de nombreux voisins proches autres que A

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

$$d_p(\mathbf{a},\mathbf{b}) = \sqrt{\left(\sum_i |a_i - b_i|^p\right)^{\frac{1}{p}}} \tag{5.6}$$

Selon la valeur de p, on retrouve plusieurs distances connues :

1. Si $p = 1$ cette distance est la distance de Manhattan définie par

$$d_m(\mathbf{a},\mathbf{b}) = \{|a_1 - b_1| + |a_2 - b_2| + ... + |a_n - b_n|\} \tag{5.7}$$

2. Si $p = 2$ c'est la distance euclidienne définie par

$$d_e(\mathbf{a},\mathbf{b}) = \sqrt{\sum_i (a_i - b_i)^2} \tag{5.8}$$

3. Si $p = \infty$ c'est la distance de Chebyshev [Thiel, 2001] définie par

$$d_c(\mathbf{a},\mathbf{b}) = \max\{|a_1 - b_1|, |a_2 - b_2|, ..., |a_n - b_n|\} \tag{5.9}$$

ainsi, on attribue au texte t la classe majoritaire parmi ses K plus proches voisins dans l'ensemble d'apprentissage.

Notons que la décision finale est influencée par le choix de la distance. Prenons l'exemple suivante [Gilleron and Tommasi, 2000] :

Considérons trois points $\mathbf{a} = (30, 1, 1000)$, $\mathbf{b} = (40, 0, 2200)$ et $\mathbf{c} = (45, 1, 4000)$ où la première variable est l'âge, le second indique si le client est propriétaire, ou non, de sa résidence principale, le troisième est le montant des mensualités des crédits en cours. Les variables étant hétérogènes, on les a normalisés. En utilisant les deux formules de distance 5.7 et 5.8, on obtient respectivement :

- $d_e(\mathbf{a},\mathbf{b}) = \sqrt{\frac{10}{15}^2 + 1^2 + \frac{1200}{3000}^2} \approx 1.27$,
- $d_e(\mathbf{a},\mathbf{c}) = \sqrt{\frac{15}{15}^2 + 0^2 + \frac{3000}{3000}^2} \approx 1.41$,
- $d_e(\mathbf{b},\mathbf{c}) = \sqrt{\frac{5}{15}^2 + 1^2 + \frac{1800}{3000}^2} \approx 1.21$,

et

- $d_m(\mathbf{a},\mathbf{b}) = \frac{10}{15} + 1 + \frac{1200}{3000} \approx 2.07$,
- $d_m(\mathbf{b},\mathbf{c}) = \frac{15}{15} + 0 + \frac{3000}{3000} = 2$,
- $d_m(\mathbf{a},\mathbf{c}) = \frac{5}{15} + 1 + \frac{1800}{3000} \approx 1.83$.

Selon la distance choisie, le plus proche voisin de $\mathbf{a}$ est soit $\mathbf{b}$, soit $\mathbf{c}$. La distance euclidienne favorise les voisins dont tous les variables sont assez proches, la distance de Manhattan permet de tolérer une distance importante sur l'une des variables.

Sélection de la classe

L'emploi de *k* voisins, au lieu d'un seul, assure une plus grande robustesse à la prédiction. Classiquement, dans le cas où la variable à prédire comporte deux étiquettes, ce paramètre k est une valeur impaire afin d'avoir une majorité plus facilement décidable (en évitant les *ex æquo*). Toutefois, la valeur de *k* peut changer les performances du modèle, comme cela est présenté en figure 5.3.

Nous supposons avoir déterminé les k plus proches voisins $(\mathbf{x}_1, c(\mathbf{x}_1))$, ..., $(\mathbf{x}_k, c(\mathbf{x}_k))$ d'un enregistrement $\mathbf{y}$ auquel on souhaite attribuer une classe $c(\mathbf{y})$.

Une première façon de combiner les k classes des k plus proches voisins est le vote majoritaire. Elle consiste simplement à prendre la classe majoritaire.

Une seconde façon est le vote majoritaire pondéré. Chaque vote, c'est-à-dire la classe d'un des k plus proches voisins, est pondéré : soit $\mathbf{x}_i$ le voisin considéré, le poids de $c(\mathbf{x}_1)$ est inversement proportionnel à la distance entre l'enregistrement $\mathbf{y}$ à classer et $\mathbf{x}_i$.

Dans les deux cas précédents, on peut mesurer la confiance dans la classe attribuée par le rapport entre le nombre de voix de la classe majoritaire et *k*, le nombre total de votants.

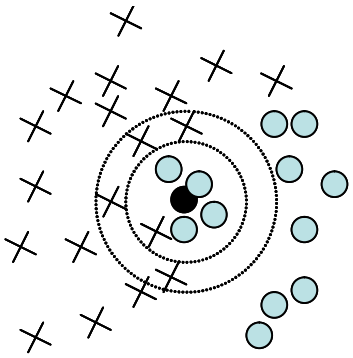

FIG. 5.3.: Choix de *k* influence la décision : pour $k = 5$, la décision est de classer l'exemple « noir » dans la classe « ronds ». Pour $k = 9$, la décision est de le classer en tant que « croix »

Mise en place de la méthode

La méthode ne nécessite pas de phase d'apprentissage. Le modèle est constitué des trois éléments : 1) l'échantillon d'apprentissage, 2) la distance et 3) la méthode de combinaison des voisins. L'efficacité de la méthode dépend de ces trois éléments.

Il faut choisir l'échantillon, c'est-à-dire les attributs pertinents pour la tâche de classification considérée et l'ensemble des enregistrements. Il faut veiller à disposer d'un nombre assez grand d'enregistrements par rapport au nombre d'attributs et à ce que chacune des classes soit bien représentée dans l'échantillon choisi [Hastie et al., 2001, en particulier la section traitant "the curse of dimensionality" pages 22–24].

