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

Table des figures

Liste des tableaux

Liste des algorithmes

Remerciement

Mille mercis au professeur Jean-Hugues CHAUCHAT qui a accepté de diriger mes travaux de recherches et qui m'a beaucoup enrichi par ses conseils, son expérience et sa disponibilité. Sa confiance et son soutien m'ont été d'une grande aide.

Je suis très reconnaissant à Madame Geneviève BOIDIN-LALLICH de me faire l'honneur de présider le jury de thèse. J'exprime toute ma gratitude à l'égard de Madame Annie MORIN et de Messieurs Yves KODRATOFF et Martin RAJMAN pour l'intérêt qu'ils ont porté à mon travail et pour avoir accepté la charge d'être rapporteurs. Je remercie également Monsieur Ludovic LEBART de me faire l'honneur de juger mon travail.

Ce travail de recherche n'aurait jamais été terminé sans le soutien et l'encouragement de plusieurs personnes qui n'ont pas hésité à me donner de la force pour l'accomplir. S'engager dans un travail de recherche, qui dure plusieurs années, est une décision difficile.

Cette décision a énormément pesé sur mon entourage, je pense surtout à mon épouse Sawsan et à mes deux fils Edouar et Majd, je pense également à mes parents qui n'ont jamais hésité à me tant donner. C'est à eux que je dédie ce travail.

Entreprendre une thèse en informatique lorsqu'on a fait un Master en économie (littéraire en grande partie) n'est pas habituel. Les premiers mois de la thèse étaient extrêmement difficiles mais motivants.

Je salue mes amis au laboratoire qui m'ont aidé par leurs encouragements à surmonter les difficultés. Je remercie David S. pour ses conseils simples mais efficaces. Je pense à Tiffany pour ses encouragements et à sa gentillesse, à Jérémy, le breton, et à ses coups de gueules énergisants, à Mihaela et Marian pour leur modestie, et à David C. pour son enthousiasme. Je n'oublie pas, bien sûr, Fadila et son bon humeur éternel, Ricco pour ses blagues encourageantes, Fabrice que je ne cesse pas à découvrir et Jérome pour son extrême organisation.

Introduction

La recherche accorde, ces dernières années, beaucoup d'importance au traitement des données textuelles [Lebart, 2001, Kodratoff, 2001] et en particulier aux données multilingues. Ceci pour plusieurs raisons : un nombre croissant de collections mises en réseau et distribuées au plan international, le développement de l'infrastructure de communication et de l'Internet, la progression constante du nombre de personnes connectées au réseau mondial et dont la langue maternelle n'est pas l'anglais [Peters and Sheridan, 2001]. Ceci a créé de nouveaux besoins pour organiser et traiter ces immenses volumes de données. Les traitements manuels de ces données (systèmes experts, ou à base de connaissances) s'avèrent très coûteux en temps et en personnel, ils sont peu flexibles et leur généralisation à d'autres domaines sont quasi impossibles ; c'est pourquoi on cherche à mettre au point des méthodes automatiques [Moulinier, 1996, Sebastiani, 2002].

Dans ce travail nous nous intéressons à l'utilisation de l'apprentissage automatique pour catégoriser (ou classer) les textes. L'apprentissage automatique est un processus d'induction général qui permet la construction automatique de classifieurs [Mitchell, 1997]. Ici, il s'agit d'affecter une ou plusieurs catégories à des documents : l'objectif est de trouver une liaison fonctionnelle, que l'on appelle également modèle de prédiction, entre les textes à classer et l'ensemble des catégories. Pour estimer le modèle de prédiction, il faut disposer d'un ensemble de textes préalablement étiquetés, dit ensemble d'apprentissage, à partir duquel on estime les paramètres du modèle de prédiction le plus performant possible, c'est-à-dire qui produit le moins d'erreurs en prédiction.

L'objectif de la catégorisation de texte est donc d'associer automatiquement une étiquette à tout nouveau texte à classer. Une application typique est le classement automatique de dépêches de presse en différents thèmes (par exemple, "les actualités du monde", "l'économie", "les sports", *etc.*), à l'aide de leurs contenus textuels [Lewis, 1992a].

Dans ce travail, nous étendons la catégorisation de textes à la catégorisation de textes *multilingues*. La nouveauté apportée est la possibilité d'inférer pour un texte rédigé dans une langue quelconque. En reprenant l'exemple précédent, nous apprenons à sélectionner, parmi l'ensemble des dépêches, celles qui concernent l'*économie* ; ceci quelque soit leur langue.

Comme dans le cas monolingue habituel, la phase d'apprentissage s'effectue à partir d'un corpus d'apprentissage étiqueté. Bien entendu, comme pour la catégorisation de textes monolingue, le texte à classer doit appartenir aux mêmes domaines que ceux utilisés lors de l'apprentissage. On ne saurait, par exemple, essayer de classer un article scientifique à partir d'un modèle construit sur un ensemble d'apprentissage constitué d'articles de journaux de mode [Sebastiani, 2002].

Cette extension au cas multilingue introduit des contraintes supplémentaires. Il faut adapter le processus habituellement mis en œuvre pour classer les nouveaux textes ; et certaines techniques à base linguistique, utilisées en monolingue, deviennent alors inopérantes [Biskri and Delisle, 2001].

Plan de la thèse

Notre travail s'intéresse à l'application de méthodes issues de l'apprentissage automatique à la catégorisation de textes multilingues. Il comporte deux parties.

Une première partie donne une présentation générale de la catégorisation de textes :

- les définitions, objectifs généraux et domaines d'application (chapitre 1 page 7).
- l'adaptation des algorithmes d'apprentissage aux spécificités des textes ; représentation vectorielle d'un texte ; choix des unités d'information (mots, lemme, phrase, n-grammes, *etc.*) ; choix des valeurs numériques associées (présence/absence, fréquences, information mutuelle, *etc.*) ; réduction de l'espace de représentation, par sélection de termes, ou bien par extraction (chapitre 2 page 23).
- la méthode de sélection de termes multivariée (chapitre 3 page 35).
- le codage en n-grammes (chapitre 4 page 41) :
  - les avantages et inconvénients des n-grammes, en particulier dans le contexte multilingue,
  - le lien entre les n-grammes et les mots.
- les méthodes d'apprentissage et la mesure de leurs performances (chapitre 5 page 53).
- les tests réalisés pour comparer les algorithmes d'apprentissage sur les textes monolingues : les méthodes des "k-plus proches voisins", des machines à vecteurs de support (SVM), les fonctions à bases radiales (RBF) et les graphes d'induction (sections 5.8.3 page 80 et 7.4 page 109).

La deuxième partie s'intéresse à l'apprentissage de textes multilingues en comparant deux chaînes possibles (chapitres 6 page 87 et 9 page 119) :

- **chaîne 1 :** reconnaissance de la langue, puis utilisation de règles de classement construites pour chaque langue ; il faut alors avoir construit un modèle adapté à chacune des langues.
- **chaîne 2 :** utilisation de la traduction automatique dans le processus de catégorisation ; cette solution permet d'utiliser un seul ensemble de règles de classement. Ici, il y a deux options :

  1. construire un modèle unique sur l'ensemble d'apprentissage d'une langue donnée ; ensuite, pour classer un nouveau texte, (i) reconnaissance de sa langue, (ii) traduction de ce texte vers la langue d'apprentissage, (iii) application du modèle de prédiction sur le texte traduit ; ici la phase de traduction n'intervient que dans la phase de classement.

  2. faire intervenir la traduction automatique dès la phase d'apprentissage : à partir d'un ensemble étiqueté de textes en différentes langues, traduction automatique de tous ces textes vers une langue cible et apprentissage sur cet ensemble de textes traduits ; ensuite, pour classer un nouveau texte, la procédure est la même qu'en 1.

Ces deux chaînes dépendent de la qualité du traducteur automatique (chapitre 8 page 113).

Nous testons nos algorithmes sur des corpus multilingues pour pouvoir évaluer et comparer la performance de ces deux chaînes de traitement (section 9.3 page 122). Nos principales contributions portent sur :

- l'exploration des raisons de l'efficacité de la représentation des textes par le codage en n-grammes ; ce travail a été présenté aux JADT-2002 [Jalam and Chauchat, 2002] ; le chapitre 4 page 41 développe ce travail.
- la proposition d'une nouvelle méthode de sélection de termes multivariée pour l'apprentissage basées sur la statistique χ^2, publiée dans SFdS-2003 [Clech et al., 2003] ; cette proposition est détaillée dans le chapitre 3 page 35.
- une nouvelle utilisation des SVM et des RBF avec le χ^2 comme noyau, publiée dans EGC-2001 et IJCNN-2001 [Jalam and Teytaud, 2001, Teytaud and Jalam, 2001] ; les sections 5.8.3 page 80 et 7.4 page 109 présentent les résultats auxquels nous sommes parvenus avec ces méthodes.
- la proposition d'un nouveau cadre pour la catégorisation de textes multilingues, dans les chapitres 6 page 87 et 9 page 119.
- l'adaptation du corpus CLEF pour disposer d'un échantillon de textes multilingues étiquetés en vue de l'apprentissage ; ce corpus adapté est, à notre connaissance, le premier jeu de données dans ce domaine. Il est présenté au chapitre 9 page 119.

Première partie .

Catégorisation de textes monolingues

Chapitre 1

Catégorisation de textes

Sommaire

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 *applications* 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.

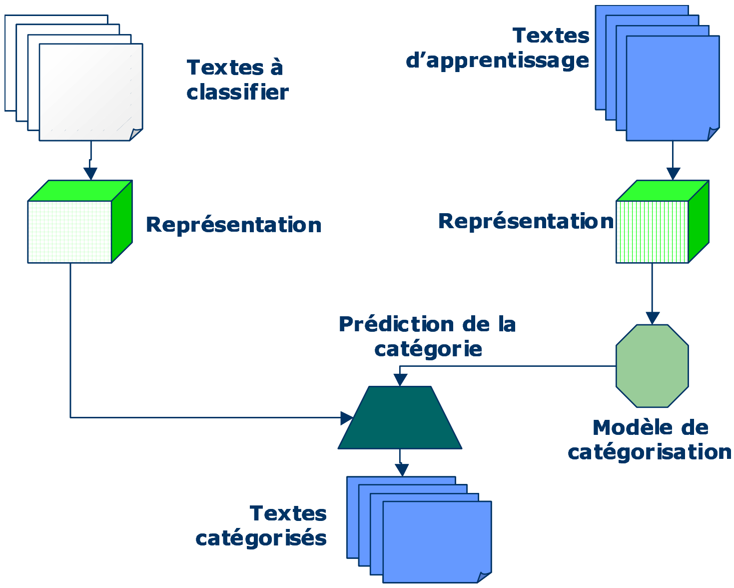


FIG. 1.1.: Processus de la catégorisation de textes

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 30 du chapitre 2, nous présentons en détail ces techniques pour ensuite, dans le chapitre 4 page 41, proposer et justifier notre choix de représentation qui est basée sur les n-grammes. Le chapitre 3 page 35 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 *boosting* [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 75. 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 77 et 5.7.1 page 77.

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 61), le k-PPV (voir section 5 page 66) 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 30.

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 ;

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

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.

1.7. 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/`.

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).

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*

[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 83.

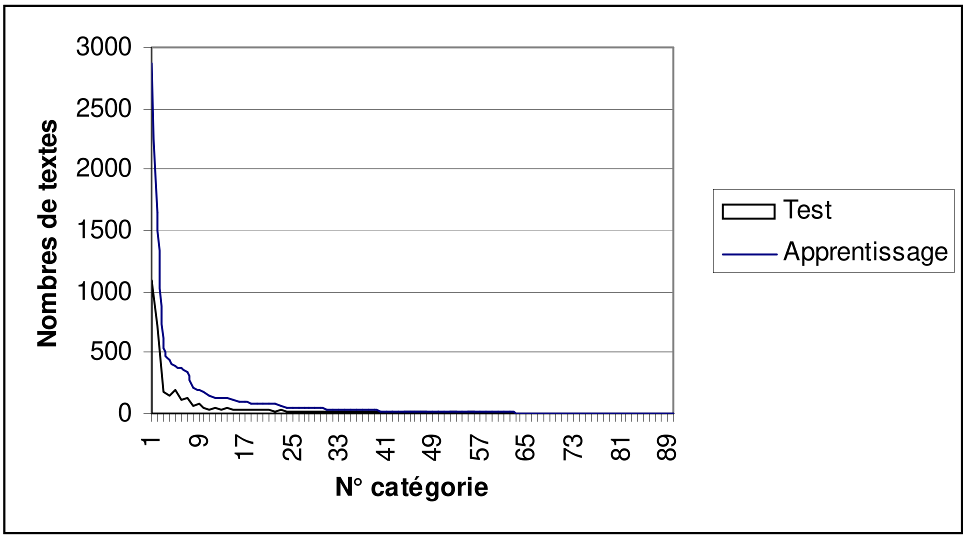


FIG. 1.2.: Nombre de textes par catégories de la collection Reuters. Seules les 20 premières catégories contiennent plus de 100 textes

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

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 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

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 prétraitement linguistique. On peut choisir de conserver les majuscules pour aider,

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

par exemple, à la reconnaissance de noms propres, mais il faut alors résoudre le problème des débuts de phrases.

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.

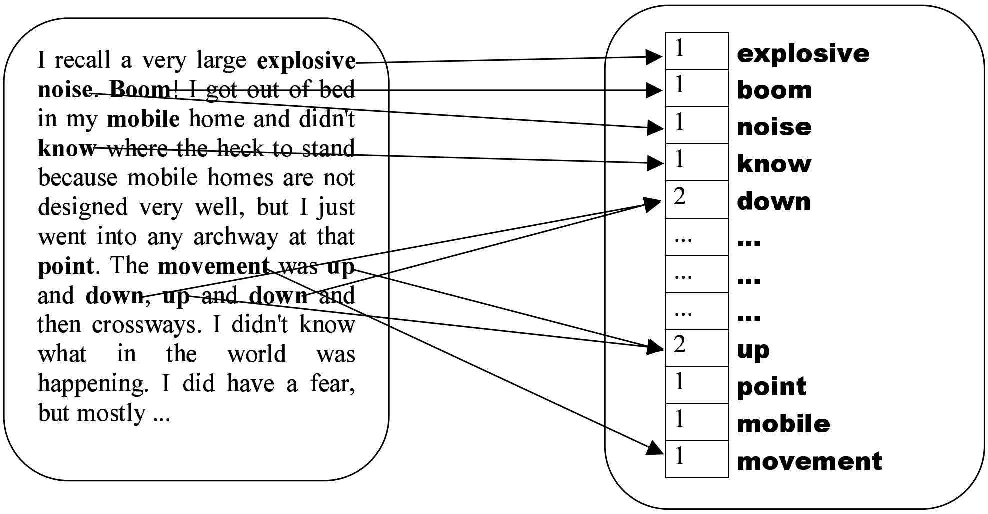


Fig. 2.1.: Exemple de la représentation d'un texte extrait de la collection CLEF en sac de mots (voir la section 9.3 page 122 pour une représentation de cette collection)

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'utiliser 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 page suivante montre quelques suffixes et mots grammaticaux extraits en utilisant ces techniques.
- 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") ;

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 »

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 :

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

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.

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

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).

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

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

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

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

que l'information 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 descripeurs) 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].

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

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 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

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.

	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

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 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

1: **pour** chaque classe $c_i \in \mathcal{C}$ **faire**
2: rechercher tous les termes dans tous les textes d'apprentissage
3: **fin pour**
4: constituer le tableau des N_{ki} nombre d'occurrences du terme k dans la classe i
5: calculer les fréquences f_{ki} relatives : $f_{ki} = \frac{N_{ki}}{N}$
6: calculer les contributions des cellules (ki) à la statistique du χ^2_{multi} : $\chi^2_{ki} = \frac{\left(N_{ki} - \frac{N_{k.} \times N_{.i}}{N}\right)^2}{\frac{N_{k.} \times N_{.i}}{N}} = N \times \frac{(f_{ki} - f_{k.} \times f_{.i})^2}{f_{k.} \times f_{.i}}$
7: calculer le $\chi^2_{ki} \times signe(f_{ki} - f_{k.} \times f_{.i})$
8: trier le tableau des χ^2_{ki} dans l'ordre décroisant
9: **pour** chaque classe i **faire**
10: déterminer la liste $\{terme_{ki}\}$ des K premiers termes de la classe (K est un paramètre de l'algorithme).
11: **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 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×IDF (*Term Frequency×Inverse Document Frequency*), là encore largement utilisé dans les problèmes de catégorisation (voir la section 2.3.1 page 29).