La distance par variable et le mode de combinaison de ces distances sont choisis en fonction du type des variables et des connaissances préalables concernant le domaine. Il est possible d'optimiser la distance en faisant varier les paramètres et en estimant l'erreur en généralisation pour chacun des choix. L'estimation de l'erreur en généralisation se fait classiquement, soit avec un ensemble test, soit en validation croisée.

Le choix du nombre k de voisins peut, lui aussi, être déterminé par utilisation d'un ensemble test ou par validation croisée. Une heuristique fréquemment utilisée est de prendre k égal au nombre d'attributs plus 1. La méthode de vote ou d'estimation peut aussi être choisie par test ou validation croisée.

Un ensemble de limitations des plus proches voisins découle des propriétés mentionnées dans la page 3 ; voici quelques limites qui affectent la catégorisation de textes [Breiman et al., 1984] :

1. les algorithmes de type plus proche voisin sont longs en phase de généralisation, puisqu'ils sauvegardent tous les exemples de la phase d'entraînement ;

2. ils sont sensibles au bruit sur les variables prédictives ;
3. ils sont sensibles aux variables prédictives non pertinentes ; le choix des variables prédictives est donc important.
4. ils sont sensibles au choix de la fonction de similarité de l'algorithme.

5.5. Fonctions à bases radiales

Les Fonctions à bases radiales (*Radial Basis Function* ou RBF), modèle proposé par [Powell, 1987], [Broomhead and Loewe, 1988], [Moody and Darken, 1989] et [Poggio and Girosi, 1989], sont apparues à la fin des années 80 comme des variantes des réseaux de neurones. Cependant, leurs racines se retrouvent dans les techniques de reconnaissance de forme les plus anciennes comme les fonctions de potentiel (traduction de *potential functions*), le *clustering* et l'approximation fonctionnelle.

Un RBF est constitué uniquement de 3 couches : la couche d'entrée qui retransmet les entrées sans distorsion, la couche cachée RBF qui contient les neurones RBF et la couche de sortie, une simple couche contenant une fonction linéaire. Chaque couche est « *fully connected* » à la suivante.

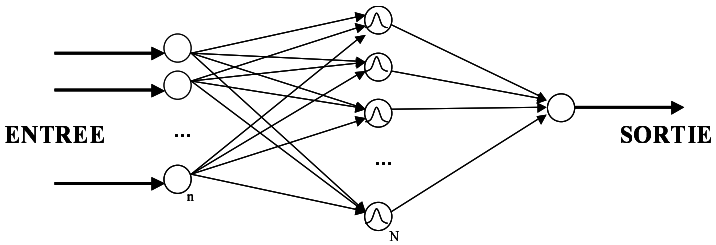

FIG. 5.4.: Chaque neurone RBF contient une gaussienne qui est centrée sur un point de l'espace d'entrée. Pour une entrée donnée, la sortie du neurone RBF est la hauteur de la gaussienne en ce point. La fonction gaussienne permet aux neurones de ne répondre qu'à une petite région de l'espace d'entrée, région sur laquelle la gaussienne est centrée. La sortie du réseau est une combinaison linéaire des sorties des neurones RBF multipliés par le poids de leur connexion respective

La fonction gaussienne d'activation est la suivante :

$$\Phi_j(\mathbf{X}) = \exp\left[-(\mathbf{X}-\mu_j)^T\Sigma_j^{-1}(\mathbf{X}-\mu_j)\right] \tag{5.10}$$

avec $j = 1, \ldots, L$ le nombre d'unité cachés, μ_j le vecteur des moyennes et Σ_j^{-1} la matrice de covariance de la $j^{\text{ème}}$ fonction gaussienne. Géometriquement, une fonction RBF représente une bosse dans un espace multidimensionnel dont la dimension est

donnée par le nombre d'entrée (les variables). μ_j représente le largeur, Σ_j représente la forme de la fonction d'activation.

Statistiquement, une fonction d'activation modélise une fonction de densité de probabilité où les μ_j et Σ_j représentent les premier et deuxième moments.

La couche de sortie implémente une somme pondéré des sorties des unités cachées. Ainsi,

$$\psi_k(\mathbf{X}) = \sum_{j=1}^{L} \lambda_{jk} \varphi(\mathbf{X}) \tag{5.11}$$

pour $k = 1, \ldots, M$ avec M, le nombre d'unité de sortie, et λ_{jk}, les poids de sortie.

Les RBF forment une classe particulière de réseaux multi-couches. Chaque cellule de la couche cachée utilise une fonction noyau (*kernel function*), telle que la Gaussienne, en tant que fonction d'activation. Cette fonction est centrée au point spécifié par le vecteur de poids associé à la cellule.

Il y a 4 paramètres principaux à régler dans un réseau RBF :

1. Le nombre de neurones RBF (nombre de neurones dans l'unique couche cachée).
2. La position des centres des gaussiennes de chacun des neurones.
3. La largeur de ces gaussiennes.
4. Le poids des connexions entre les neurones RBF et le(s) neurone(s) de sortie.

Tout modification d'un de ces paramètres entraîne directement un changement du comportement du réseau.

5.6. Machine à Vecteurs de Support

Il s'agit d'une classe récente de méthodes d'apprentissage automatique. Les Machine à Vecteurs de Support (SVM)[2] ont été introduites par [Vapnik, 1995, Vapnik, 1998]. Cette classe de méthodes est basée sur la minimisation de risque structurel[3]. Les SVM cherchent une surface de décision « épaisse » pour séparer les points de l'ensemble d'apprentissage en deux classes. La décision prise est fondée sur les vecteurs de support (traduction de *support vectors*) sélectionnés pour définir la frontière entre les classes.

[2] Certains auteurs les nomment aussi « Séparateurs à Vastes Marges », voir : [Cornuéjols, 2002].

[3] Vapnik propose des bornes reliant le risque réel $R_{rel}(\alpha)$ au risque empirique $R_{emp}(\alpha)$ mesuré sur l'ensemble d'apprentissage. Avec la probabilité $1 - \eta$, l'inégalité suivante est vrai [Vapnik, 1995] : $R_{rel}(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{1}{l}(h(\log(2l/h) + 1) - \log(\eta/4))}$ avec h la dimension VC ; l la taille de l'échantillon d'apprentissage. L'approche « minimisation structurelle du risque » (SRM) consiste à minimiser cette borne (en choisissant la bonne classe de fonction α) au lieu de minimiser simplement le risque empirique. Les SVM sont une implémentation de la méthode SRM [Viennet and Fernandez, 1999].

5.6.1. Cas des classes linéairement séparables

Soit S un ensemble de l points séparables linéairement, $S = \{\mathbf{x}_i \in \mathbb{R}^n \mid i = 1, \cdots, l\}$. Chaque point $\mathbf{x}_i$ appartient à une classe $y_i \in \{-1, +1\}$. Un hyperplan sépare l'ensemble S selon les deux classes. Cet hyperplan séparateur est défini en fonction du vecteur de poids $\mathbf{w}$ (vecteur normal à l'hyperplan) et b (où $b/\|\mathbf{w}\|$ est la distance de l'hyperplan à l'origine) et $\|\mathbf{w}\|$ est la norme euclidienne de $\mathbf{w}$; l'hyperplan vérifie l'équation :

$$\mathbf{w}.\mathbf{x} + b = 0 \tag{5.12}$$

$$\text{tel que} \begin{cases} \mathbf{w}.\mathbf{x}_i + b \geq +1 \text{ si } y_i = +1 \\ \mathbf{w}.\mathbf{x}_i + b \leq -1 \text{ si } y_i = -1 \end{cases} \tag{5.13}$$

pour $i = 1, 2, \cdots, l$; où le produit scalaire $\mathbf{w}.\mathbf{x}$ est calculé comme

$$\mathbf{w}.\mathbf{x} = \sum_j w_j x_j \text{ pour } j = 1, \cdots, n \tag{5.14}$$

En supposant qu'il existe un tel hyperplan, « *nous n'allons plus nous contenter d'en trouver un, mais nous allons en plus chercher parmi ceux-ci celui qui passe « au milieu » des points des deux classes d'exemples. Pourquoi ? Intuitivement, cela revient à chercher l'hyperplan le « plus sûr ». En effet, supposons qu'un exemple n'ait pas été décrit parfaitement, une petite variation ne modifiera pas sa classification si sa distance à l'hyperplan est grande* » [Cornuéjols, 2002]. L'objectif des SVM est alors de chercher un hyperplan *optimal* dont la distance aux exemples d'apprentissage soit maximale. Cette distance est appelé la « *marge* ».

Pour cet hyperplan, la marge vaut $1/\|\mathbf{w}\|$, et donc la recherche de l'hyperplan optimal revient à minimiser $\|\mathbf{w}\|$ [Elisseeff, 2000]. Ceci peut être formalisé ainsi :

$$\begin{array}{rl} \text{minimiser} & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{sous les contraintes} & \begin{cases} \mathbf{w}.\mathbf{x}_i + b \geq +1 \text{ si } y_i = +1 \\ \mathbf{w}.\mathbf{x}_i + b \leq -1 \text{ si } y_i = -1 \end{cases} \text{ pour } i = 1, 2, \cdots, l \end{array} \tag{5.15}$$

Les inégalités (5.15) peuvent être résumé ainsi : $y_i.f(\mathbf{x}_i, \mathbf{w}, b) \geq 1, i = 1, \cdots, l$ [Viennet and Fernandez, 1999].

En écrivant le lagrangien, on montre que la solution f s'écrit sous la forme

$$f(\mathbf{x}) = \mathbf{w}_0.\mathbf{x} + b = \sum_i \alpha_i \mathbf{x}_i \mathbf{x} + b \tag{5.16}$$

où α_i est le multiplicateur de Lagrange associé à l'exemple i. Les $\mathbf{x}_i$ qui interviennent dans la solution sont nommés *vecteurs de support*. On notera l'ensemble de

ces point $SV = \{\mathbf{x}_i\}$ pour $i = 1, \cdots, m$ avec $m \leq l$. Ce sont les points de S les plus proches de l'hyperplan qui suffisent à déterminer cet hyperplan optimal (voir la figure 5.5).

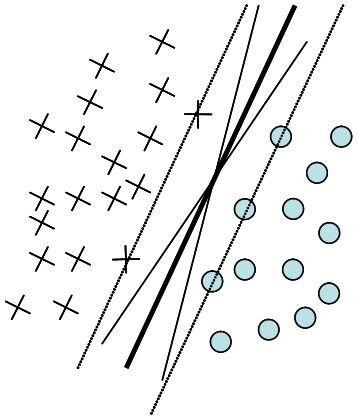

Figure 5.5.: Hyperplans, solutions d'un problème de classification linéairement séparable. L'hyperplan optimal séparant les points de deux classes est celui qui passe au milieu de l'espace entre ces classes. Les exemples les plus proches et qui suffisent à déterminer l'hyperplan optimal sont appelés vecteurs de support. La distance entre les vecteurs de support et ce plan est appelée marge

5.6.2. Cas des classes non séparables

En relâchant les contraintes, cette approche s'étend au cas où les données ne sont pas séparables grâce à des variables d'écart qui permettent à certains points de se situer du mauvais coté de la frontière (5.15).