Notre chaîne de traitements consiste donc en trois étapes : 1) sélection de termes (multivariée ou univariée), 2) pondération des termes par le TF×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).

		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

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

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 26 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.

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

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 complète 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

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

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 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 × classes j) **avec un pré-traitement** : élimination des ponctuations et espaces,
   b) calculs des χ^2_{ij} sur le tableau (**mots** i × classes j) **sans pré-traitement** : on laisse les ponctuations et les espaces,
   c) calculs des χ^2_{ij} sur le tableau (**n-grammes** i × classes j) **avec un pré-traitement** : élimination des ponctuations et espaces,
   d) calculs des χ^2_{ij} sur le tableau (**n-grammes** i × 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_] [nes] [arv] [–_] [ghu] [bea] [rai] [cor] [ize] [_us] [hum] [da_] [-_-] [sov] [far] [_so] [arm] [ltu] [ush] [rgh] [gri] [u.s] [.s.] [iet] [_to] [ssr] [ram] [mai] [uga] [por] [ean] [nro] [._c] [mm_] [_mm] [ogr] [vie] [rog] [icu] [huc] [s._] [gnu] [ort] [_u.] [enr] [rtm] [bue] [86/] [xpo] [wee] [_ep] [cul] [6/8] [/87] [cre] [ain] [cer] [nkn] [eat] [ovi] [nup] [aby] [rod] [odu] [pik] [rn,] [kab] [_gr] [rva] [whe] [ric] [ze,] [sug] [epa] [hea] [cro] [sr_] [duc] [liv] [dob] [rn.] [mpo] [rme] [eag] [llm] [fob] [unk] [she] [fre] [-55] [rts] [tot]	[corn] [corn.] [corn,] [tonnes,] [tonnes] [tonne] [tonnes.] [tonne,] [maize] [maize,] [acreage] [acres.] [acres] [washington,] [grain] [program] [program,] [grains] [program.] [usda] [usda's] [usda.] [soybeans] [soybean] [soybeans,] [harvest] [sorghum] [rains] [record] [ussr] [soviet] [soviets] [farmers] [farm] [sources] [agriculture] [agricultural] [bushel] [bushels] [u.s.] [u.s.-ussr] [to] [total] [sugar] [reported] [export] [report] [imports] [exports] [exporters] [import] [reports] [imports.] [enrollment] [u.s. corn] [huckaby] [u.s. agriculture] [department] [1986/87] [week] [between] [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] [...] [rgy] [eum] [leu] [xpl] [els] [sau] [dor] [pet] [ado] [zue] [ezu] [0_b] [nez] [uel] [ira] [aud] [ole] [di_] [bbl] [ec_] [pip] [lor] [cks] [l_p] [pd,] [tpu] [plo] [utp] [gy_] [..1] [odu] [cub] [rod] [ude] [kuw] [uwa] [pd.] [n_b] [i_a] [_cr] [pel] [iea] [rre] [bic] [_ir] [ice] [xxo] [exx] [raq] [bia] [ner] [udi] [/bb] [ara] [ipe] [rab] [ene] [pd)] [mex] [obr] [thq] [hqu] [s/b] [uak] [_op] [duc] [al-] [ref] [(bp] [e_o] [ubi] [fie] [ait] [pec] [tro] [fue] [uot] [_ie] [ora] [ls_] [dri] [quo] [fsh] [efi] [eia] [mob] [as_] [exa] [pdv] [vsa] [dvs] [wai] [_ku] [_ga] [f_o] [ric] [um_] [_bb] [ces] [ukm] [dez] [iel] [urk] [aqi] [try] [xic] [uct] [abi] [naz] [rol] [xac] [a_b] [erg] [eik] [_dr] [put] [prt] [qi_] [c_m] [kh_] [ian] [tex] [ikh] [aeg] [ia_] [ia'] [_pd] [aq_] [rs/] [wti] [_km] [noc] [ope] [ubr] [il,]	[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. 4.4.: 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] [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

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 :

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

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 $\mathbf{d}_\mathrm{j}$ est ensuite normalisé, par la normalisation en cosinus, afin de ne pas favoriser les documents les plus longs.

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

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 :

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

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

[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.

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 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].

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'algorithme procède de façon descendante : il part de la racine puis, récursivement, choisit l'étiquette des fils. L'algorithme 3 page 60 est l'algorithme générique pour les arbres de décision.

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

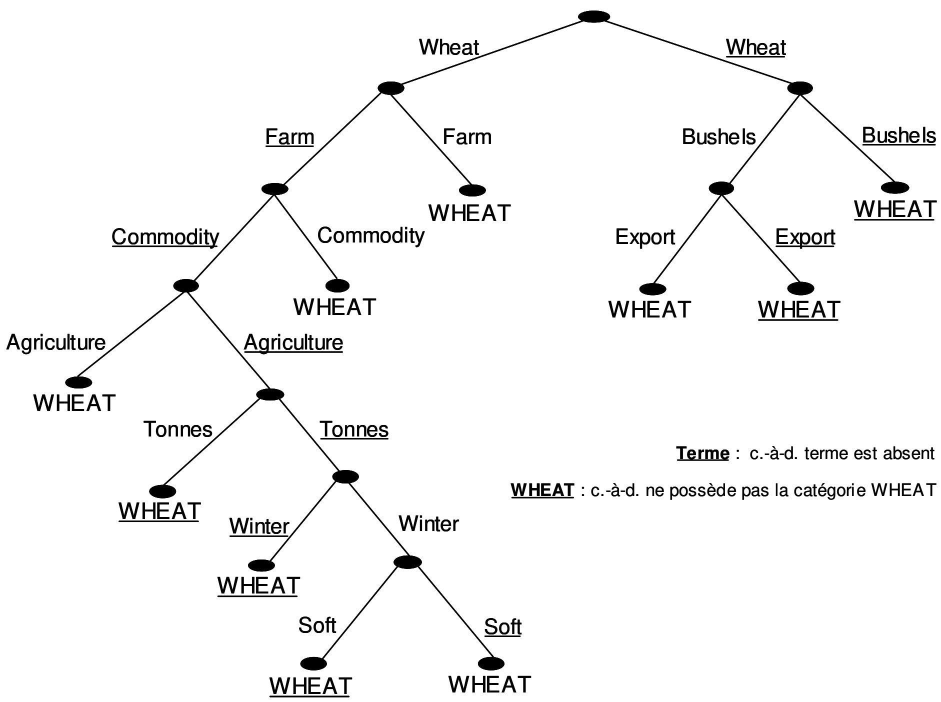


FIG. 5.1.: Exemple d'arbre de décision appliquée sur le corpus Reuters. 'WHEAT' correspond au concept *le document possède la catégorie* 'WHEAT'

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

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

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.

Pour que l'indicateur d'incertitude $I(S)$ soit utilisable lors de la construction

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

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

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

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.

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

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

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

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 plusieurs 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 :

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

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
$$d_e(\mathbf{a},\mathbf{b})=\sqrt{\sum_i (\mathrm{a}_i - \mathrm{b}_i)^2} \tag{5.8}$$
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 **a** est soit **b**, soit **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.

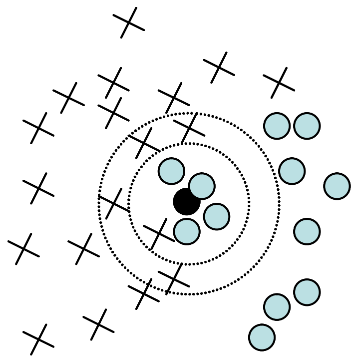


FIG. 5.3.: Choix de k influence la décision : pour $k = 5$, la décision est de classer l'exemple « noir » dans la classe « ronds ». Pour $k = 9$, la décision est de le classer en tant que « croix »

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.

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.

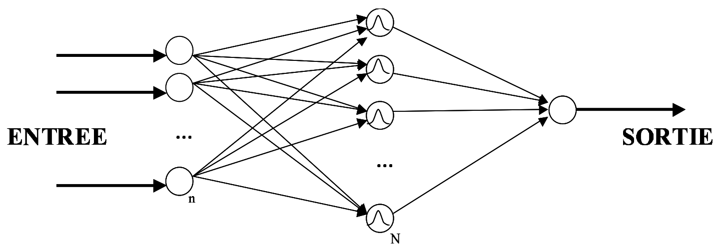


FIG. 5.4.: Chaque neurone RBF contient une gaussienne qui est centrée sur un point de l'espace d'entrée. Pour une entrée donnée, la sortie du neurone RBF est la hauteur de la gaussienne en ce point. La fonction gaussienne permet aux neurones de ne répondre qu'à une petite région de l'espace d'entrée, 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 :

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

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.

5.6.1. Cas des classes linéairement séparables

Soit S un ensemble de l points séparables linéairement, $S = \left\{\mathbf{x}_i \in \mathbb{R}^n \mid i = 1, \cdots, l\right\}$. 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 $b/\left\|\mathbf{w}\right\|$ est la distance de l'hyperplan à l'origine) et $\left\|\mathbf{w}\right\|$ est la norme euclidienne de $\mathbf{w}$; l'hyperplan vérifie l'équation :

[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].

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

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

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 :

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

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).

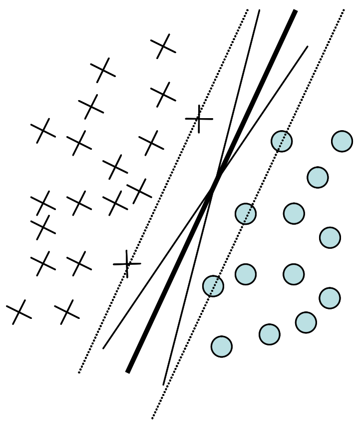


Figure 5.5.: Hyperplans, solutions d'un problème de classification linéairement séparable. L'hyperplan optimal séparant les points de deux classes est celui qui passe au milieu de l'espace entre ces classes. Les exemples les plus proches et qui suffisent à déterminer l'hyperplan optimal sont appelés 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 17).

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 ».

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

[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].

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 Classifieur		Décision du l'expert	
	$C = \{c_1, c_2, ..., c_{\lvert\mathcal{C}\rvert}\}$	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 :

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

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 75 n'est qu'un cas particulier de la matrice d'utilité où les fonctions d'utilité sont $^uVP = {}^uVN > {}^uFP = {}^uFN$.

		Expert	
		c_i	$\neg c_i$
Classifieur	c_i	^uTP_i	^uFP_i
	$\neg c_i$	^uFN_i	^uTN_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.

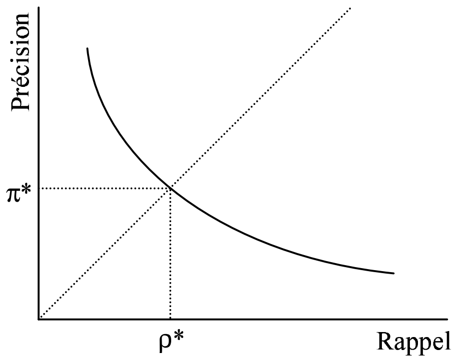


FIG. 5.6.: Courbe de « rappel et précision » et le « point moyen »

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 55).

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 75, 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 109) . 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

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 $\arg\max_k O'(i,k)$.

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).

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% :

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*)

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

[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 :

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

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 109, 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 19) ;
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 ») :

Reuters22173 "ModLewis" $\gg$ Reuters22173 "ModWiener" $>$ Reuters22173 "ModApte" $\approx$
Reuters21578 "ModApte" $>$ 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.

Deuxième partie .

Catégorisation de textes multilingues

Chapitre 6

Catégorisation multilingue : les solutions proposées

Sommaire

6.1. Introduction

La recherche accorde, ces dernières années, beaucoup d'importance au traitement de données multilingues. Les utilisateurs ne se contentent plus d'accéder aux informations et de les manipuler dans leurs langues maternelles, mais ils tendent de plus en plus de franchir le pas vers les autres langues.

La recherche documentaire multilingue est l'une des solutions proposée pour répondre à ces nouveaux besoins. Son objectif est de permettre la formulation d'une requête dans une langue et d'extraire les documents correspondants en différentes langues. Les solutions proposées par les équipes de recherches ont prouvé la faisabilité de l'approche mais aucun moteur de recherche ne l'a offerte à ses clients.

Nous, de notre côté, nous allons proposer et décrire trois solutions pour catégoriser les textes multilingues. Pour ce faire nous faisons appel à deux étapes supplémentaires : l'indentification de la langue et la traduction automatique.

Dans ce chapitre nous allons aborder les facteurs qui ont contribué à l'intérêt accru de traitement de données multilingues, pour ensuite présenter les solutions proposées par d'autres pour le traitement de ces données multilingues ; enfin, nous proposerons notre propre solution qui sera testé dans la section 9.3 page 122.

6.2. Intérêt accru aux traitements multilingues

Plusieurs raisons ont été à l'origine de l'intérêt accru pour les traitements de données multilingues : la disponibilité de plus en plus large des documents mis en réseau et distribués au plan international, le nombre croissant de « non-anglophone » qui se connectent en ligne, la globalisation, la création de zones de coopération entre des pays (Union Européenne, ALENA, Forum Asie-Pacifique, etc.), le développement de l'infrastructure de communication et de l'Internet.

6.2.1. Davantage de collections numériques

La société globale de l'information a radicalement transformé la façon d'acquérir la connaissance, de la disséminer et de l'échanger, provoquant une révolution dans le monde des bibliothèques. La disponibilité des collections, mises en réseau et distribuées au niveau mondial, a créé chez les utilisateurs de nouveaux besoins de trouver, de retrouver et de comprendre une information pertinente, quelle que soient la langue et la forme de stockage [Peters and Sheridan, 2001]. Il est, par exemple, impensable pour l'utilisateur final qui désire interroger une collection multilingues de formuler des requêtes dans toutes les langues.

	Anglais	Non-anglais	Total langu. europe (sauf l'anglais)
Accès à l'Internet (M)	230,6	403,5	224,1
%âge pop. en ligne	36,5%	63,5%	35,5%
2004 (est. in M)	280	657	328
Total pop. (M)	508	5.633	1.218
Produit national brut ($B)	13.812	27.590	14.112
%âge de l'économie mondial	33,4%	66,6%	33,9%

TAB. 6.1.: Répartition de la population mondiale connectée à internet en 2003

6.2.2. Plus de personnes connectées en ligne

Le nombre de langues parlées sur cette planète s'élève à 6.700 dans 228 pays. L'anglais est la langue maternelle de 6% de la population mondiale et pourtant elle est la langue dominante pour les collections et les ressources sur le réseau mondial internet. Néanmoins cette domination est en train de reculer pour ouvrir la voie vers un réseau mondial multilingue [Nunberg, 2000].

En 1998, d'après les chiffres publiés par les organismes internationaux comme L'UNESCO (http://www.unesco.org) et le NUA (http://www.nua.com), environ 60% de la population connectée en ligne parlait (ou lisait) l'anglais et 30% communiquaient par les autres langues européennes. Plus précisément, 147 millions de personnes étaient connectées et, parmi elles, 87 millions aux États Unis et au Canada, 33,5 millions de l'Europe, 22 millions des pays d'Asie, et seulement 800.000 dans les pays d'Afrique.

En 2002, la situation est en train de changer. La table 6.1 montre les nouvelles statistiques confirmant la progression constante des personnes connectées au réseau mondial des autres pays ; nous pouvons constater que la population « non anglaise » représente 63,5 % (403 millions) et la population dont l'anglais est la langue maternelle ne représente désormais que 36,5% (230,6 millions)[1].

6.2.3. Plus de globalisation et de pays unifiés

La globalisation du monde s'accentue. Plusieurs zones de coopération, voire d'union, sont en cours de création. L'unification et l'élargissement de l'Europe, qui a pour objectif d'ouvrir le marché entre les pays et de créer un espace de coopération politique européen, est à l'origine de plusieurs projets de recherches sur le multilinguisme. Parmi ces projets nous pouvons citer le projet EMIR

[1] Ces statistiques sont extraites en grande partie à partir de cette page : http://www.nua.com/surveys/index.cgi?f=VS&art_id=905358509&rel=true

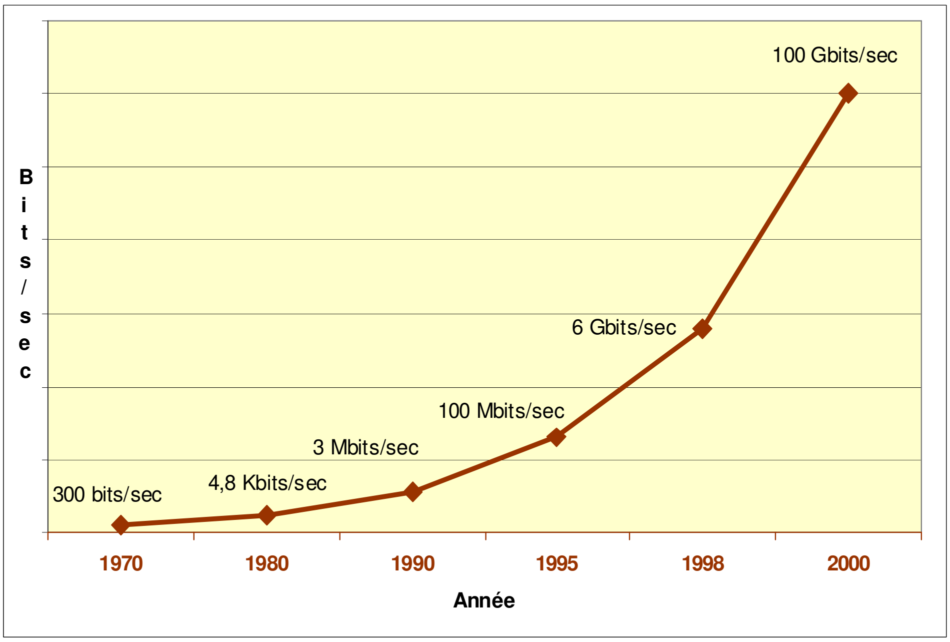


FIG. 6.1.: Évolution de la vitesse de transmission des réseaux

(*European Multilingual Information Retrieval*) qui a lancé le système commercial SPIRIT supportant les langues française, anglaise, allemande, hollandaise et russe. L'UNESCO a également financé des projets comme le projet MEDLIB, pour démocratiser l'accès au patrimoine culturel mondial de la Méditerranée.

6.2.4. Réseau plus rapide et plus souple

La vitesse de transmission des réseaux, comme le montre la figure 6.1, est une vraie révolution. L'accès distant à des masses considérables de données pour un coût quasi dérisoire conduit à une duplication des données par copie. En une décennie, nous sommes passés d'une ère où les données étaient rares, et où leur regroupement dans une base de données leur donnait une valeur économique intrinsèque, vers une ère d'abondance de données où la valeur de celles-ci s'est complètement diluée au profit de l'information ou de la connaissance potentielle qu'elles renferment [Zighed and Rakotomalala, 2002].

6.3. Recherche documentaire multilingues

Les facteurs présentés ci-dessus ont aidé à la disponibilité de l'information multilingue en format électronique et ont provoqué de nouveaux besoins. Ces besoins concernent à la fois : - la manière de rechercher une information en langue étrangère et - la façon d'accéder à cette information écrite dans différents alphabets et/ou différents codages de caractères. Ici, nous nous limitons à présenter les solutions proposées par les chercheurs pour la recherche documentaire multilingue.

Comme il est dit dans la section 1.6 page 17, la recherche documentaire est le processus par lequel l'utilisateur cherche à localiser les documents dont le sujet correspond à ses besoins, exprimés par une requête.

Beaucoup d'utilisateurs ont une connaissance partielle de langues étrangères, mais leur capacité est insuffisante pour formuler une requête dans ces langues [Oard and Dorr, 1996]. Il y a bien d'autres circonstances dans lesquelles l'utilisateur, non familier avec la langue principale d'une collection, peut utiliser la recherche d'information multilingue ; par exemple :

1. la recherche dans une collection d'images indexées et annotées dans une langue non familière,
2. la recherche des organismes, ou des individus, qui s'intéressent à un certain domaine (veille technologique),

La disponibilité de moyens de traduction automatique permettant à l'utilisateur de traduire des textes sélectionnés développe l'intérêt de la recherche d'information multilingue.

La recherche documentaire multilingue a pour objectif de permettre à l'utilisateur de formuler une requête dans sa langue maternelle afin de trouver les documents dans plusieurs langues.

A ce jour, trois familles d'approches pour la recherche documentaire multilingue ont été proposées [Fluhr, 1995] : les approches basées sur la traduction automatique, les approches basées sur les thésaurus multilingues et les approches basées sur les dictionnaires.

6.3.1. Approches basées sur la traduction automatique

Les objectifs d'un système de recherche documentaire et d'un système de traduction automatique ne sont pas les mêmes. La recherche multilingue vise à trouver les documents, dans une langue cible, les plus proches au sens d'une mesure de similarité, de la requête dans la langue d'origine. La traduction automatique, de son côté, vise à produire une version lisible et fiable d'un document d'une langue d'origine vers la langue cible.

La similarité et la lisibilité sont deux objectifs différents ; pour reproduire le sens, le traducteur automatique n'utilise pas forcement le même vocabulaire que celui utilisé dans la requête.

La traduction d'une collection de documents dans la langue de requête (la langue cible) est coûteuse. Les recherches fondées sur un système de traduction automatique se sont donc concentrées sur la traduction des requêtes plutôt que celle des documents [Peters and Sheridan, 2001]. Les requêtes sont en général un ensemble de mots avec peu ou pas de structure syntaxique. De ce fait, l'analyse grammaticale et les méthodes d'identification de polysémie ne peuvent s'appliquer sur ces requêtes car celles-ci ne sont pas sémantiquement cohérentes. De nombreux travaux ont montré que les techniques fondées simplement sur des dictionnaires peuvent être plus efficaces que les systèmes de traduction automatique [Ballesteros and Croft, 1998].

6.3.2. Thésaurus multilingues

Le vocabulaire contrôlé est fréquemment utilisé pour indexer et décrire un document. Il s'agit d'un ensemble fini de concepts que l'on doit associer aux textes d'une collection. Ainsi, un document sera représenté par un ou plusieurs concepts.

Un thésaurus multilingue est une extension de la notion de vocabulaire contrôlé monolingue. Un thésaurus multilingue peut être vu comme un ensemble de thésaurus monolingues qui englobent tous un même système de concepts. Les utilisateurs, pour rechercher une information écrite dans d'autres langues, formulent leurs requêtes dans leurs langues maternelles puis le système, automatiquement, réalise en interne la relation entre terme et descripteur [Soergel, 1997].

Notons que cette approche souffre des limites relatives au vocabulaire contrôlé : très chers à construire, coûteux à maintenir et difficile à mettre à jour ; il faut ajouter les efforts nécessaires pour former les utilisateurs qui doivent utiliser de tels systèmes [Peters and Sheridan, 2001].

6.3.3. Utilisation de dictionnaires

L'utilisation de traducteurs automatiques pour traduire les requêtes a montré ses limites car l'absence de structure syntaxique de requête empêche l'analyse grammaticale et n'aide pas, par conséquent, à lever l'ambiguïté. D'autres solutions, comme l'utilisation de dictionnaires bilingues informatisés, sont proposées. Ces outils sont de plus en plus disponibles en ligne. Certains auteurs ont utilisé les dictionnaires pour traduire les requêtes et ont obtenu de résultats acceptables mais ils sont bien inférieurs à ceux obtenus par la recherche monolingue. [Ballesteros and Croft, 1998, Hull and Grefenstette, 1996] obtiennent, à titre d'exemple, des performances de 40% à 60% inférieurs à celles obtenue par une recherche monolingue. [Peters and Sheridan, 2001] reportent trois raisons principales qui expliquent cette situation : (i) les dictionnaires généraux n'incluent pas tous les vocabulaires spécialisés ; (ii) échec de la traduction de termes constitués de plusieurs mots (exemples : sécurité sociale, Armée du Salut, chemin de fer) ; (iii) problème de l'ambiguïté.

Finalement, le problème de la recherche documentaire multilingue a-t-il trouvé sa solution ? D'après [Oard, 2002], la réponse est oui et non. La réponse est oui puisque plusieurs équipes de recherches ont montré la faisabilité de l'approche. La réponse est non puisque aucun moteur de recherche, ou service de recherche commerciale (comme Dialog), n'offre ce service à ses clients. En effet, nombre de problèmes sont encore sans solution, tels que :

- **La généralité** : la recherche documentaire multilingue a été testée avec succès sur certains types de textes, et dans certains domaines, mais le problème est loin d'être réglé pour d'autres domaines comme les brevets, les résumés de papiers scientifiques, *etc.*
- **L'efficacité** : peut-on fournir ces services avec des coûts comparables à ceux de la recherche monolingue ? La réponse est non car ceci impose des coûts importants de temps d'accès aux donnée et de traduction de requêtes et/ou de documents.
- **L'interaction** : lorsqu'un système de recherche monolingue fournit une liste ordonnée de réponses pour une requête, cette liste fixe les frontières de recherche pour un utilisateur et cette liste devient utilisable telle quelle. Mais fournir une liste qui pointe vers des pages multilingues est beaucoup plus difficile à utiliser. Ainsi, il faut viser à proposer des moyens permettant plus d'interaction entre utilisateur et système de recherche multilingue.

6.4. Nos solutions pour catégoriser des textes multilingues

Nous proposons maintenant un cadre pour la catégorisation de textes multilingues. L'objectif est d'apprendre à inférer sur des textes rédigés dans une langue quelconque, à partir d'un modèle de prédiction construit sur un corpus de textes rédigés dans un langue donnée. Par rapport à la généralisation classique, la phase d'inférence comprend deux étapes supplémentaires :

1. la détection automatique de la langue du texte,
2. puis la traduction automatique du texte vers la langue de référence.

Nous supposons avoir $\mathfrak{L}$ corpus de textes, chaque corpus $\mathfrak{L}_l$ comportant des textes écrits dans la langue $\mathfrak{L}_l$ (nous utiliseront $\mathfrak{L}_l$ pour désigner le corpus et la langue du corpus) avec $l = 1, ..., |\mathfrak{L}|$; d_{jl} désigne un texte d_j qui fait partie du corpus $\mathfrak{L}_l$ (dont la langue est L_l) ; chaque texte d_{jl} peut être associé à une ou plusieurs classes $c_i \in \mathcal{C}$. Notre objectif est de pouvoir associer une classe c_i à chaque texte d_{xy}, dont la langue et la classe sont a priori inconnues. Il est évident que les classes des corpus doivent êtres comparables c'est pour cette raison que nous utilisons un seul ensemble de classes $\mathcal{C}$.

Nous allons présenter trois schémas.

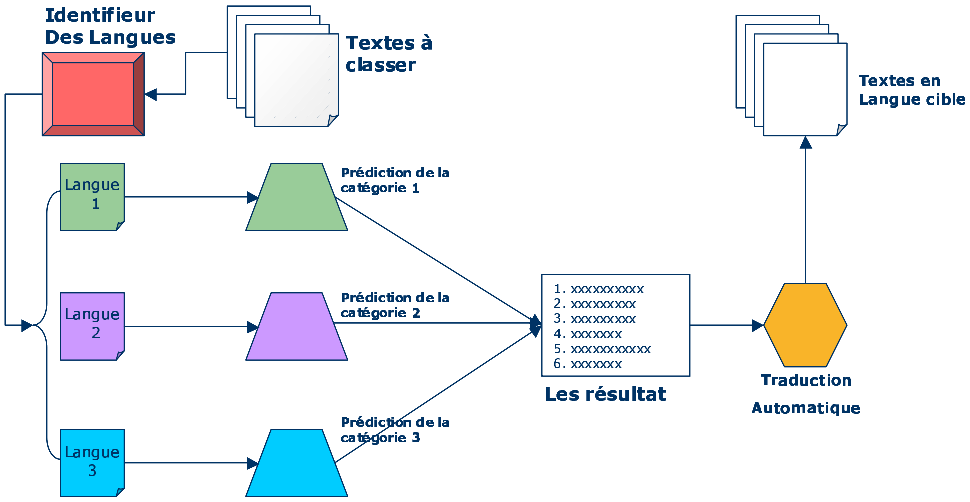


FIG. 6.2.: Phase de classement selon le schéma trivial (extension de schéma monolingue) : apprendre un modèle pour chaque langue et ainsi il faut apprendre $|\mathfrak{L}|$ modèles

- Le premier, nommé le schéma « trivial », représente une extension naïve du schéma de catégorisation monolingue habituel ; Il consiste en l'apprentissage de $|\mathfrak{L}|$ modèles (un modèle pour chaque langue).
- Le deuxième est un schéma permettant l'apprentissage d'un seul modèle.
- Le troisième consiste en la traduction des textes de plusieurs ensembles d'apprentissage vers une langue cible pour ensuite apprendre un seul modèle « mixte ».

Nous allons maintenant détailler ces trois schémas.

6.4.1. Premier schéma : le schéma trivial

Nous appelons ce schéma « trivial » car il s'agit d'une extension directe de la catégorisation « monolangue ». La figure 6.2 décrit le processus de catégorisation. Pour chacun des $\mathfrak{L}$ corpus (langues) nous apprenons un modèle, soit $|\mathfrak{L}|$ modèles à construire. Pour chaque texte à classer 1) nous identifions sa langue puis 2) nous appliquons le modèle de prédiction appris pour cette langue. Si le texte est identifié comme « intéressant » nous le passons à un traducteur automatique afin de le traduire vers la langue cible (par exemple le français). L'avantage avec ce schéma est l'intervention de la traduction à la fin de processus : le traducteur n'intervient pas dans l'apprentissage et, par conséquent, aucune distorsion d'information ni perte n'est commise à cette étape.

Mais ce schéma présente de sérieuses limites :

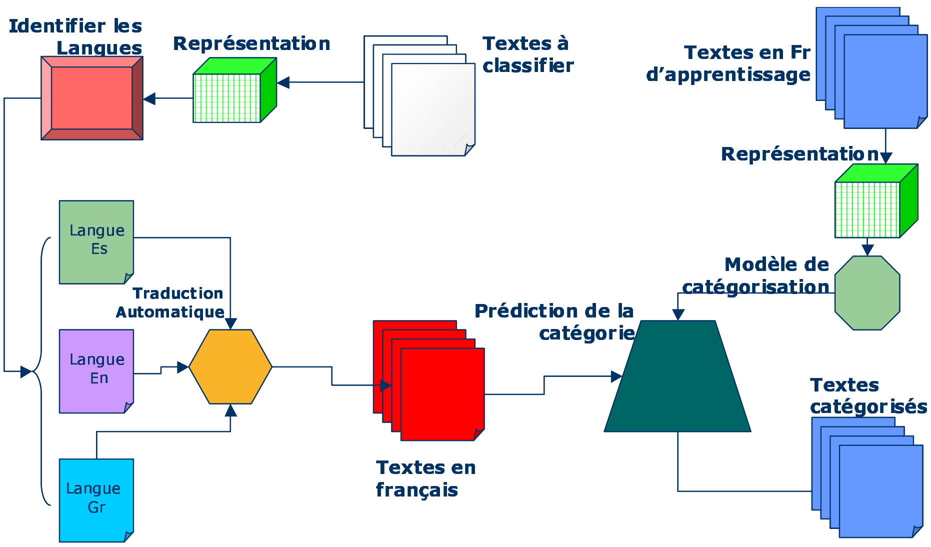


Fig. 6.3.: Deuxième schéma : choix d'une langue d'apprentissage et utilisation d'un seul modèle, dans cette langue

- il exige de faire $|\mathfrak{L}|$ apprentissages, un par langue,
- et ceci suppose d'avoir des quantités suffisantes de textes étiquetés dans chaque langue et dans chaque classe. Ceci est difficile surtout pour les langues peu présentes sur le Web.

Mais ce schéma sera notre référence pour valider les autres solutions proposées maintenant.

6.4.2. Deuxième schéma : choisir une seule langue d'apprentissage

Le deuxième schéma que nous proposons pallie en partie les deux critiques précédentes, car nous utilisons ici un seul modèle de prédiction. Pour chaque texte à classer nous identifions d'abord sa langue ; si cette langue est supportée par le traducteur automatique, nous traduisons le texte dans la langue d'apprentissage (la langue du modèle de prédiction) ; enfin, nous appliquons le modèle de la langue d'apprentissage sur ce texte pour le classer. Dans ce schéma le traducteur joue un rôle primordial dans la phase de classement (voir la figure 6.3).