De plus, les SVM s'étendent très élégamment pour construire des modèles non linéaires en enrichissant l'espace de représentation. Pour cela, on remarque que, dans la formule (5.16), seul intervient le produit scalaire entre deux points. On peut utiliser comme produit scalaire toute *fonction noyau symétrique* $K(\mathbf{x}, \mathbf{y})$ respectant certaines conditions (appelées conditions de Mercer [Burges, 1998]) et obtenir un comportement non linéaire :

$$f(\mathbf{x}) = \sum_{i \in SV} K(\mathbf{x}_i, \mathbf{x}) + bY \tag{5.17}$$

Les premières fonctions noyaux étudiées ont été :

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}.\mathbf{y} + 1)^p \tag{5.18}$$

$$K(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2} \tag{5.19}$$

$$K(\mathbf{x}, \mathbf{y}) = \tanh((a(\mathbf{x}.\mathbf{y}) - b)) \tag{5.20}$$

L'équation (5.18) correspond à un classifieur polynomial de degré p, l'équation (5.19) à un classifieur basé sur des gaussiennes à base radiale et l'équation (5.20) à une forme particulière de réseau de neurones avec fonctions d'activation sigmoïdales [Amini, 2001]. Notons que, pour certaines valeurs de a et b dans l'équation (5.20), le théorème de Mercer n'est pas vérifié.

Pendant la classification, les SVM prennent la décision en se fondant sur l'hyperplan optimal au lieu de voir la totalité de l'ensemble d'apprentissage. Ils cherchent simplement à déterminer sur quel côté de cet hyperplan se trouve l'exemple à classer. Cette propriété fait des SVM une méthode compétitive, en temps de calcul et en qualité de prédiction[4].

Actuellement, un nombre important d'auteurs étudient les SVM et les appliquent au texte ; citons à titre d'exemple [Joachims, 1998, Joachims, 1999, Joachims, 2000, Dumais et al., 1998, He et al., 2000].

5.7. Évaluation de classifieurs de textes

Il existe de multiples algorithmes permettant d'inférer à partir de données étiquetées, c'est-à-dire de construire des outils de catégorisation ; et le choix opérationnel du classifieur à utiliser pour traiter un type de corpus donné, spécifique à une application, reste une question difficile.

L'évaluation de classifieurs se fait habituellement d'une manière empirique (en se basant sur un échantillon de textes), et non analytique, puisque la catégorisation « vraie » est par définition subjective : c'est l'association d'un texte libre à une catégorie ou classe, *en fonction des informations que contient* ce texte. Et l'appartenance d'un texte à un catégorie ou à une autre est subjective car elle dépend du centre d'intérêt de chaque personne (voir la section 1.5.6 page 16).

Pour mesurer les performances des classifieurs, plusieurs mesures sont proposées, chacune s'intéressant à un aspect de la catégorisation. Dans la présente section nous allons présenter les mesures de performance souvent utilisées dans la littérature ; nous allons montrer leurs particularités et leur utilisation pour des problèmes de catégorisation « binaires » et « multiclasses ».

[4] « *Si la mise en oeuvre d'un algorithme de SVM est en générale peu coûteuse en temps, il faut cependant compter que la recherche des meilleurs paramètres peut requérir des phases de test assez longues* » [Cornuéjols, 2002].

5.7.1. Évaluation des classifieurs, l'approche « binaire »

Lors de la catégorisation multiclasses de textes, c.-à-d. lorsque $|\mathcal{C}| > 2$, une approche commune consiste à « couper » le processus de catégorisation en sous-problèmes. Chaque sous-problème concerne uniquement une classe et l'objectif est alors de juger si le nouveau texte appartient ou n'appartient pas à cette classe par opposition aux autres. Lors de l'évaluation de tels classifieurs à partir d'un ensemble de test, quatre nombres sont importants pour chaque classe (voir la table 5.3) :

1. le nombre de textes correctement classés comme appartenant à la classe i, noté VP_i (pour Vrai Positif).
2. le nombre de textes incorrectement classés comme appartenant à la classe, noté FP_i (pour Faux Positif).
3. le nombre de textes incorrectement rejetés, noté FN_i (pour Faux Négatif).
4. le nombre de textes correctement rejetés, noté VN_i (pour Vrai Négatif).

		Expert	
		c_i	$\neg c_i$
Classifieur	c_i	VP_i	FP_i
	$\neg c_i$	FN_i	TN_i

TAB. 5.3.: Table de contingence qui résume les $|\mathcal{C}|$ décisions prises par un système de catégorisation : c_i correspond à la décision « le document possède la catégorie c_i » tandis que $\neg c_i$ traduit « le document ne possède pas la catégorie c_i »

Précision et Rappel

Pour chaque classe c_i, deux mesures, la précision notée π_i et le rappel notée ρ_i, sont à la base des évaluations en recherche documentaire ; elles ont été adaptées pour la catégorisation de textes [Lewis, 1992a].

[Kohavi, 1995] définit la précision en apprentissage comme la probabilité conditionnelle qu'un exemple choisi aléatoirement soit bien classé par le système, autrement dit, $p(\breve{\Phi}(d_x, c_i) = Vrai / \Phi(d_x, c_i) = Vrai)$.

Le rappel mesure la « largeur de l'apprentissage » et correspond à la fraction des documents jugés pertinents par le classifieur. [Sebastiani, 2002] le présente comme $p(\Phi(d_x, c_i) = Vrai / \breve{\Phi}(d_x, c_i) = Vrai)$.

Les deux probabilités, π_i et ρ_i sont des probabilités subjectives car elles mesurent « *the expectation of the user that the system will behave correctly when classifying an unseen document under* c_i » [Sebastiani, 2002, page 33].

Ces probabilités peuvent être estimées à partir de la table de contingence 5.3, ainsi :

$$\widehat{\pi}_i = \frac{VP_i}{VP_i+FP_i} \; et \quad \widehat{\rho}_i = \frac{VP_i}{VP_i+FN_i} \tag{5.21}$$

La précision et le rappel globaux, c-à-d, sur toutes les classes, notés respectivement π et ρ, peuvent être calculés à travers une moyenne des résultats obtenus pour chaque catégorie. Deux approches sont distinguées :

1. donner un poids égal à toutes les classes, on parle alors de « macro-moyenne »
2. donner un poids proportionnel à la fréquence de chaque classe, il s'agit de la « micro-moyenne ».