6.4.3. Troisième schéma : mélanger les ensembles d'apprentissage

Dans le deuxième schéma, la traduction intervient uniquement dans le processus de classement. Il nous semble intéressant de tester une solution où la

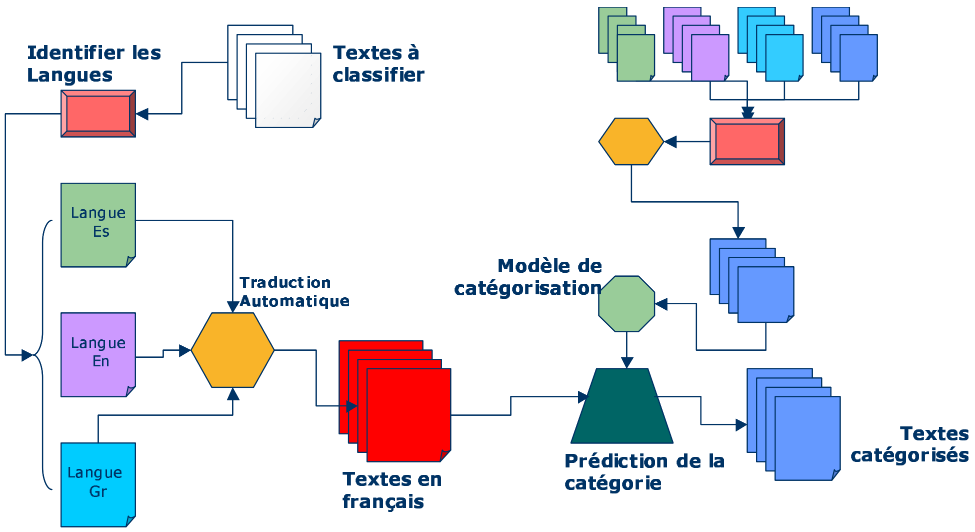


FIG. 6.4.: Troisième schéma : réunion des ensembles d'apprentissage après avoir traduit tous les corpus vers une seule langue, et apprentissagr d'un seul modèle, adapté aux textes traduits

traduction intervient dans les deux processus : apprentissage et classement.

Disposant d'un ensemble de textes (étiquetés) écrits dans différentes langues, on les traduits tous dans une langue d'apprentissage $\mathfrak{L}_{app}$ unique, puis on les réunit en un corpus unique, sur lequel on procédera à l'apprentissage des étiquettes (classes). Ainsi le modèle sera appris à partir des textes produits par les traducteurs automatiques et il sera plus à même de classer un nouveau texte, traduit par le même outil.

Ensuite, pour chaque texte à classer, 1) nous identifions sa langue ; 2) si la langue est prise en charge par le traducteur, nous le traduisons vers la langue cible $\mathfrak{L}_{app}$, et 3) nous appliquons le modèle (voir la figure 6.4).

Nous somme actuellement en train de tester ce schéma. Nous espérons qu'il donnera de bons résultats car les traducteurs ont tendance à « agréger » les vocabulaires et ainsi à fournir un corpus homogène.

6.5. Conclusion

De nombreux facteurs concourent à la disponibilité de textes en plusieurs langues. Les utilisateurs ne se contentent plus d'utiliser leur langue maternelle, mais ils s'intéressent de plus en plus aux textes écrits dans d'autres langues. De nouveaux outils sont apparus pour répondre à ces besoins émergeants. La recherche documentaire multilingue, à titre d'exemple, doit de permettre à l'utilisateur de formuler une requête dans sa langue d'origine et de lui fournir des textes écrits dans d'autres langues.

Nous proposons des méthodes de catégorisation de textes, queles que soient leurs langues ; ces méthodes peuvent sembler naïves, mais elles donnent d'assez bons résultats comme nous le verrons au chapitre 9 page 119. Les trois schémas proposés font appel à deux étapes supplémentaires pour le classement et/ou l'apprentissage de textes : l'identification de la langue et la traduction automatique. Ces deux étapes sont importantes et les bonnes performances des modèles d'apprentissage sont directement affectées par ces deux étapes.

Dans les deux chapitres suivants page 99 et page 113, nous allons présenter les méthodes d'identification automatique de la langue d'un texte puis de traduction automatique. Le chapitre 9 page 119 présentera nos expérimentations sur les deux premiers schémas discutés dans les sections 6.4.1 et 6.4.2 ci-dessus.

Chapitre 7

Identification de la langue

Sommaire

7.1. Introduction

L'identification de la langue consiste à attribuer une unité textuelle, supposée monolingue, à une langue. Cette identification est devenue importante puisque les données textuelles, en différentes langues, trouvent de plus en plus leur chemin sur le réseau mondial. En outre la bioinformatique fournit de vastes problèmes traitables par des outils similaires.

Dans ce chapitre nous allons identifier les facteurs à considérer, puis nous allons présenter les techniques utilisées et proposer deux nouvelles techniques, la première basée sur la mesure de distance de χ^2 et la deuxième sur les RBFs.

L'identification automatique de la langue est possible car (i) les langues naturelles écrites sont extrêmement non-aléatoires ; (ii) elles présentent chacune des régularités dans l'utilisation des caractères ou séquences de caractères ; (iii) l'alphabet de chaque langue est soit unique, soit très caractéristique de cette langue. Les informations sur la stabilité et la constance des fréquences de lettres et séquences de lettres ne sont pas nouvelles ; elles sont connues depuis des centaines d'années ; Kahn, dans son survey sur la cryptographie, mentionne que ces régularités ont été reportées dans une encyclopédie arabe écrite par Qalqashandi en 1412 qui, à son tour, attribue la majorité de ces informations à Ibn ad-Duraihim qui vivait entre 1312 et 1361 [Beesley, 1988].

Il est prouvé statistiquement que, pour chaque langue, les nombres d'occurrences des séquence de deux, trois, quatre ou cinq lettres sont stables et différents des autres langues. Par exemple, en anglais, dans un texte quelconque, la fréquence de la lettre « E » est d'environ 13%, la fréquence de la lettre « U » d'environ 3% et la fréquence de la lettre « Z » d'environ 0.1%. Pour les séquences de deux caractères ou bi-grammes, on trouve par exemple que la probabilité d'avoir la chaîne « TH » en anglais est relativement grande ; en espagnol et portugais, cette probabilité s'approche de zéro. Dans le même ordre d'idées, la probabilité d'avoir « SZ » en hongrois et polonais est grande ; la chaîne « TION » caractérise le français et l'anglais.

Partant de ces probabilités d'apparition des lettres et séquences de lettres, on peut concevoir un algorithme capable d'identifier la langue d'un texte.

Plusieur approches peuvent être utilisées lors la conception de méthodes d'identification. A titre d'exemple, [Sibun and Reynar, 1996] considèrent les distinctions suivantes :

1. Le type des éléments significatifs : un caractère, un n-gramme, ou d'autres ? Utilise-t-on des connaissances linguistiques ou non (possibilité d'adaptation à d'autres problèmes, par exemple ceux issus de la bioinformatique) ?
2. La forme de l'analyse : quel algorithme faut-il utiliser pour identifier la langue ? Certains algorithmes demandent une intervention humaine alors que d'autres sont totalement automatisés. Certains sont basés sur l'exis-

tence, ou non, de certaines séquences (mots courts), tandis que d'autres s'intéressent aux fréquences des séquences.

3. La forme de codage : quelle est la meilleure représentation du point de vue des critères de rapidité, robustesse et précision ? Doit-on utiliser un codage compréhensif ou un codage simplifié (en supprimant les caractères accentués) ?
4. Le choix de l'univers des langues à identifier.
5. La taille du texte : il est souvent nécessaire d'identifier la langue à partir d'un court extrait (une phrase, voire quelques mots). Dans les systèmes de recherche documentaire, il est important de détecter les passages multilingues afin de construire un index pour chacune des langues. Si l'identification sur de courts passages est satisfaisante, alors, pour les documents longs on a intérêt à analyser quelques échantillons de texte au lieu d'analyser le texte tout entier.
6. La forme des entrées : textes en ligne, images en ligne ou les deux.
7. Le choix de la méthode statistique : on peut les comparer selon divers critères. La performance ne doit pas se dégrader si on ajoute plus de données pour l'apprentissage, ou si le texte à identifier devient plus long (par exemple, l'identification d'une langue ne doit pas être infectée par la présence ou l'absence d'autres langues. On doit pouvoir tenir compte d'informations de fréquences ; [Cavnar and Trenkle, 1994] utilisent une méthode qui calcule la distance en fonction de la position des n-grammes selon leur fréquence, si deux langues possèdent les mêmes rangs alors la méthode de Cavnar n'arrivera pas à les distinguer bien que les fréquences des n-grammes ne soient pas les mêmes.

Plusieurs éléments significatifs peuvent être considérés pour l'identification de la langue : la présence de certains caractères [Mustonen, 1965] [Souter et al., 1994], la présence de certains mots [Souter et al., 1994] [Giguet, 1998] ou la fréquence de n-grammes [Beesley, 1988] [Cavnar and Trenkle, 1994] [Dunning, 1994] [Grefenstette, 1995].

Nous distinguerons deux familles d'approches : les approches linguistiques (en section 7.2) et les approches statistiques et probabilistes (en section 7.3).

7.2. Approches linguistiques

Ces approches d'identification nécessitent une connaissance linguistique préalable et ces connaissances linguistiques sont intégrées dans le programme informatique.

7.2.1. Présence de certains chaînes de caractères spécifiques

On repère des chaînes de caractères qui sont uniques pour chaque langue. Par exemple, le tableau 7.1 montre une liste proposée par plusieurs auteurs (cité dans [Dunning, 1994]).

Langue	Chaîne
Néerlandais	« vnd »
Anglais	« ery_ »
Français	« eux_ »
Gaélique	« mh »
Allemand	« _der_ »
Italien	« cchi_ »
Portugais	« _seu_ »
Serbo-croate	« lj »
Espagnol	« _ir_ »

TAB. 7.1.: Quelques chaînes de caractères spécifiques à une langue [Dunning, 1994]

Cette approche est valide intuitivement, mais pas assez pour établir des règles de décisions sûres [Souter et al., 1994] et [Dunning, 1994] : on ne peut pas attribuer la langue italienne à tout texte qui parle de « Zucchini » ou de « Pinocchio », ni associer la langue française à un texte anglais qui cite le mot français « milieux ». De plus, ces chaînes de caractères uniques sont relativement rares et par conséquent ils peuvent être absents dans des textes courts (taille 50 ou 100 octets par exemple). De toute façon, comme nous le verrons plus loin (voir section 7.3.2 page 106), les approches basées sur les n-grammes capturent aussi ce type de connaissances.

7.2.2. Présence de certains mots

Une autre approche consiste à choisir les mots grammaticaux comme discriminants pour l'identification de la langue [Grefenstette, 1995]. L'utilisation de ces mot grammaticaux (prépositions, conjonctions, déterminants, pronoms, adverbes en liste close, auxiliaires) se justifie par le fait qu'ils permettent un diagnostic sûr : 1) ils représentent en moyenne 50% des mots d'une phrase dans la plupart des langues et que leur présence est indispensable [Giguet, 1998], 2) ces mots grammaticaux ont la propriété d'être courts et en nombre limité ce qui permet la construction de listes exhaustives.

Cette approche est assez efficace : elle obtient des scores supérieurs à ceux obtenus par l'approche précédente (la présence de chaînes de caractères uniques) et elle est rapide par rapport à d'autres méthodes basées sur les n-grammes que nous aborderons plus loin ; elle demande moins de calculs. Mais les performances sont mauvaises sur les énoncés très courts (comme les titres des sections) qui ne contiennent pas forcément ces mots courts [Grefenstette, 1995]. Et il faut évidement que les langues à reconnaître soient segmentées en mots (nous abordons ce point en section 7.3.1).

L'identification de la langue n'est pas un objectif en soi ; elle fait souvent partie d'un processus plus complexe comme la recherche documentaire, ou l'indexation automatique des documents multilingues. Les mots grammaticaux dans un tel processus ne rapportent pas beaucoup d'informations ; au contraire, les chaînes significatives sont généralement les plus rares et les plus longues. Les mots courts, employés pour fournir une structure syntaxique de la langue, sont utilisés indépendemment du contenu et les systèmes de recherche documentaire et d'indexation automatique éliminent ces mots. La recherche de mots courts pour identifier la langue implique donc une étape de traitement supplémentaire. Un système basé sur les n-grammes n'aura pas ce genre de problème (voir section 2.2.4 page 27).

7.2.3. Approche lexicale

Si on définit la langue d'un énoncé par la langue des mots qui le composent, le simple recours à des lexiques de mots est suffisant pour identifier la langue : il suffit de reconnaître la langue du segment comme étant celle dont le lexique contient tous les mots [Giguet, 1998].

Mais cette approche ignore l'incomplétude des lexiques : de nombreux termes techniques, particuliers à certaines domaines, ne figurent jamais dans des lexiques généraux. Cette approche suppose par ailleurs l'absence de fautes dans les énoncés, fautes d'orthographe, fautes de frappe, qui sont très courantes et qui perturbent la reconnaissance des mots.

7.2.4. Approche plus linguistique

Cette approche s'intéresse aux propriétés positives intrinsèques des langues : par ce que l'on trouve obligatoirement ou très souvent dans une langue, et non par opposition aux autres langues. Elle choisit les mots grammaticaux, l'alphabet, les affixes fréquents, comme outils discriminants pour l'identification de la langue en les présentant comme porteurs de la langue de l'énoncé [Giguet, 1998].

Dans la section 7.2.2, nous avons présenté les avantages de l'utilisation des mots grammaticaux. L'utilisation de l'alphabet se justifie par le fait que si le mot ne peut s'écrire dans l'alphabet d'une langue alors il n'en fait pas partie. La vérification d'appartenance de toutes les lettres d'un mot à un alphabet n'est pas suffisante pour identifier une langue, mais cela peut renforcer d'autres techniques, en écartant certaines langues.

L'utilisation des affixes (préfixes et suffixes) peut contribuer à discriminer une langue ; ceci nous rappelle l'approche étudiée en section 7.2.1. Comme la technique précédente, la technique des affixes ne suffit pas pour identifier la langue, mais elle peut être utilisée en combinaison avec d'autres techniques.

Ces techniques (mots grammaticaux, l'alphabet, et les affixes fréquents) se distinguent de l'utilisation des mots les plus fréquents (technique illustrée plus loin) qui s'acquièrent sur des corpus d'apprentissage (lourds à construire) et qui sont tributaire du choix du nombre de mots à considérer comme faisant partie des plus fréquents [Péry-Woodley, 1995]. [Déjean, 1998] propose une méthode multilingue d'extraction automatique de ces mots sur des corpus non annotés, pour la constitution de ces ressources linguistiques (les mots grammaticaux, les suffixes).

Toutes ces approches demandent une acquisition non triviale des connaissances linguistiques sur toutes les langues, tâche coûteuse et impossible à réaliser pour toutes les langues. Ces approches ne sont justifiées que si l'identification de la langue constitue une étape d'un processus d'analyse linguistique plus complexe, qui s'intéresse à la compréhension et la modélisation des phénomènes linguistiques et vise à maîtriser leur application.

Finalement, il nous semble que cette approche (l'approche plus linguistique) n'est qu'une version plus développée des autres approches présentées en sections 7.2.1 et 7.2.2 car les mots grammaticaux, l'alphabet et les affixes qui discriminent une langue peuvent être vues comme des chaînes uniques appartenant à cette langue.

7.3. Approches statistiques et probabilistes

Ces approches utilisent des ressources construites automatiquement à partir d'un corpus textuel représentatif de la langue. L'objectif est de capturer au moyen de modèles statistiques ou probabilistes certaines régularités formelles des langues et d'estimer leurs probabilités d'apparitions. Ces régularités empiriques jouent le rôle de connaissances linguistiques. L'identification consiste à calculer la probabilité pour un énoncé d'appartenir aux différentes langues, en fonction des régularités observées. Deux principaux types de régularités sont couramment exploités : les mots les plus fréquents, et les séquences de n-grammes les plus fréquentes.

7.3.1. Mots les plus fréquents

Cette approche est plus générale que celle étudiée en section 7.2.2. L'idée de base est proche : détecter empiriquement la présence de certains mots très présents dans une langue. Mais ici la démarche est purement statistique et la construction des listes ne demande aucune connaissance linguistique préalable ; la détection de ces mots se fait en calculant les fréquences d'apparition dans un corpus de textes et on retient les mots dépassant une fréquence donnée. Cette approche semble intuitivement valide : il est de bon sens d'attribuer la langue anglaise à un passage de texte qui contient beaucoup de mots comme *are*, *that*, *the*, *and*, *of*, ... et d'attribuer la langue espagnole à un passage de texte qui comporte des mots communs comme *los*, *del*, *por*, *para*, ..., etc. La table 7.2 montre les mots les plus fréquents, pour dix langues européennes, calculées à partir du corpus ECI [Grefenstette, 1995].