		Décision du l'expert	
Décision du Classifieur	$C = \{c_1, c_2, ..., c_{\lvert\mathcal{C}\rvert}\}$	c_i	$\neg c_i$
	c_i	$VP = \sum_{i=1}^{\lvert\mathcal{C}\rvert} VP_i$	$FP = \sum_{i=1}^{\lvert\mathcal{C}\rvert} FP_i$
	$\neg c_i$	$FN = \sum_{i=1}^{\lvert\mathcal{C}\rvert} FN_i$	$TN = \sum_{i=1}^{\lvert\mathcal{C}\rvert} TP_i$

TAB. 5.4.: Table de contingence globale

La micro-moyenne (traduction de *micro-averaging*) calcule les mesures rappel et précision de façon globale : si l'on considère les tables de contingences associées à chaque catégorie, cela revient à sommer les cases VP et FP de chaque catégorie pour obtenir la table de contingence globale (voir la table 5.4). Les différentes mesures comme le π et ρ sont ensuite calculées à partir des valeurs cumulées. La micro-moyenne accorde donc des poids importants aux catégories ayant beaucoup d'exemples. La performance du classifieur dépend surtout de sa capacité à traiter les catégories les plus fréquentes [Moulinier, 1996]. Ainsi, la précision micro-moyenne et le rappel micro-moyenne sont estimés comme suit :

$$\begin{aligned} \hat{\pi}^{\mu} &= \frac{VP}{VP+FP} = \frac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} VP_i}{\sum_{i=1}^{\lvert\mathcal{C}\rvert} (VP_i+FP_i)} \\ \hat{\rho}^{\mu} &= \frac{VP}{VP+FN} = \frac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} VP_i}{\sum_{i=1}^{\lvert\mathcal{C}\rvert} (VP_i+FN_i)} \end{aligned} \tag{5.22}$$

La macro-moyenne (traduction de *macro-averaging*) évalue d'abord indépendamment chaque catégorie. Ensuite, la performance globale du classifieur est calculée en faisant la moyenne des mesures individuelles. Les différentes catégories ont alors la même importance. La précision et le rappel macro-moyenne sont calculés comme suit :

$$\hat{\pi}^M = \frac{\sum_{i=1}^{|\mathcal{C}|} \hat{\pi}_i}{|\mathcal{C}|} \quad \hat{\rho}^M = \frac{\sum_{i=1}^{|\mathcal{C}|} \hat{\rho}_i}{|\mathcal{C}|} \tag{5.23}$$

Taux de succès et taux d'erreur

Le taux de succès et le taux d'erreur sont deux mesures souvent utilisés par la communauté de l'apprentissage automatique. Le taux de succès (traduction de *accuracy rate*) désigne le pourcentage d'exemples bien classés par le classifieur, tandis que le taux d'erreur (*error rate*) désigne le pourcentage d'exemples mal classés. Les deux taux sont estimés comme suit :

$$\hat{A} = \frac{VP+VN}{VP+VN+FP+FN} \qquad \hat{E} = \frac{FP+FN}{VP+VN+FP+FN} = 1 - \hat{A} \tag{5.24}$$

En catégorisation de textes, « *l'évaluation doit prendre en compte les objectifs du système. Un système de catégorisation se concentre sur l'affectation de catégorie et n'utilise l'information « ne possède pas la catégorie » que pour limiter le nombre d'erreurs. La mesure correspondant aux taux d'erreur en apprentissage ne reflète pas cet objectif, puisqu'elle prend en compte les exemples négatifs* »[Moulinier, 1996, page 66]. Pour [Yang, 1999], la grande valeur du dénominateur laisse le taux de succès trop insensible aux variations du nombre d'exemples correctement classés ($VP + VN$) ; de ce point de vue π et ρ sont plus adaptés.

Prenons l'exemple de la version 3 de la collection Reuters, qui possède 93 catégories. Un document est affecté en moyenne à 1.2 catégories, donc la probabilité moyenne qu'un document soit affecté à une catégorie est de 1.3% $(1.2/93 = 0.013)$. Un classifieur trivial qui consiste à ne pas affecter la catégorie (c'est-à-dire, $\forall i, j\, \Phi(d_x, c_i) = faux$) aura un taux de succès de 98.7%, alors que le système n'a permis de catégoriser aucun texte. Un tel classifieur trivial est généralement plus performant que d'autres classifieurs non-triviaux !

Autres mesures de qualité d'un classifieur

Les mesures π, ρ, A et E ne prennent en compte ni le temps de calcul (ce qu'on appelle l'efficacité de calcul), ni le fait que les coûts ou bénéfices d'affectation des quatre situations (vrais positifs, vrais négatif, faux positifs et faux négatifs) ne sont pas égaux.

Bien que le temps de mise en oeuvre et les limites du matériel soient des éléments cruciaux pour les applications réelles, on observe dans la littérature académique sur la catégorisation de textes que les auteurs s'intéressent rarement aux temps de calcul ou à la volatilité (les ressources en mémoire) exigée par un classifieur. Par exemple,[Dumais et al., 1998] ont conduit une étude comparative de cinq méthodes d'apprentissage en prenant en compte trois critères :

1. la capacité d'apprentissage (traduction de *effectiveness*),

2. l'efficacité d'apprentissage (traduction de *training efficiency*, c.-à-d. le temps moyen nécessaire pour apprendre un classifieur),
3. l'efficacité de classement (traduction de *classification efficiency*, c.-à-d. le temps moyen nécessaire pour classer un nouveau texte).

Pour notre part, nous pensons aussi que, à capacités d'apprentissage et à performances similaires, le temps de calcul doit être pris en compte lors du choix entre les classifieurs.