Dan	**Dut**	**Eng**	**Fre**	**Ger**	**Ita**	**Nor**	**Por**	**Spa**	**Swe**
i	de	the	de	des	di	.	de	de	och
af	van	and	la	die	e	og	a	la	i
og	het	to	le	und	il	det	que	que	att
at	een	of		den	che	,	o	el	som
ğ	en	a	et	in	la	han	e	en	en
til	in	in	des	von	a	i	do	y	r
for	dat	was	les	.	in	er	da	a	p
en	is	his	du	zu	per	"	no	los	det
om	te	that	"	dem	del	p	um	del	av
der	op	I	en	,	un	til	em	se	fr
er	voor	he	un	fr		at	para	por	med
U	met	as	que	mit	non	som	com	las	den
ikke	die	had	a	das	i	var	se	con	till
eller	De	with	qui	des	si	jeg		un	har
som	zijn	it	dans	ist	le	med	os	para	de

TAB. 7.2.: Mots les plus fréquents d'après le « ECI Multilingual Corpus »

Ensuite, le principe d'identification de la langue est simple : à partir de la liste des mots retenus pour chaque langue, la fréquence d'apparition de chaque mot m est transformée en fréquence relative. Pour identifier la langue d'un texte, on découpe ce texte en mots. Pour ceux qui apparaissent dans la liste on leur attribue les probabilités correspondantes et pour ceux qui n'apparaissent pas on leur attribue des probabilités minimales puis on calcule le produit des probabilités. La langue de ce texte est celle pour laquelle ce produit est maximal. Cette technique est utilisée par [Grefenstette, 1995] et, bien avant, par [Beesley, 1988] (voir page 106).

Cette approche est simple, intuitive et relativement efficace, mais elle présente quelques inconvénients.

Premièrement, ses performances se dégradent lorsque les textes à identifier deviennent courts du fait qu'ils ne contiennent pas forcément des mots les plus fréquents de leur langue. [Cowie et al., 1998] montre que les performances des approches basées sur les mots se dégradent considérablement lorsque la taille du texte devient petit : les taux d'erreurs sont autour de 20% pour des textes de 50 octets et de 40% pour des textes de 20 octets, lors de tests sur 34 langues. Deuxièmement, cette approche repose sur l'hypothèse du découpage de texte en mots, or, pour certaines langues, comme le chinois, il est difficile de faire la segmentation de texte en mots ; dans le cas de séquences génétiques ADN, on ne peut pas non plus utiliser la notion de mots.

7.3.2. Méthodes basées sur les *n*-grammes

Nous avons présenté la notion de n-grammes et les avantages de l'utilisation de n-grammes dans la section 2.2.4 page 26. Rappelons que le « profil n-grammes » d'un document est la liste des contiguïtés de n caractères les plus fréquents, classées par ordre décroissant de leurs fréquences d'apparition dans le document, ainsi que ces fréquences ; un document est ainsi caractérisé par son profil n-grammes [Cavnar and Trenkle, 1994].

Les systèmes d'identification de langues basés sur les n-grammes suivent en général le même schéma :

- Dans la phase d'acquisition automatique des connaissances linguistiques, on choisit un corpus représentatif pour chaque langue puis on génère un profil caractéristique qui fera office de référence, c.-à-d. calculer les nombres d'occurrences des différents n-grammes (souvent n est compris entre 1 et 5) et les transformer en fréquences.
- Dans la phase de diagnostic, pour chaque texte à identifier, on construit son profil n-grammes et on cherche le profil de référence le plus similaire. Plusieurs méthodes sont proposées pour mesurer cette similarité.

Méthodes de plus proche voisin

Plusieurs algorithmes de catégorisation de texte sont basés sur une distance ou, plus généralement, sur une similarité ou une dis-similarité. L'idée générale est de chercher le texte, parmi l'ensemble d'apprentissage, qui soit le plus proche du texte à classer pour ensuite attribuer la classe de ce texte au texte dont la classe est inconnue. La difficulté est de définir une distance. En pratique, on utilise des pseudo-distances ; voir la section 5.4 page 65, pour une description de cette technique.

Distance de Beesley La méthode de [Beesley, 1988] est basée sur des modèles mathématiques linguistiques qui, à l'origine, étaient utilisés pour décoder les textes cryptés. L'identification comporte deux phases :

- La phase d'acquisition des connaissances : établir, pour chaque langue, un profil bi-grammes (dans l'ordre sans pourtant utiliser une lettre plus d'une fois - il ne s'agit pas du profil au sens usuel) qui fera office de référence, puis calculer les probabilités d'apparitions de chaque bi-gramme dans chaque langue.
- La phase de diagnostic : rechercher, pour le texte à identifier, le profil de référence le plus proche en utilisant une mesure de similarité. Cette phase est composée de trois étapes :
  - Segmenter le texte à identifier en mots.
  - Pour chaque mot, calculer tous les bi-grammes, puis, *pour chaque langue candidate*, attribuer à chaque bi-gramme sa probabilité dans le profil de référence de cette langue et faire le produit de toutes ces probabilités.
  - La langue du texte est celle pour laquelle le produit est maximum.

Exemple : Pour identifier la phrase « LES ORDINATEURS SONT APPELÉS A JOUER UN RÔLE », [Beesley, 1988] segmente cette phrase en mots puis calcule les bi-grammes de chaque mot (en blocs consécutifs, chaque lettre est utilisée une fois seulement) ; si un bi-gramme apparaît dans le profil de référence d'une langue alors sa probabilité sera celle observée dans le profil référence. Le tableau 7.3 montre les scores obtenus ainsi que la décision prise pour le mot « ORDINATEUR ».

Langue	**_O**	**RD**	**IN**	**AT**	**EU**	**RS**	**Décision**
Anglais	1.8905	0.1492	1.9568	1.0945	0	0.1492	éliminé
Espagnol	0.1860	0.1691	0.7102	0.2705	0	0.0338	éliminé
Français	0.4437	0.0657	0.9202	0.8052	0.4601	0.2136	retenu

TAB. 7.3.: Probabilités (multipliées par 100 pour faciliter la lecture) et la décision finale pour le mot « ordinateur »

Par conséquent, il suffit qu'un seul bi-gramme d'un mot soit absent d'un profil de langue pour éliminer l'appartenance du mot à cette langue. Le tableau 7.4 montre les décisions finales.

Cette méthode suppose la segmentation du texte en mots ce qui empêche, par conséquent, son utilisation à des langues dont les frontières entre les mots ne sont pas claires (c.f., page 106). Elle se limite également à prendre en compte uniquement les bi-grammes alors qu'il est important de travailler aussi avec des tri-grammes ou quadri-grammes pour mieux conserver la spécificité de chaque langue : par exemple, la séquence « tion » est typique du français et de l'anglais, si on la découpe en « ti » et « on » alors le système aura du mal à pénaliser les autres langues comme l'espagnol et le portugais qui utilisent « ti » et « on » mais pas « tion ». Autre remarque : la génération des profils ne correspond pas à ce qu'on entend par profil (voir la page 106). Finalement, on remarque que cette méthode est très sensible aux fautes d'orthographes et aux déformations causées pendant les OCR, il suffit qu'un mot soit mal saisi pour pénaliser la langue concernée.

Langue	**Retenu**	$\simeq$ **Retenu**	$\simeq$ **Éliminé**	**Éliminé**
Anglais	1	6	0	1
Espagnol	0	1	4	3
Français	7	1	0	0

TAB. 7.4.: Décisions finales concernant les mots du texte

Distance de Cavnar et Trenkle (CT) Ces deux auteurs [Cavnar and Trenkle, 1994] proposent une méthode d'identification de la langue en deux phases :

Phase d'acquisition : Établir pour chaque langue un profil tri-grammes caractéristique acquis automatiquement qui fera office de référence.

Phase de diagnostic : Rechercher, pour chaque texte à classer, le profil de référence le plus proche en utilisant une mesure de similarité. Cette étape se fait en trois étapes :

- Le profil tri-gramme du nouveau texte est calculé, comme s'il s'agissait d'un corpus de référence ;
- Pour chaque langue, une distance est calculée entre son profil caractéristique et celui du nouveau texte. Cette distance repose sur la définition d'une mesure de similarité : elle correspond à la somme des écarts de rangs entre chaque tri-gramme du nouveau profil et ce même tri-gramme dans le profil de référence, s'il est présent (un écart maximal est attribué si le tri-gramme est absent du profil de référence) ;
- La langue diagnostiquée est celle pour laquelle la distance est la plus petite.

Formellement, la distance entre deux profils P_1 et P_2 est définie comme

$$CT(P_1, P_2) = \sum_{w \in P_1,\, R_{P_1}(w) < NMAX} min(|R_{P_2}(w) - R_{p_1}(w)|, DMAX)$$

où $|x|$ est la valeur absolue de x ; et où $R_P(w)$ (avec w un n-gramme et P un profil de langue) est le rang de w dans le profil P, si w fait parti de P, et $DMAX$ dans le cas contraire (exemples usuels : $NMAX = 500$ et $DMAX = 1000$).

Distance de Kullbach-Leibler(KL) [Sibun and Reynar, 1996] proposent une méthode qui utilise l'entropie relative de Kullbach et Leibler comme mesure de distance. Une entropie relative entre deux distributions de probabilité reflète la quantité d'informations supplémentaires nécessaire pour coder la deuxième distribution en utilisant un code optimal généré pour la première. Techniquement, cette mesure n'est pas une distance car elle n'est pas symétrique.

Formellement,

$$KL(T_1, T_2) = \sum_{N_g} f_2(N_g). \log(\frac{f_2(N_g)}{f_1(N_g)})$$

Ici, la somme est prise sur tous les n-grammes, avec T_1 et T_2 des textes, $f_i(N_g)$ est la fréquence du n-gramme N_g dans le texte T_i. Si le n-gramme N_g est absent dans T_i, une demi-fréquence est alors ajouté pour éviter que le score tombe vers moins l'infini.

Distance du χ^2 Cette distance est présentée dans [Benzecri, 1973] et rappelée dans notre domaine par [Rajman and Lebart, 1998]. Ici p_{ig} est la fréquence du terme (n-gramme) g dans le texte i, et $p_{.g}$ la fréquence du terme g dans l'ensemble de tout le corpus.

$$\chi^2(T_1, T_2) = \sum_{N_g} \frac{1}{p_{.g}} \left(\frac{p_{1g}}{p_{1.}} - \frac{p_{2g}}{p_{2.}} \right)^2 \tag{7.1}$$

Cette distance possède la propriété d'équivalence distributionnelle soulignée par [Benzecri, 1973] : si deux textes T_1 et T_2 ont le même profil n-gramme, on peut les fondre en un seul texte sans que la distance du χ^2 soit modifiée, ni entre les textes, ni entres les descripteurs g.

7.4. Expériences pour la reconnaissance de langue

Dans les expérimentations que nous allons présenter dans cette section, nous nous concentrons sur les nouvelles utilisations de deux algorithmes : les RBFs, pour leurs bonnes performances signalées en section 5.8.3 page 80, et les 1-*plus proches voisins*, pour leur simplicité d'utilisation et leur fréquente utilisation dans le domaine du texte. Tout le travail a été réalisée avec des implémentations Java utilisant le package Jama (voir `http://math.nist.gov/javanumerics`).

La tâche consiste à reconnaître la langue d'un texte. Nous travaillons sur 5 langues : le français, l'arabe, l'anglais, l'espagnol et l'allemand[1]. La tâche étant facile, nous la compliquons en utilisons des chaînes très courtes pour tester les algorithmes.

[1] L'utilisation de nos algorithmes pour l'identification d'autres langues est très facile à mettre en œuvre. En effet, il nous suffit de quelques textes pour chaque langue afin de lui calculer et associer un profil représentatif. Cette opération est quasi immédiat (moins d'une seconde).

Avec des ensembles-tests composés d'échantillon de longueur 100, 50 ou 20 bytes ; la table 7.5 donne les taux de succès (TS) selon la longueur des échantillons à classer. Le taux de succès est évalué par test sur un ensemble disjoint de l'ensemble d'apprentissage (ou *training set*).

Notons que le taux de succès coïncide avec les micro- et macro-moyen pour le rappel et précision puisque le problème consiste à classer de textes appartenant chacun à une et une seule langue (voir la section 9.3.5 page 126).

Algorithme	TS (100 bytes)	TS (50 bytes)	TS (20 bytes)
1-NN (KL)	100 %	99.4 %	92.8 %
1-NN (χ^2)	98.8 %	96.6 %	87.92 %
RBF ($\sigma^2 = 10$)	37.6 % (100 %)		
RBF ($\sigma^2 = 100$)	98.8 %	93 %	71.04 %

Tab. 7.5.: Résultats obtenus en reconnaissance des langues

Les algorithmes avec des caractéristiques entre parenthèse, exemple : RBF (2-regroupés prof.), correspondent à des résultats obtenus en coupant l'ensemble d'apprentissage en 50 sous-ensembles au lieu de déterminer les profils sur l'ensemble du training set. Ceci améliore parfois l'apprentissage RBF mais n'a apporté aucune amélioration dans certains cas d'échantillons très courts. Nous travaillons maintenant sur 250 échantillons de 100 bytes comme ensemble d'apprentissage, pour étudier l'influence de ce "regroupement" pour les RBFs ou les 1-plus proches voisins. Les résultats sont donnés en table 7.6.

L'hyperparamètre σ^2 pour l'apprentissage RBF était facilement choisi dans notre jeu d'essai (voir section 5.6 page 82), car le taux de succès était stable sur une grande plage de valeurs de σ ; mais dans le cas de très courtes chaînes, l'efficacité variait beaucoup selon σ et en fonction du "regroupement". Ceci amène à deux hyperparamètres difficiles à calculer.

7.5. Conclusion

L'objectif de ce chapitre était d'examiner les approches existantes pour l'identification de la langue. Nous avons présenté les deux familles d'approches : celles basée sur des connaissance linguistique et celle basée sur l'approche purement statistique. Nous avons préféré l'approche statistique basée sur les n-grammes, car elle est indépendante de la langue, tolérante aux fautes d'orthographe et déformations, et capture des connaissances linguistiques tel que les mots les plus fréquents.

Les expériences menées nous montrent la fiabilité de l'approche de n-grammes basée sur la distance du χ^2 puisque le taux de succès est de 100%

Algorithme	Hyperparamètres	TS (100)	TS (50)	TS (20)
RBF	$\sigma^2 = 10$	99.2 %	84.8 %	31.52 %
	$\sigma^2 = 100$	98 %	93.2 %	71 %
RBF (2-regroupés prof.)	$\sigma^2 = 100$	97.2 %	88 %	68.56 %
RBF (5-regroupés prof.)	$\sigma^2 = 1000$	98.8 %	94 %	80.88 %
RBF (10-regroupés prof.)	$\sigma^2 = 1000$	99.2 %	95.2 %	76.72 %
RBF (25-regroupés prof.)	$\sigma^2 = 1000$	98.8 %	94.8 %	82.4 %
RBF (regroupés prof.)	$\sigma^2 = 100$	88.4 %	80.6 %	
	$\sigma^2 = 100000$	87.6%	77.4 %	61.36 %
1-PPV	χ^2	99.2 %	96.6 %	88.4 %
1-PPV	KL			47.2 %
1-PPV (2-regroupés prof.)	χ^2	99.6 %	96.8 %	88.8 %
1-PPV (5-regroupés prof.)	χ^2	100 %	97.6 %	90 %
1-PPV (10-regroupés prof.)	χ^2	99.2 %	97.2 %	88.56 %
1-PPV (10-regroupés prof.)	KL			89.84 %
1-PPV (25-regroupés prof.)	χ^2	100 %	96.8 %	87.2 %
1-PPV (regroupés prof.)	χ^2	100 %	93 %	84.56 %
1-PPV (regroupés prof.)	KL	99.7 %	97.4 %	89.4 %

TAB. 7.6.: Résultats obtenus en reconnaissance des langues, en testant différents degrés de regroupement. "m-regroupés profils" signifie que les échantillons d'apprentissage ont été regroupés m par m ; "regroupés", que tous les textes d'une même classe sont regroupés. Une petite valeur de m préserve la variabilité de l'ensemble d'apprentissage, un m plus grand conduit à des profils mieux définis. La dissimilarité de Kullbach-Leibler semble meilleure que celle du χ^2 pour des petits échantillons en test, mais supporte mal les petits échantillons en apprentissage (comme on s'y attend en examinant le comportement de cette dissimilarité pour de petites fréquences)

lorsque les textes à identifier sont de taille supérieure à 100 caractères (entre 8 et 15 mots). Les expériences ont été menées sur des textes très courts.

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. Les résultats de [Dunning, 1994] utilisant des modèles de Markov, avec deux langues au lieu de 5 ici, suggèrent que les modèles de Markov entraînés avec $50Ko$ de données ont environ les mêmes performances que les 1-plus proches voisins avec $25Ko$. Le taux de succès en cas de réponse aléatoire étant de 20% dans le cas de 5 langues et de 50% dans le cas de 2 langues. En rappelant que les deux langues utilisées par Dunning sont testées sur le même échantillon, nous supposons que les 1-plus proches voisins sont plus adaptés pour ce problème.