Un autre point de vue pour évaluer les performances de classifieurs est de prendre en compte la notion de coût. En effet, il est fréquent dans la pratique que les conséquences d'un faux positif et d'un faux négatif ne soient pas les mêmes. Prenons par exemple l'analyse automatique des comptes-rendus écrits par des médecin radiologues, lors d'un dépistage du cancer. Il est évident que les coûts des erreurs ne sont pas identiques ; il est beaucoup plus grave de se tromper en déclarant bénin un cas cancéreux (car la patient ne sera pas soigné à temps) que de déclarer cancéreux un cas bénin (car le patient sera l'objet d'examens complémentaires qui pourront corriger cette erreur).

La table 5.5 montre la matrice d'utilité selon les différentes décisions prises par le système. Sur cette matrice on voit que la table de contingence 5.3 page 72 n'est qu'un cas particulier de la matrice d'utilité où les fonctions d'utilité sont ${}^{u}VP = {}^{u}VN > {}^{u}FP = {}^{u}FN$.

Classifieur \ Expert	c_i	$\neg c_i$
c_i	${}^{u}TP_i$	${}^{u}FP_i$
$\neg c_i$	${}^{u}FN_i$	${}^{u}TN_i$

TAB. 5.5.: Matrice d'utilité qui prend en compte les différentes fonctions de coûts, c_i correspond à la décision « le document possède la catégorie c_i » tandis que $\neg c_i$ traduit « le document ne possède pas la catégorie c_i »

[Androutsopoulos et al., 2000] utilisent les fonctions d'utilité pour comparer les performances d'un classifieur bayesien et d'un classifieur à base de mots-clés pour la détection de *spams*. [Lewis, 1995] présente une étude détaillée de l'utilisation de ces fonctions en catégorisation de textes. D'autres, comme [Schapire et al., 1998, Amati and Crestani, 1999, Cohen and Singer, 1999], l'ont utilisé dans leurs expérimentations. Notons, pourtant, que les fonctions d'utilité sont étroitement dépendantes du sujet traité ce qui rend difficile la comparaison générale des classifieurs. La table 5.5 et les formules 5.22 et 5.23 montrent que les mesures de performances sont dépendantes de ces fonctions.

D'autres auteurs ont proposé de « combiner » les résultats issus d'autres mesures. En effet, comme le montre la figure 5.6, on observe généralement que la croissance

du rappel entraîne la diminution de la précision. A la limite, si le classifieur affecte tous les documents d_j à la classe c_i (c.-à-d. $\forall i, j\, \Phi(d_x, c_i) = vrai$) alors la rappel $\rho = 1$. Et, dans ce cas, π aura une valeur très mauvaise.

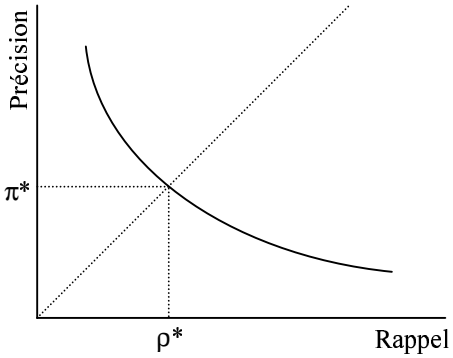

FIG. 5.6.: Courbe de « rappel et précision » et le « point moyen »

Pour pallier ce problème, on a proposé, en recherche documentaire, d'utiliser une mesure qui prend en compte les deux valeurs π et ρ. Cette mesure est appelé « le point moyen » [Lewis, 1992b, Apté et al., 1994, Moulinier and Ganascia, 1996, Joachims, 1998, Cohen and Singer, 1999]. Il se calcule directement à partir de la courbe : c'est le point où le rappel et la précision ont la même valeur. [Lewis, 1995] critique cette mesure car elle traduit plus une propriété de la courbe que du système. Pour cette raison Lewis propose d'utiliser la mesure F_β, souvent utilisée en recherche documentaire [Van Rijsbergen, 1979] :

$$F_\beta = \frac{(\beta^2 + 1)\pi\rho}{(\beta^2\pi + \rho} \tag{5.25}$$

Selon cette formule, le paramètre β peut influencer l'importance accordée aux π et ρ. Si $\beta = 0$ alors F_β coïncide avec π. Si $\beta = +\infty$ alors F_β coïncide avec ρ. Évidement, changer de mesure d'évaluation rend difficile la comparaison avec les différents travaux passés. [Moulinier and Ganascia, 1996, Yang, 1999, Moulinier, 1996, page 64] établissent la relation entre le point moyen et le F_β :

Le point moyen d'un classifieur Φ est une borne inférieure du meilleur F_β que l'on peut obtenir à partir d'une courbe rappel-précision.

5.7.2. Évaluation des classifieurs, l'approche « multi-classes »

Dans la section 5.7.1 nous avons traité le cas où le classifieur fournit une décision binaire concernant l'appartenance, ou non, à une classe c_i. Certains classifieurs, comme le classifieur bayésien, produisent une liste ordonnée de catégories

pour chaque nouveau texte à classer. Dans cette liste, les catégories sont classées par probabilités décroissantes (voir la section 5.1.1 page 53).

Pour mesurer les performances de tels classifieurs une approche dite « 11-*point average precision* » peut être utilisée. Le principe est de calculer la précision π_i et le rappel ρ_i pour chacun des 11 seuils $\tau_i = \{0.0, 0.1, 0.2, \cdots, 0.9, 1.0\}$ en utilisant la formule 5.21 page 73, puis de calculer les π et ρ moyens. Cette méthode est inspirée de la recherche documentaire ; elle est utilisée par [Yang, 1994, Yang and Pedersen, 1997, Yang, 1999, Larkey and Croft, 1996, Lam et al., 1999, Schapire and Singer, 2000].

5.8. Contributions personnelles

5.8.1. Nouvelle utilisation des SVM

L'encodage dans $\mathbb{R}^n$ permet l'utilisation de nombreux algorithmes, en particulier des SVMs. Mais on peut utiliser les SVMs d'une autre façon : on définit $K(d_1, d_2) = exp(-d(d_1, d_2))$ avec d une mesure de dissimilarité et d_1, d_2 deux textes. Nous avons choisi la distance du χ^2, sans parvenir à prouver que cette distance vérifie la condition de Mercer [Burges, 1998].