Chapitre 8

Traduction automatique

Sommaire

8.1. Introduction

Comme nous utilisons un traducteur automatique dans nos chaînes de traitement, nous plaçons ici un court chapitre faisant le point sur ce sujet difficile : histoire, espoirs passés, solutions actuelles et critères d'évaluation. On verra que nous n'avons pas besoin d'un traducteur parfait, qui n'existera sans doute jamais, car les outils actuellement disponibles nous suffisent pour retrouver les thèmes d'un texte et donc pour catégoriser automatiquement les textes multilingues, de façon assez correcte.

La plupart des grands projets de traduction automatique sont nés entre 1958 et 1966 des besoins de traduction à partir du russe engendrés par la guerre froide. Le système Systran, utilisé par la *Foreign Technology Division* de l'armée de l'air américaine, est l'un des systèmes utilisés pour cette tâche. Ce système utilisait un énorme dictionnaire russe-anglais couvrant un grand nombre de domaines. La sortie de ce système était ensuite examinée par des spécialistes qui étaient chargés d'isoler les textes ayant une valeur stratégique ou scientifique pour les passer ensuite à des traducteurs humains.

En 1966, la recherche en traduction automatique a connu un coup d'arrêt avec la sortie du rapport *ALPAC de la National Science Foundation* qui concluait à l'impossibilité d'une traduction automatique de qualité. La plupart des chercheurs américains se sont alors tournés vers une science émergente : l'intelligence artificielle [Chandioux, 1998].

8.2. Premières approches historiques de la traduction automatique

[Chandioux, 1998] présente un panorama des approches utilisées pour la traduction automatique. Historiquement, trois approches sont expérimentés :

8.2.1. Décryptage

Les développeurs de la fin des années cinquante concevaient la traduction automatique comme un simple problème de décryptage. A chaque mot du texte incompréhensible correspondait un mot de la langue connue.

Cette approche a montré ses limites de fait que la traduction n'était pas biunivoque, qu'un même mot pouvait avoir plusieurs catégories grammaticales et qu'à chaque catégorie grammaticale pouvaient correspondre plusieurs sens.

8.2.2. Analyse par micro-contexte

Des tests de compréhension ont révélé que si un locuteur humain lisait un texte en déplaçant une feuille de carton avec un trou qui laissait paraître quelques mots contigus, beaucoup d'ambiguïtés pouvaient être levées. Ceci a donné naissance à l'analyse par micro-contexte, qui est le principe de fonctionnement des systèmes de traduction automatique dits de première génération comme le système Systran.

8.2.3. Imiter la traduction humaine

Les systèmes de traduction automatique dites de deuxième génération s'inspirent de la traduction humaine. La traduction humaine se décompose habituellement en trois étapes : la compréhension du message en langue source, la transposition de la teneur du message en langue cible et la formulation du message selon les règles de la langue cible. Parallèlement, un système de traduction automatique de deuxième génération opère au niveau de la phrase et la traite en trois étapes : l'analyse, le transfert et la génération. Les chercheurs du GETA (Groupe d'étude pour la traduction automatique) à Grenoble ont été les premiers à jeter les bases de cette nouvelle génération.

Malheureusement, on n'est pas parvenu à développer un système général de traduction automatique de deuxième génération, même en se limitant aux textes scientifiques et techniques [Chandioux, 1998].

8.3. Nouvelles approches, plus modestes

Les nouvelles approches sont moins ambitieuses et tentent de se limiter à certains types de textes, à certains domaines, ou encore en réduisant l'ambiguïté des textes lors de leur rédaction. Dans les prochains paragraphes, nous présentons les solutions proposées :

8.3.1. Mémoire de traduction

Il s'agit d'une base de données qui stocke les phrases issues de deux langues, source et cible. Tout ce qu'on a traduit est contenu dans deux entités : le texte original dans la langue source et sa traduction antérieure dans la langue cible. Lorsque on travaille sur une nouvelle traduction, le système de mémoire de traduction recherche chaque phrase du texte original dans la base de données et, si la recherche est fructueuse, la phrase est traduite et il ne reste qu'à confirmer la traduction.

De nombreux systèmes commerciaux de mémoire de traduction sont proposés comme *Translation Manager 2* d'IBM, *Translator's Workbench* de Trados ou *Star Transit* qui mémorisent toutes les phrases au fur et à mesure de leur traduction.

Un système à base de mémoire de traduction ne peut réaliser lui même aucune traduction ; c'est un support qui aide le traducteur humain dans son travail. Bien que les systèmes de mémoire de traduction soient réellement utiles sur du texte répétitif comme les contrats d'assurances, les conventions collectives ou les descriptions de tâches, ils sont moins utiles dans des textes généraux. [Volk, 1998] compare les systèmes à base de mémoire de traduction avec les systèmes de traduction automatique.

8.3.2. Sous-langages et langages contrôlés

On appelle sous-langage un sous-ensemble de la langue ayant ses propres règles syntaxiques et un vocabulaire limité dont les relations sémantiques sont cernables. Le traducteur automatique METEO est un exemple connu d'application de la traduction automatique à un sous-langage. Ce système a pour vocation de traduire les prévisions publiques, agricoles et maritimes émises par Environnement Canada [Chandioux, 1998].

D'une façon générale, cette approche part de l'idée simple suivante : si on ne peut pas traduire n'importe quel texte, pourquoi ne pas écrire en fonction de la machine quand on sait que le texte devra être disponible en plusieurs langues.

« *Les premières recherches sur les langages contrôlés ont porté sur l'anglais technique, le but étant d'obtenir des documentations non ambiguës tant pour des utilisateurs anglophones que pour des utilisateurs dont l'anglais n'est pas la langue maternelle. Un bon exemple est le "Simplified English" développé par l'AECMA (Association européenne de constructeurs de matériel aéronautique) et dont l'usage est recommandé par l'ATA (American Transport Association). Le GIFAS (Groupement des Industries Françaises Aéronautiques et Spatiales) mène depuis quelques années des travaux similaires sur le « français rationalisé »* » [Chandioux, 1998].

8.4. Évaluer la traduction automatique

Le problème de l'évaluation de la traduction est difficile. Il y a beaucoup de littérature sur cette question [Hovy et al., 2002]. Le gouvernement américain finance depuis quelque temps des recherches sur les évaluations comparatives, du genre des TREC (*The Text REtrieval Conference*, voir `http://trec.nist.gov/`).

Dans les années antérieures, on a expérimenté des méthodes d'évaluation complexes et coûteuses qui font appel à des tests objectifs de compréhension ainsi qu'à des évaluations subjectives de la fluidité du texte traduit. On compare plusieurs traductions humaines et machines sur ces plans. Ces évaluation tentent de juger si un système de traduction automatique est fiable ou non. Pour ce faire elles s'intéressent à la qualité des traductions que le système produit, sa rapidité d'exécution, son coût, le temps qu'il exige pour la post-édition, sa convivialité, ou encore son potentiel de développement. D'autres questions se posent : que faut-il utiliser pour tester les systèmes : des corpus réels ou des corpus fabriqués ? Serait-il utile d'élaborer une méthodologie générale d'évaluation ? Voir [Cormier, 1992, King, 1992] pour une étude complète sur ce sujet. Notons que, vu le nombre limité de systèmes, ces évaluations ne concernaient qu'un seul système à la fois.

Récemment, suite à la disponibilité de plusieurs systèmes de traduction, une étude comparative de ces système est devenue possible. A titre d'exemple, [Volk, 1997] propose une évaluation de plusieurs systèmes commerciaux de traduction automatique pour les langues anglais-allemand. Il évalue la performance de six systèmes à savoir :

1. German Assistant in Accent Duo V2.0 (développer par Micro-Tac/Globalink ; distributeur : Accent)
2. Langenscheidts T1 Standard V3.0 (développer par GMS ; distributeur : Langenscheidt)
3. Personal Translator plus V2.0 (développer par IBM ; distributeur : von Rheinbaben & Busch)
4. Power Translator Professional (développeur et distributeur : Globalink)
5. Systran Professional for Windows (développer par Systran S.A. ; distributeur : Mysoft)
6. Telegraph V1.0 (développeur et distributeur : Globalink)

Cette évaluation est basée sur des corpus artificiels (en anglais *test suites*), qui regroupent des phrases dont la construction syntaxique est représentative des textes d'un client ; ils sont très utiles au développeur qui peut, grâce à eux, vérifier la force réelle de son système.

Une tendance assez forte à l'heure actuelle consiste à utiliser des mesures extrêmement simples qui semblent avoir une très bonne corrélation avec les méthodes coûteuses. [Papineni et al., 2002], par exemple, présente une technique simple et rapide pour évaluer les performances de traducteurs automatiques. Cette technique est basé sur l'idée simple que si l'on souhaite évaluer la performances d'un système de traduction (ST), on doit comparer les résultats de traduction du système avec ceux produits par un traducteur humain (TH). Plus le texte du ST est proche de celui TH plus la traduction est bonne. La proximité est mesurée par une métrique. [Papineni et al., 2002] propose une mesure appelé BLEU (pour *<u>Bi</u><u>L</u>ingual <u>E</u>valuation <u>U</u>nderstudy*) qui consiste à calculer le nombre de mots communs entre les phrases fournit par le ST et le TH. Leurs résultats semblent consistants et les jugements donnés par leur système correspondent à ceux proposés par les THs. Ceci est très intéressant car il permet aux développeurs de systèmes de traduction d'évaluer leurs systèmes et tester leurs hypothèses rapidement avec les moindres coûts possibles.

8.5. Conclusion

Les traducteurs automatiques présentent une fiabilité toute relative. Aucun n'offre de résultats parfaits, même si plusieurs procédés différents sont mis en oeuvre. Les traductions sont approximatives.

Ce chapitre a pour objectif de montrer les approches successives proposées pour la traduction automatique. Nous avons montré que l'objectif initial était trop ambitieux : dans les années 60, on rêvait d'une machine rendant des textes parfaitement traduits. De nouvelles approches, qui tentent de se limiter à certaines taches et à certains types de textes, sont alors proposées. Un autre objectif de ce chapitre était de montrer comment les traducteurs automatiques sont évalués. En fin, nous terminons ce chapitre par une phrase qui résume, à notre sens, les tendances actuelles pour la traduction automatique :

« *Si l'ordinateur ne comprend pas ce que nous écrivons, écrivons dans un langage qu'il comprend et laissons la littérature aux humains ...* » [Chandioux, 1998].

Chapitre 9

Cadre pour la catégorisation de textes multilingues

Sommaire

9.1. Introduction

Dans ce chapitre, nous proposons des solutions pour étendre la catégorisation de textes aux corpus multilingues. Ceci introduit des contraintes supplémentaires dans la catégorisation de textes : il faut reconnaître automatiquement la langue d'un texte puis procéder à une traduction automatique. Notre approche semble naïve, mais elle a le mérite d'être 1) la première solution automatique proposée, à notre connaissance, 2) opérationnelle, comme le montre les premières expérimentations sur un corpus d'articles de journaux allemands, anglais et français.

La phase d'apprentissage s'effectuera, comme en monolingue, à partir d'un corpus d'apprentissage étiqueté rédigé dans une langue donnée $\mathfrak{L}_{app}$. Mais l'inférence sera possible pour un texte rédigé dans une langue quelconque, dès qu'un traducteur automatique sera disponible de cette langue vers $\mathfrak{L}_{app}$. Notons que notre approche exclut les méthodes qui utilisent de manière explicite des informations spécifiques à chaque langue.

Bien entendu, à l'instar de la catégorisation de textes monolingue, le texte à classer doit appartenir au même domaine que les textes utilisés lors de l'apprentissage. On ne saurait, par exemple, essayer de classer un article scientifique à partir d'un modèle construit sur un ensemble d'apprentissage constitués d'articles de journaux à scandale.

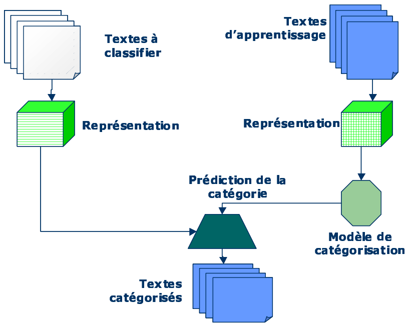
Ce chapitre est organisé de la manière suivante : dans la section 9.2, nous exposerons l'approche que nous avons pour étendre la catégorisation de textes au cas multilingue. Dans la section 9.3, nous mettrons en oeuvre le cadre proposé sur un exemple réel de catégorisation de journaux. Dans la section 9.4, nous discutons des résultats obtenus, et nous essayerons de mettre en perspective notre démarche afin de la faire évoluer. Nous conclurons alors dans la section 9.5.

9.2. Méthodes pour la catégorisation de textes multilingues

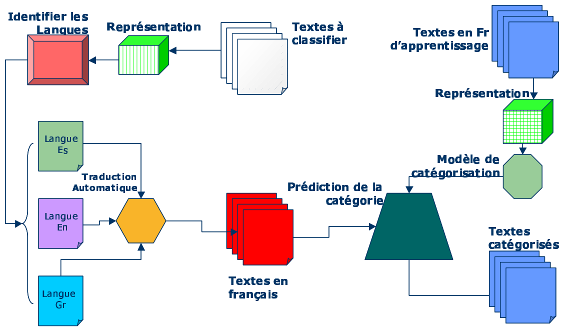
9.2.1. Nouveau cadre pour la catégorisation multilingue

Dans le cadre de la catégorisation de textes multilingues, le processus comporte deux nouvelles exigences : le corpus de textes étiquetés utilisé pour l'apprentissage est disponible dans une langue $\mathfrak{L}_{app}$ donnée ; chaque nouveau texte à classer est dans une langue que l'on doit d'abord déterminer avant de pouvoir lui associer son étiquette.

Pour répondre à ces nouvelles contraintes, nous aménageons le processus de catégorisation exposé dans la section 1.3 page 9. La phase d'apprentissage n'est pas modifiée, en revanche la phase de classement comporte deux étapes



(a) Catégorisation de textes monolingue



(b) Catégorisation de textes multilingue

Figure 9.1.: Schémas généraux de catégorisations mono et multilingue

supplémentaires (figure 9.1.b) :

1. nous devons tout d'abord détecter la langue dans laquelle le texte d_j à classer est rédigé ;
2. si la langue est reconnue par le traducteur, il est traduit vers la langue $\mathfrak{L}_{app}$ et le texte à classer devient d_{app}.

Nous recherchons alors les occurrences des termes $(w_{1j}, w_{2j}, ..., w_{|\mathcal{T}|j})$ dans d_{app} afin de pouvoir appliquer le modèle prédictif et ainsi associer une catégorie c_i au texte à classer.

9.2.2. Détection de la langue du texte à classer

Il est important de détecter avec précision la langue dans laquelle le texte à classer est rédigé, car une erreur à ce niveau voue à l'échec les étapes suivantes.

Il existe deux familles d'approches dans l'identification de la langue : linguistique ou statistique. Afin de rester cohérent avec nos choix pour la catégorisation de textes, nous avons privilégié l'approche statistique. Cette approche capture automatiquement certaines régularités statistiques des langues. Comme pour la catégorisation de textes, nous avons utilisé les 3-grammes qui sont des séquences de 3 caractères consécutifs extraits du texte à classer. Nos précédents travaux ont montré qu'un texte de longueur de 100 octets permettait d'obtenir une reconnaissance d'excellente qualité, à près de 99% [Teytaud and Jalam, 2001]. Et pour des textes plus longs, la reconnaissance est parfaite.

9.2.3. Traduction du texte à classer

La traduction du texte à classer dans la langue du corpus d'apprentissage $\mathfrak{L}_{app}$ est également une étape primordiale. L'objectif ici n'est pas de produire un texte traduit retraçant fidèlement les propriétés sémantiques de l'original, mais de fournir un texte assurant une qualité de classement suffisante. Il est évident que le résultat obtenu dépendra du traducteur utilisé, une des perspectives immédiates de notre travail actuel consistera à analyser de manière approfondie le traducteur pour évaluer son efficacité lors de la catégorisation de textes.

Afin de défricher le terrain, nous avons utilisé un traducteur en ligne disponible sur Internet (`http://babelfish.altavista.com/`, qui utilise la technologie Systran). Ce traducteur n'est sans doute pas le meilleur, mais comme il est publiquement disponible, nous aurons la possibilité par la suite de comparer nos résultats avec d'autres études. L'utilisation d'un meilleur traducteur ne peut qu'améliorer la catégorisation des textes.

9.3. Application sur les corpus CLEF

9.3.1. Constitution du corpus

La constitution de ce corpus a été un travail long et difficile ; malgré nos recherches, nous n'avons pas trouvé de corpus multilingues de textes étiquetés en classes comparables. Nous avons dû adapter pour cela les documents proposés par les organisateurs du concours CLEF (voir la page web : `http://clef.iei.pi.cnr.it`).

Les corpus de documents utilisés dans la campagne d'évaluation CLEF proviennent de différents journaux tels que le Los Angeles Times (États-Unis), Le Monde (France), La Stampa (Italie), Der Spiegel et Frankfurter Rundschau (Allemagne), d'agences de presse comme EFE (Espagne) ou des dépêches de l'agence télégraphique suisse (disponibles en allemand, français et italien). Les documents de ces corpus sont tous extraits de l'année 1994 et les thèmes abordés sont approximativement comparables.

Ces corpus sont destinés aux tâches de la recherche documentaire (RI) monolingues, bilingues et multilingues. Les fichiers de ces corpus sont au format SGML, par exemple, le fichier `lemonde_19940604.sgml` concerne les articles apparus dans le journal Le Monde (LM) lors de la journée du 4 juin 1994. Chaque article de ce fichier contient des balises sgml qui décrivent son contenu (table 9.1). Il est facile d'y distinguer son numéro d'identification, son titre, et son corps.

La taille des corpus varie fortement entre les langues, avec des volumes plus restreints pour le français et l'italien. Le nombre de mots par article reste assez similaire (environ 130), avec une moyenne un peu plus élevée pour la collection anglaise (167). Par contre, la variabilité de cette longueur demeure assez forte

```
<DOC>
<DOCNO>LEMONDE94-000386-19940604</DOCNO>
<DOCID>LEMONDE94-000386-19940604</DOCID>
<ACCOUNT>339369</ACCOUNT>
<GENRE>RECTIF</GENRE>
<DATE>19940604</DATE>
<LMDOC>MHB</LMDOC>
<FAB>06031011</FAB>
<SUBJECTS>DEMENTI,DESSIN</SUBJECTS>
<GO21>PUBLICATION</GO21>
<NAMES>SERGUEI</NAMES>
<PUM1>QUO</PUM1>
<REFERENCE1>2-002-06</REFERENCE1>
<SEC1>IDE</SEC1>
<PAGE>13</PAGE>
<TITLE>Serguei précise</TITLE>
<TIO1>PAS DE PANIQUE A BORD</TIO1>
<TEXT>Un lecteur s'est étonné de constater qu'une publication satirique d'extrême droite, Pas de
panique à bord, citait, parmi les noms de ses collaborateurs, celui du dessinateur Sergueï. Etonnement
encore plus grand _ pour ne pas dire plus _ de Sergueï, celui que nos lecteurs connaissent bien et
qui ne saurait être le même que son homonyme de Pas de panique à bord, s'il existe. &gt ;</TEXT>
</DOC>
```

Tab. 9.1.: Exemple extrait de la collection CLEF et qui montre un article du journal Le Monde publié le 4 juin 1994

(écart-type d'environ 120), sauf pour les langues espagnole, où l'écart-type est de 60, et italienne, où l'écart-type est de 97 [Savoy, 2002].

Comme on le sait, la catégorisation de textes nécessite d'avoir un échantillon d'apprentissage composé de textes associés à leurs étiquettes (ou classes). Ces exemples serviront, dans le processus d'apprentissage, à apprendre un classifieur (ou modèle) qui sera ensuite appliqué sur les textes à catégoriser. Une première approche peut consister à utiliser les termes-index associés à chaque article comme classes d'appartenance. En général, l'indexation est une opération qui *décrit* et *caractérise* un document, ou un fragment de document, en repérant les thèmes présents dans ce document [Afnor, 1993]. Malheureusement, les résultats de nos expérimentations utilisant les termes-index, sur les corpus français et allemand, sont médiocres : le taux d'erreur est proche de celui du hasard.

Il est difficile d'apprendre si l'on utilise les termes-index proposés en tant que classes pour différentes raisons :

- Les termes-index associés aux articles français et allemand sont trop nombreux, plus de 16200 termes (et donc classes) pour le corpus français et plus de 32000 termes pour le corpus SDA allemand. Il n'y a pas eu de règle d'indexation basée sur des vocabulaires contrôlés et beaucoup de termes sont des synonymes (mort, morts ; France, fr ; German, gr).
- Les termes-index proposés ne représentent vraiment pas les thèmes abor-

Classes	CLEF id	CLEF topic	#LAT	#LM	#SDA
C_1	92	U.N. sanctions against Iraq	27	24	23
C_2	95	Conflict in Palestine	96	89	66
C_3	103	Conflict of Interests in Italy	10	24	70
C_4	108	Southern Yemen Secession	18	19	63
C_5	119	Destruction of Ukrainian nuclear weapons	54	33	55
C_6	122	North American car industry	27	23	6
C_7	124	Common foreign and security policy (CFSP)	32	48	24
C_8	131	Intellectual Property Rights	40	43	43
C_9	133	German Armed Forces Out-of-area	10	21	17
C_{10}	140	Mobile phones	70	95	23
Total			**384**	**419**	**390**

TAB. 9.2.: Description des catégories utilisées

dés dans les articles ; par exemple, on associe à la dépêche de la table 9.1 le mot clé dessin alors qu'elle parle d'un démenti.
- Les termes-index dans les dépêches de l'agence télégraphique suisse (allemand et français) ne sont pas séparés par des signes de ponctuation ainsi, il est difficile d'extraire les termes composés comme "*conseil de sécurité*". Le corpus de Los Angeles Times, à la différence des autres corpus, ne fournit pas de tout de termes-index, ces termes-index aident normalement à décrire le contenu d'un article.

Dans nos expérimentations, nous choisissons donc de considérer les thèmes proposés dans la campagne CLEF 2002 comme classes à prédire. Ces dernières sont très variées ; on trouve par exemple : "U.N. sanctions against Iraq", "Conflict in Palestine" ou "Leaning Tower of Pisa". Nous avons travaillé sur trois corpus de langues anglaise, française et allemande : le Los Angeles Times (LAT), Le Monde (LM) et l'agence télégraphique suisse (SDA). Les thèmes utilisés dans nos évaluations sont décrits dans la table 9.2.

9.3.2. Représentation des textes

La représentation des textes est une étape critique. Nous avons choisi d'utiliser les mots et les 3-, 4- et 5-grammes. Nous avons appliqué notre algorithme de sélection de termes χ^2_{multi} présentées dans la section 3 page 35 [Clech et al., 2003] en sélectionnant 100 mots avec pour seuls prétraitements l'uniformisation de la casse et la suppression des StopWords. En outre, nous avons sélectionnés 200 3-, 4- et 5-grammes avec pour seul prétraitement l'uniformisation de la casse. Nous avons choisi de sélectionner moins de mots que de n-grammes puisqu'un mot est composé de plusieurs n-grammes ; après plusieurs expériences, le choix de 100 mots et 200 n-grammes apparaît comme un bon compromis conservant la structure informationnelle du corpus.

L'étude des termes sélectionnés révèle la présence importante de noms propres, tels les noms des pays ou de leurs ressortissants, ou encore les noms de personnalités. L'étude indique également certaines difficultés de traduction. Par exemple, l'expression française "téléphone portable" est traduite en anglais par "portable phone" ce qui n'a pas le sens voulu. Les noms propres ne sont pas non plus épargnés par des difficultés de traduction ; ainsi le terme français "Koweït" est laissé tel quel au lieu d'être traduit par le terme "Kuwait".

9.3.3. Algorithmes d'apprentissage

Dans notre application, nous utilisons deux méthodes renvoyant à des paradigmes d'apprentissage très différents (voir les sections 5.3 page 58 et 5.4 page 65) :

- une méthode arborescente : C4.5 [Quinlan, 1993]. Cette algorithme glouton a la particularité de sélectionner les variables et effectue des découpages parallèles aux axes ;
- une méthode d'estimation des probabilités locales : les 3 plus proches voisins (3-ppv) [Aha et al., 1991]. Cet algorithme est sensible aux variables non pertinentes ainsi qu'aux espaces de représentations creux [Mitchell, 1997].

9.3.4. Reconnaissance de la langue

Dans nos expérimentation, nous avons utilisé la distance de χ^2 pour identifier la langue de texte parmi les trois langues français, anglais et allemand. Nos nouveaux tests confirment nos résultats précédents [Teytaud and Jalam, 2001] : pour des textes de taille égale ou supérieure à 100 caractères le taux de reconnaissance de la langue est de 100%.

9.3.5. Catégorisation des articles

Afin de pouvoir évaluer notre processus de catégorisation de textes dans un corpus multilingue, nous avons effectué plusieurs expériences de catégorisation. Pour chacune d'elles, nous avons évalué le taux d'erreur pour toutes les configurations de représentations des textes (100 mots, 200 3-grammes, 4-grammes et 5-grammes) et des modèles utilisés (C4.5 et les 3 plus proches voisins).

Usuellement, en catégorisation de textes, un document peut appartenir à n classes. Cependant, dans l'application présentée ici (voir la table 9.2), chaque document n'appartient qu'à une classe, et une seule. Il en résulte que les rappel et précision "microaveraging" sont identiques et égaux au taux de succès ;

	10-V.C. LAT		10-V.C. LM		10-V.C. SDA	
taux d'erreur	**C4.5**	**3-NN**	**C4.5**	**3-NN**	**C4.5**	**3-NN**
100 mots	16%	8%	24%	5%	15%	3%
200 3-grammes	23%	9%	20%	9%	11%	2%
200 4-grammes	16%	7%	16%	5%	8%	2%
200 5-grammes	14%	7%	16%	5%	9%	2%

TAB. 9.3.: Taux d'erreur, en validation croisée (V.C.), pour les articles de journaux dans leur langue originelle

	LM (An)		SDA (An)	
Taux d'erreur	**C4-5**	**3-NN**	**C4-5**	**3-NN**
100 mots	25%	6%	12%	3%
200 3-grammes	16%	6%	9%	3%
200 4-grammes	12%	6%	9%	2%
200 5-grammes	13%	5%	12%	3%

TAB. 9.4.: Taux d'erreur, en validation croisée, pour les articles de journaux traduits en anglais. LM (An) désigne le corpus français Le Monde (LM) traduit en anglais (An). SDA (An) désigne le corpus allemand de l'agence télégraphique suisse (SDA) traduit en anglais (An)

poue cette raison nous allons aussi utiliser le taux d'erreur pour mesurer les performances.

Nous présentons tout d'abord les résultats en catégorisation monolingue (table 9.3), ils ont servi de référence pour juger de la qualité de ceux obtenus dans un contexte multilingue.

Catégorisation monolingue

Afin d'évaluer le niveau intrinsèque de difficulté de catégorisation des articles, nous avons mesuré l'erreur de classement pour nos 3 corpus (LAT, LM et SDA) dans leur langue d'origine.

Les résultats (par *10-validation croisée* [Stone, 1974]) sont présentés dans la table 9.3) ; nous en dégageons 4 principaux résultats :

- Il y a un apprentissage effectif puisque le taux d'erreur est largement inférieur aux taux d'erreur du classifieur par défaut ;
- Il y a un avantage important pour la méthode du 3-ppv (un écart supérieur à 10 points) par rapport à C4.5 qui souffre de la fragmentation des données ;
- Le bon apprentissage du 3-ppv laisse penser que les termes sélectionnés

sont pertinents ;
– Le corpus allemand est plus facile à catégoriser que les deux autres.

Catégorisation multilingue

Comme décrite précédemment, la catégorisation multilingue nécessite des étapes intermédiaires complémentaires, chacune pouvant générer du biais pour le processus même de la catégorisation. En effet, si l'étape de traduction est de qualité médiocre, l'apprentissage sera difficile et les résultats de la catégorisation seront de mauvaise qualité. Par ailleurs, comme nous avons dit dans la section 9.3.2, les noms propres sont très présents dans les mots sélectionnés, et de ce fait nous pourrions nous demander si de bons résultats de catégorisation ne seraient pas simplement dus à ces noms propres. Ainsi, dans l'objectif de valider notre processus de catégorisation multilingue, nous évaluons d'abord l'effet de la traduction, puis l'effet potentiel des noms propres et enfin la capacité même du modèle à être appliqué sur des textes traduits.

Les effets du traducteur Pour mesurer l'effet du traducteur sur le contenu informationnel des documents, nous avons traduit vers l'anglais le corpus français LM et le corpus allemand SDA. Nous avons appliqué le schéma d'apprentissage monolingue (voir la figure 9.1.a) et évalué l'erreur en validation croisée (table 9.4). Les résultats montrent que le contenu informationnel des corpus, du moins ce qui est nécessaire à la catégorisation, est très peu dégradé par la traduction. En effet, les différences des taux d'erreurs obtenus après traduction (table 9.4) comparées à ceux obtenus dans la langue d'origine (table 9.3) ne sont pas significatives.

Les effets de noms propres Pour évaluer comment les noms propres peuvent influencer les résultats de l'étape d'apprentissage, nous avons estimé un modèle à partir du corpus LAT dans sa langue d'origine (anglais) que nous appliquons directement sur les corpus LM et SDA laissés dans leur langue d'origine (respectivement français et allemand). Nous supposions que, si les résultats ne sont pas significativement différents de ceux obtenus après traduction, alors, la contribution des noms propres lors de l'étape de catégorisation est supérieure à la contribution des mots communs traduits.

Enfin, nous avons étudié les résultats de catégorisation après traduction en anglais des corpus à classer (LM et SDA) en appliquant le modèle élaboré à partir du corpus du LAT en anglais.

Nous présentons seulement les résultats concernant la représentation utilisant 100 mots et celle utilisant 200 4-grammes. Le tableau 9.5 regroupe les résultats obtenus pour le corpus LM et le tableau 9.6 ceux de SDA. Nous en dégageons 3 résultats principaux :

(a) représentation avec 100 mots

	Appris et appliqué sur LM (Fr)		**Appris sur LAT (An) et appliqué sur**			
	LM (Fr)		**LM (Fr)**		**LM (An)**	
	3-NN	**C4.5**	**3-NN**	**C4.5**	**3-NN**	**C4.5**
$\rho^\mu = \pi^\mu$	95%	76%	78%	12%	89%	60%
ρ^M	94%	68%	78%	14%	92%	55%
π^M	92%	71%	80%	10%	88%	52%

(b) représentation avec 200 4-grammes

	Appris et appliqué sur LM (Fr)		**Appris sur LAT (An) et appliqué sur**			
	LM (Fr)		**LM (Fr)**		**LM (An)**	
	3-NN	**C4.5**	**3-NN**	**C4.5**	**3-NN**	**C4.5**
$\rho^\mu = \pi^\mu$	95%	84%	86%	55%	90%	68%
ρ^M	96%	80%	81%	43%	89%	59%
π^M	95%	79%	90%	32%	91%	50%

Tab. 9.5.: Précision et Rappel (micro et macro-moyen), pour le corpus Le Monde (LM) représenté par 100 mots (table a) et pour 200 4-grammes (table b). Les deux tables montrent les performances obtenues en validation croisée. On applique le modèle LM (Fr) sur des textes écrits en français et les résultats obtenus sont comparés avec ceux du modèle LAT anglais sur le corpus “LM écrit en français” et sur le corpus “LM traduit en anglais”

– Le premier concerne la viabilité de notre approche : même si le taux d'erreur s'accroît quand on passe d'un apprentissage sur les traductions anglaise au lieu des textes originaux, la qualité de prédiction surpasse largement celle du classifieur par défaut.
– Le second concerne la faible variabilité des résultats obtenus en fonction des représentations de texte utilisées (mots ou n-grammes) : on ne peut pas dire si l'une est plus robuste que l'autre.
– La troisième concerne le faible biais introduit par les noms propres ; les tableaux 9.5 et 9.6, montrent que l'écart entre les résultats avant et après traduction sont suffisamment significatifs : nous observons pour le modèle 3-PPV un saut de plus de 10 points pour le corpus LM (Tableau 9.5) et un saut de plus de 70 points pour le corpus SDA (Tableau 9.6) ; rappelons que nous supposions que, si les résultats du modèle estimé pour l'anglais mais appliqué à des textes en français ou allemand, n'étaient pas significativement inférieurs de ceux obtenus sur ces textes traduits, alors, la contribution des noms propres lors de l'étape de catégorisation serait supérieure à la contribution des mots communs traduits.