5.8.2. Nouvelle utilisation des réseaux RBF

Comme dans le cas des SVMs, on peut utiliser un réseau RBF avec un encodage des textes dans $\mathbb{R}^n$; mais nous avons estimé plus naturel d'utiliser une distance adaptée à des distributions, par exemple la distance du χ^2 (voir la formule 7.1 page 106) . Cette méthode est résumée dans l'algorithme 6.

5.8.3. Nos expérimentations

Le taux de succès est évalué par « *leave-one-out* » connu aussi sous le nom de « *Jacknife* » dans le cas de la reconnaissance d'écrivains, en raison de la petite taille du problème (28 classes, 130 textes).

Nous utilisons 130 livres en français, écrits par 28 auteurs comme Balzac, Bloy, Corneille, Diderot, Engels, Flaubert, Fourier, France, Gaberel, Gautier, Gobineau, Hugo, Huysmans, Lamartine, Leibnitz, Maistre, Maupassant, Molière, Pascal, Racine, Renard, Rostand, Rousseau, Sand, Stendhal, Verne, Voltaire, Zola. Certains auteurs sont traduits depuis d'autres langues ; ceci, et le fait que les textes sont de tailles différentes, sont considérés comme des difficultés supplémentaires pour l'algorithme.

La plupart des textes proviennent du site `http://abu.cnam.fr`. Les autres proviennent de la Bibliothèque Nationale de France : `http://www.bnf.fr`. Les résultats expérimentaux obtenus avec des 3-grammes sont donnés dasn la table 5.6.

Tous ces tests sont réalisés en Octave (voir `http://www.che.wisc.edu/octave/` pour une description de cette application gratuite clone à Matlab).

Algorithme 6 Méthode de RBF avec la distance de χ^2 comme noyau

Soient $\mathcal{D} = \mathbf{d}_1, \mathbf{d}_i, ..., \mathbf{d}_{|\mathcal{D}|}$ et $\mathcal{D}' = \mathbf{d}'_1, \mathbf{d}'_i, ..., \mathbf{d}'_{|\mathcal{D}'|}$ les ensembles de textes d'apprentissage et de teste respectivement, soit $\mathcal{C}$ l'ensemble des catégories
Soit O la matrice telle que :
si $\mathbf{d}_i$ appartient à la classe c_j **alors**
    $O_{i,j} = 1$
sinon
    $O_{i,j} = -1$
fin si
Soit $K_{i,j} = exp(-d(\mathbf{d}_i, \mathbf{d}_j))$
Soit $K'_{ij} = exp(-d(\mathbf{d}'_i, \mathbf{d}_j))$
Soit W telle que $K1 \times W = O$. //$K1$ est la matrice K, plus une colonne de 1 à sa droite
Soit $O' = K'1 \times W$.
On assigne à $\mathbf{d}'_i$ la classe $\arg\max_k O'(i,k)$.

On appelle « SVM multiclasse » une SVM spécialisée dans le multiclasses, comme défini dans [Guermeur et al., 2000]. On peut signaler que dans ce cas précis (très haute dimension, 28 classes) cette SVM est significativement meilleure que la méthode usuelle combinant des SVMs un-contre-tous comme suggéré dans [Vapnik, 1995]. On a à la fois SVM multiclasse avec χ^2 significativement meilleure que SVM avec χ^2 et SVM multiclasse linéaire significativement meilleure que la SVM linéaire.

[Yang and Liu, 1999] conclue que les SVM sont meilleures que les k-plus proches voisins qui sont eux-mêmes meilleurs que LLSF (voir [Yang and Liu, 1999]), et que les réseaux de neurones multicouches basés sur la rétropropagation, eux-mêmes meilleurs que l'algorithme Bayésien Naïf (voir [Good, 1965]). Nos essais confirment ces résultats et nous pouvons préciser, avec $\gg$ signalant une différence significative avec risque d'au plus 5%, $>$ une différence significative avec risque d'au plus 10%, $\geq$ une différence d'au plus 15% :

$$\text{RBF - SVM Multiclass } (\chi^2) \gg \text{SVM Multiclass} \geq \text{SVM } \chi^2 \text{ - SVM - 1-NN}$$

[Joachims, 1998] explique (partiellement) le bon comportement des SVMs pour la catégorisation de texte par sa capacité à traiter de nombreuses dimensions sans avoir à sélectionner des variables. On voit ici que les RBFs bénéficient de la même caractéristique.

On peut noter que nos expérimentations, comme celles de [Yang and Liu, 1999], concernent des textes suffisamment grands pour une bonne appréciation des fréquences. Sur des ensembles de données pour lesquels les fréquences sont bien définies, on peut finalement résumer nos résultats et ceux de [Yang and Liu, 1999] et [Joachims, 1998] par :

Algorithme	Taux de succès
RBF avec noyau $(\chi^2)^p$ pour $p = \frac{1}{2}, \frac{1}{4}, \ldots \frac{1}{32}$ RBF avec noyau χ^2 pour	87.69 % 86.15 %
Multiclasse SVM avec noyau χ^2	86.15 %
Multiclasse SVM linéaire	78.46 %
SVM avec noyau χ^2 , $C = 1$	72.31 %
1-PPV avec dissimilarité χ^2	70.77 %
SVM linéaire	67.69 %
1-PPV avec dissimilarité de Cavnar and Trenkle	65.38 %
1-PPV avec dissimilarité cosinus	61.54 %
1-NN avec dissimilarité Kullbach-Leibler	52.31 %

TAB. 5.6.: Taux de succès pour la reconnaissance des 28 écrivains (par la méthode de *Jacknife*)

$$\text{RBF} > \text{SVM Mc}(\chi^2) > \text{SVM Mc} > \text{SVM}(\chi^2) > \text{SVM} > 1-\text{NN} > \text{LLSF}, \text{ C4.5}, \text{ NNets} > \text{NB}$$

Avec : SVM Mc, la SVM multiclasse précédemment définie, SVM la SVM linéaire, LLSF comme décrit en [Yang and Liu, 1999], NNets des réseaux de neurones autres que SVM et NB l'algorithme bayésien naïf décrit en [Good, 1965]. Notons que RBF $>$ SVM Mc n'est pas significatif en terme de performance ; nous le conservons simplement en raison de l'avantage de vitesse et de simplicité.

Après ce premier jeu d'essai (traduction de *benchmark*), on pourrait conclure (trop vite) que les RBFs avec noyau χ^2 semblent très performants sur la catégorisation de texte. En fait la SVM multiclasse s'est avérée aussi puissante, mais les RBF sont beaucoup plus simples à implémenter et beaucoup plus rapides. Dans la section 7.4 page 107, nous allons montrer que, dans le cas de fréquences moins bien définies, les 1-plus proches voisins semblent meilleurs que les RBF, en particulier avec de très petits échantillons d'apprentissage.

5.9. Conclusion : quel est le meilleur classifieur ?

Beaucoup d'approches différentes ont été utilisées pour la catégorisation de textes. La question qui se pose est : quelle est la meilleure méthode pour la catégorisation de textes ?

Idéalement, pour comparer les performances de deux méthodes, on tente **soit** d'appliquer les deux méthodes sur les mêmes données en utilisant les mêmes mesures de performance, **soit** d'adopter une méthode d'évaluation contrôlée [Yang, 1999].

La première solution semble difficile à réaliser car :

1. les méthodes déjà présentées n'ont pas été appliquées sur les mêmes jeux de données ; par exemple, il y a au moins cinq versions différentes de la collection *Reuters*, surtout en ce qui concerne la distribution des textes entre l'ensemble d'apprentissage et l'ensemble de test, puis sur le nombre des textes à utiliser pendant les expérimentations et enfin sur les catégories à évaluer (voir tableau 1.2 page 18) ;
2. les évaluations diffèrent par les mesures utilisées pour évaluer la performance. Les auteurs n'utilisent pas les mêmes mesures de performance et peuvent calculer les moyennes de manières différentes : rappel et précision, taux de succès et taux d'erreur, le point moyen, le F_β, le micro- et macro-moyen, *11-point average precision*, *etc.* ;
3. certaines conditions, non liées aux algorithmes d'apprentissage eux-mêmes, peuvent intervenir et influencer les résultats obtenus. Ceci inclut, entre autres, les unités utilisées pour représenter les textes (mot, stemme, lemme ou n-grammes), les techniques de réduction de dimension utilisées (χ^2, *le gain d'information*, *Odds ratio*, ...), les paramètres des algorithmes, les seuils τ_i, *etc*.

Finalement, les performances obtenues ne sont pas comparables ; de plus les comparaisons présentées jugent plus la capacité des auteurs à mettre en oeuvre les méthodes, que les capacités des méthodes elles-mêmes.

Enfin, même dans le cas où les auteurs utilisent les mêmes mesures, il est nécessaire d'utiliser des tests statistiques pour vérifier que les différences ne sont pas dues au hasard [Moulinier, 1996, notamment les pages 67–68].

[Yang, 1999] propose deux méthodes pour comparer les classifieurs et les évaluer. Elle indique que la comparaison peut être directe, ou indirecte :

- une comparaison directe : il s'agit de l'utilisation de plusieurs méthodes par le même auteur ; de cette manière, le découpage et les mesures sont identiques pour toutes les méthodes. [Yang and Liu, 1999] comparent les machines à vecteurs supports, les plus proches voisins, les réseaux de neurones, une combinaison linéaire, et des réseaux bayesiens. [Dumais et al., 1998] proposent une série de comparaisons en mettant en compétition une variante de l'algorithme de Rocchio (appelée *find similar*), des arbres de décision, des réseaux bayesiens et des machines à vecteurs supports. Cette méthode de comparaison est la plus crédible de point de vue scientifique [Sebastiani et al., 2000].
- une comparaison indirecte : soient Φ' et Φ'' deux classifieurs. Ces deux classifieurs peuvent être comparés si deux conditions sont réunies :
  - les deux classifieurs sont testés par différents groupes de chercheurs (même avec de conditions d'expérimentation différentes) sur deux collections Ω' et Ω'' respectivement ;
  - un ou plusieurs classifieurs de « références » $\bar{\Phi}_1, \bar{\Phi}_2, ..., \bar{\Phi}_m$ sont testés sur les deux collections Ω' et Ω'' par des comparaisons directes ; ceci donne une idée du niveau de difficulté d'apprentissage sur chaque collection.

Alors, une indication sur les performances de classifieurs Φ' et Φ'' peut être obtenue via ces deux tests.

En prenant en compte ces considérations et les résultats obtenus par d'autres auteurs sur les différentes collections de Reuters, nous pouvons avancer (avec $\gg$ signalant « nettement plus difficile », $>$ signalant « plus difficile » et $\approx$ signalant « équivalent ») :

$$\text{Reuters22173"ModLewis"} \gg \text{Reuters22173"ModWiener"} > \text{Reuters22173"ModApte"} \approx \text{Reuters21578"ModApte"} > \text{Reuters2178[10]"ModApte"}$$

Concernant les performances des classifieurs, nous pouvons, à titre d'indication, donner quelques résultats :

- Les classifieurs par combinaison de décisions, les SVM, les méthodes à base d'exemples donnent les meilleurs résultats.
- Les réseaux de neurones, les classifieurs « en ligne » donnent des résultats inférieurs à ceux des précédents.
- Les classifieurs linéaires tels que la méthode de Rocchio et le classifieur naïf de Bayes, donnent souvent de mauvais résultats.
- WORD, un classifieur qui ne comporte aucun apprentissage et qui est implémenté par [Yang, 1999], obtient les plus mauvais résultats.