Par ailleurs : le taux d'erreur de C4.5 s'explique largement par la difficulté de prédire les classes c_3, c_6 et c_9 dont les taux de rappels et de précisions sont nuls (voir les tables 9.7 et 9.8). Pour c_3 et c_9 cela est dû au seuil d'élagage (fixé à 10) correspondant au nombre de textes du corpus d'apprentissage (LAT) composant ces classes (voir la table 9.2). Pour c_6, C4.5 produit une seule règle, basée sur la présence du mot 'auto', qui provient des expressions "*auto manufacters*" ou "*auto shows*", dans le corpus du LAT ; or les expressions françaises "*salon de l'auto*" et "*industrie automobile*" des articles du Monde (LM) sont respectivement traduites par "*car show*" et "*car industries*" : le terme "*auto*" étant traduit par "*car*", le terme "*auto*" est absent des traductions de LM ; C4.5 ne peut donc appliquer son (unique) règle apprise.

9.4. Discussion

Les résultats obtenus par nos modèles sur notre corpus sont des résultats encourageants. Cette section propose de discuter les différents choix effectués pour chacune des étapes du processus afin de moduler la signification de nos résultats.

La première étape du processus consiste à définir une représentation du corpus par des termes. Nous avons choisi ici la représentation basée sur les mots et celle basée sur les n-grammes. Ce dernier choix était motivé par la capacité des n-grammes à capturer aisément les structures informationnelles basiques en s'affranchissant des problèmes de séparation des mots, de coquilles et tout autre aspect linguistique. Nos expériences n'ont pas montré une différentiation marquée entre les résultats issus d'une sélection de mots et d'une sélection de n-

(a) représentation avec 100 mots

	classifieur SDA (Ger) appliqué sur		**classifieur LAT (An) appliqué sur**			
	SDA (Ger)		**SDA (Ger)**		**SDA (An)**	
	3-NN	**C4.5**	**3-NN**	**C4.5**	**3-NN**	**C4.5**
$\rho^\mu = \pi^\mu$	97%	85%	20%	17%	97%	56%
ρ^M	93%	77%	13%	14%	95%	50%
π^M	95%	70%	12%	25%	94%	62%

(b) représentation avec 200 4-grammes

	classifieur SDA (Ger) appliqué sur		**classifieur LAT (An) appliqué sur**			
	SDA (Ger)		**SDA (Ger)**		**SDA (An)**	
	3-NN	**C4.5**	**3-NN**	**C4.5**	**3-NN**	**C4.5**
$\rho^\mu = \pi^\mu$	98%	92%	21%	17%	97%	54%
ρ^M	94%	80%	14%	14%	97%	52%
π^M	94%	78%	12%	25%	96%	54%

TAB. 9.6.: Précision et Rappel (micro et macro-moyen) pour le corpus allemand SDA représenté par 100 mots (table a) et pour 200 4-grammes (table b). Les deux tables montrent les performances obtenues en validation croisée : On applique le modèle SDA (Ger) sur des textes écrits en allemand et les résultats obtenus sont comparés avec ceux du modèle LAT anglais sur le corpus "SDA écrit en allemand" et sur le corpus "SDA traduit en anglais"

grammes ; ceci montre. Nous attribuons ces similarités à la qualité contrôlée des corpus. En effet, ces derniers sont destinés à la presse écrite, qui est exigeante envers les fautes d'orthographes et coquilles.

La deuxième étape consiste en la sélection des termes. Les coprésences des mots sélectionnés à partir du corpus du LAT et de celui du LM traduit en anglais se situent au niveau de 50 %. La moitié d'entre-eux est constituée de mots communs. Nous obtenons des résultats similaires lors de la comparaison des termes sélectionnés à partir du corpus du LAT et de celui du SDA traduit en anglais. Ceci montre la similitude informationnelle de nos 3 corpus. Par ailleurs, la forte quantité de noms propres n'est pas un problème en lui-même, et nous avons vu que son apport était faible. Cependant, lorsque le traducteur ne connaît pas un terme il le laisse tel quel. De plus, les noms propres ont une faible variabilité d'écriture dans les différentes langues. Ainsi, nous pensons que les noms propres empêchent l'évaluation complète de l'effet de la traduction.

Enfin, la sélection des termes est qualitativement intéressante puisqu'elle permet une séparabilité aisée de nos 10 classes. Là encore, nous nous interrogeons sur la constitution de notre corpus. Le corpus CLEF contient 96 000 articles. De ces 96 000, seuls 8 000 ont été assigné à 50 sujets (classes). Pour améliorer notre corpus et confirmer nos résultats, nous envisageons la généralisation en travaillant sur l'ensemble des 96 000 textes (8 000 assignés à des classes définies et les 88 000 restants assignés à une classe “autre”) ceci rendra plus difficile la séparabilité des sujets.

La dernière étape concerne l'apprentissage ; elle a montré les bons résultats du '3-ppv' sur ces corpus. Ce résultat est en opposition avec la difficulté de prédiction des k-ppv dans un espace creux et/ou avec des variables non pertinentes. Nous attribuons donc ces bons résultats des “3-ppv” à la qualité de notre espace de représentation (bon choix des descripteurs par notre méthode du χ^2_{multi} multivarié). Enfin, le C4.5 donne des résultats convenables mais est moins performant que le 3-ppv. Comme nous l'avons vu, cela est du au faible effectifs de certaines classes et au paramétrage de la méthode.

9.5. Conclusion

L'objet de ce chapitre est la définition d'un processus pour la catégorisation multilingue. Nous avons introduit deux nouvelles étapes par rapport au processus monolingue : la détection de la langue du texte à catégoriser et sa traduction dans la langue du corpus d'apprentissage. Nous avons illustré notre procédé par une application sur des corpus réels de journaux écrits en 3 langues (anglaise, française et allemande). Nous avons décrit chaque étape de notre processus et présenté les résultats de nos expériences. Nous concluons à la l'efficacité de notre approche.

Nous envisageons de perfectionner notre cadre pour la catégorisation de textes multilingues en proposant un schéma plus général dans lequel nous fusionnerons des corpus d'apprentissage traduits en une langue commune d'apprentissage (voir la figure 6.4 page 96). Nous espérons ainsi que les particularités propres à chaque langue ne seront plus retenues par les modèles estimés.

		LAT C.V.		LAT classifier applied on LM		LAT classifier applied on SDA		LM C.V.		SDA C.V.	
		3-NN	C4.5	3-NN	C4.5	3-NN	C4.5	3-NN	C4.5	3-NN	C4.5
c_1	ρ	100%	93%	100%	92%	100%	91%	100%	83%	100%	100%
	π	93%	96%	96%	100%	100%	84%	96%	87%	100%	88%
c_2	ρ	99%	97%	100%	94%	100%	97%	100%	93%	100%	95%
	π	98%	100%	100%	100%	100%	97%	100%	99%	100%	100%
c_3	ρ	100%	0%	96%	0%	97%	0%	96%	96%	100%	99%
	π	100%	0%	82%	0%	100%	0%	85%	88%	100%	99%
c_4	ρ	83%	100%	100%	84%	100%	81%	100%	11%	100%	95%
	π	100%	32%	100%	12%	100%	28%	100%	17%	100%	100%
c_5	ρ	96%	93%	100%	85%	100%	96%	97%	97%	100%	93%
	π	90%	81%	87%	78%	96%	100%	91%	100%	98%	100%
c_6	ρ	93%	70%	96%	0%	100%	17%	91%	30%	50%	0%
	π	93%	86%	88%	0%	50%	100%	88%	44%	75%	0%
c_7	ρ	72%	75%	90%	73%	100%	100%	92%	79%	96%	92%
	π	88%	80%	74%	51%	80%	41%	86%	70%	96%	88%
c_8	ρ	93%	93%	67%	84%	84%	5%	79%	63%	93%	98%
	π	84%	97%	76%	82%	95%	67%	89%	96%	98%	98%
c_9	ρ	80%	0%	95%	0%	100%	0%	95%	52%	100%	0%
	π	67%	0%	87%	0%	100%	0%	87%	44%	94%	0%
c_{10}	ρ	93%	79%	78%	29%	70%	17%	91%	80%	91%	96%
	π	98%	96%	97%	97%	100%	100%	98%	64%	84%	27%
μ	$\rho^\mu = \pi^\mu$	92%	85%	92%	54%	95%	50%	95%	76%	97%	85%
M	ρ^M	91%	70%	89%	52%	92%	62%	94%	68%	93%	77%
	π^M	91%	67%	90%	59%	96%	56%	92%	71%	95%	70%

TAB. 9.7.: Rappel et Précision sur le corpus anglais (LAT) décrit par 100 mots

		Appris et appliqué sur LAT, validation croisée		Appris sur LAT, appliqué à LM		Appris sur LAT, appliqué à SDA		Appris et appliqué sur LM, validation croisée		Appris et appliqué sur SDA, validation croisée	
		3-NN	**C4.5**	**3-NN**	**C4.5**	**3-NN**	**C4.5**	**3-NN**	**C4.5**	**3-NN**	**C4.5**
c_1	ρ	100%	96%	100%	96%	100%	100%	100%	100%	100%	96%
	π	100%	96%	92%	96%	100%	100%	100%	67%	100%	85%
c_2	ρ	99%	99%	100%	99%	100%	98%	100%	100%	100%	94%
	π	98%	99%	100%	98%	100%	97%	99%	100%	100%	98%
c_3	ρ	100%	0%	100%	0%	100%	0%	100%	96%	100%	99%
	π	100%	0%	92%	0%	100%	0%	92%	92%	100%	99%
c_4	ρ	94%	78%	100%	74%	100%	49%	100%	32%	100%	100%
	π	94%	34%	95%	16%	100%	25%	100%	35%	100%	100%
c_5	ρ	94%	94%	100%	100%	98%	96%	97%	100%	100%	96%
	π	88%	85%	89%	72%	95%	100%	97%	100%	100%	96%
c_6	ρ	89%	70%	96%	0%	100%	17%	100%	91%	67%	0%
	π	89%	86%	92%	0%	75%	100%	88%	88%	67%	0%
c_7	ρ	72%	69%	92%	67%	96%	88%	96%	85%	96%	71%
	π	82%	67%	72%	49%	79%	36%	82%	82%	88%	77%
c_8	ρ	90%	93%	65%	81%	79%	5%	77%	70%	86%	79%
	π	86%	93%	93%	74%	100%	50%	97%	88%	93%	72%
c_9	ρ	70%	0%	67%	0%	100%	0%	95%	29%	100%	65%
	π	88%	0%	88%	0%	89%	0%	91%	46%	100%	55%
c_{10}	ρ	94%	89%	91%	58%	91%	65%	93%	97%	91%	100%
	π	96%	95%	96%	90%	100%	25%	99%	94%	88%	96%
μ	$\rho^\mu = \pi^\mu$	93%	84%	91%	58%	96%	52%	95%	84%	98%	92%
M	ρ^M	90%	69%	91%	50%	94%	53%	96%	80%	94%	80%
	π^M	92%	66%	91%	67%	97%	54%	95%	79%	94%	78%

TAB. 9.8.: Rappel et Précision sur le corpus anglais (LAT) décrit par 200 4-grammes

Conclusion et perspectives

Dans ce travail, nous avions comme objectif l'adaptation des techniques de l'apprentissage automatique au problème de la catégorisation de textes multilingues.

Nous présentons ici un cadre général pour catégoriser automatiquement des textes multilingues.

La catégorisation de textes dite "monolingue" suit habituellement le schéma suivant :

- **représentation des textes** dans un format adapté aux algorithmes d'apprentissage ; on utilise souvent la représentation vectorielle proposée par [Salton and McGill, 1983] mais d'autres modes de représentation existent aussi tels la représentation probabiliste (modèle de *Robertson et Sparck-Jones* [Robertson and K.Sparck-Jones, 1976] ou le modèle *Okapi* [Robertson et al., 1996]), où l'information concernant la position de termes dans les phrases peut être conservée, contrairement à ce que fait le modèle vectoriel de [Salton and McGill, 1983]. La représentation englobe trois éléments : le choix des termes, le choix des poids associés et le choix des méthodes de sélection de termes. Nous apportons deux contributions à ce sujet :
  - pour le choix de ce qui est un terme, nos travaux [Jalam and Chauchat, 2002] montrent les raisons de l'efficacité des n-grammes comme méthode de représentation. Ces travaux ont montré par exemple que les n-grammes, comme choix de représentation, capturent les connaissances contenues dans les mots. Ainsi, à partir des n-grammes, nous sommes capables d'extraire automatiquement des candidats mots-clés, sans utiliser aucune connaissance linguistique ; ceci est intéressant car l'indépendance de la méthode par rapport à la langue est nécessaire pour traiter les textes écrits en plusieurs langues.
  - pour la sélection de termes, nos travaux [Clech et al., 2003] proposent une nouvelle utilisation de la statistique du χ^2 (multivarié) calcu-

lée sur le tableau complet (termes×classes). Cette méthode prend en compte l'interaction entre les termes eux-même, et entre les termes et les classes. Nos premiers résultats sont encourageants et montrent une amélioration des performances par rapport à l'utilisation du χ^2 (univarié), utilisé pour la sélection des descripteurs par les auteurs précédents comme [Schütze et al., 1995, Wiener et al., 1995, Yang and Pedersen, 1997, He et al., 2000] ; ce χ^2_{uni} mesurait l'écart à l'indépendance entre un seul descripteur t_k (présent ou absent) et un seul thème c_i (présent ou absent).

- **choix d'une méthode d'apprentissage** pour construire un modèle de prédiction. Il s'agit de proposer une fonction $\Phi : \mathcal{D} \times \mathcal{C} \to \{V, F\}$ qui associe une ou plusieurs étiquettes (catégories) à un document d_j telle que la décision donnée coïncide « le plus possible » avec la fonction $\breve{\Phi} : \mathcal{D} \times \mathcal{C} \to \{V, F\}$, la vraie fonction qui retourne pour chaque vecteur d_j la valeur c_i de sa classe réelle. Plusieurs critères de choix sont possibles : si les résultats du classifieur sont destinés à des humains (experts ou décideurs), il est souhaitable de privilégier les méthodes "explicatives" comme les arbres de décision ; si, au contraire, le résultat produit est à intégrer dans un processus automatique, on peut alors choisir la méthode qui donne les meilleures performances.
  Dans [Jalam and Teytaud, 2001] et [Teytaud and Jalam, 2001] nous avons proposé de nouvelles utilisations des méthodes SVM et RBF en introduisant un nouveau noyau, fondé sur le χ^2, qui donne de bons résultats et qui confirment les résultats obtenus par [Yang, 1999, Sebastiani, 2002].
- **évaluation le modèle** appris afin de s'assurer qu'il est généralisable à d'autres textes.

Nous avons ensuite étendu la catégorisation aux **textes multilingues**. Dans ce cas, les méthodes utilisant des analyses linguistiques fines deviennent impraticables. Nous avons proposé une méthode générale, automatique et largement indépendante des langues.

La phase d'apprentissage s'effectue toujours de manière classique, à partir d'un corpus d'apprentissage étiqueté, rédigé dans une langue donnée. Pour classer un texte rédigé dans une langue quelconque, il faut d'abord identifier automatiquement la langue utilisée ; ensuite, il y a deux voies possibles (voir figure 9.1 page 121) :

- soit on applique un modèle propre à chaque langue ; ceci exige de disposer d'ensembles d'apprentissage (préalablement étiquetés manuellement) suffisamment vastes et variés dans chaque langue, ce qui est souvent hors de portée ;
- soit on utilise des traducteurs automatiques vers une langue fixe (disons l'anglais), puis on construit (par apprentissage automatique) un modèle unique de catégorisation. Il y a encore deux façons de construire le modèle,

en fonction du moment où l'on place l'étape de traduction, on apprend
– soit sur des textes écrits en anglais,
– soit sur les traductions vers l'anglais de textes écrits dans différentes langues.

Nous avons proposé trois schémas et nous en avons expérimenté deux.

Les perspectives

La thèse de doctorat n'est qu'une étape dans la vie d'un chercheur. A l'issue de ce travail, je vois de nombreuses pistes qui restent à explorer.

Expérimentation de l'apprentissage direct sur les résultats des traductions automatiques

On peut sans doute améliorer le modèle de classement en estimant ses paramètres sur les résultats du traducteur automatique pour les textes de l'échantillon d'apprentissage, et non pas sur les textes originaux écrits dans la langue cible. On exploiterait ainsi jusqu'aux erreurs du traducteur. C'est la dernière méthode que nous avons proposée, sans avoir encore eu le temps de l'expérimenter.

Adaptation du corpus CLEF pour en faire un jeu d'essai multilingue étiqueté

La catégorisation de texte multilingue est un domaine nouveau qui nécessite des corpus de référence pour tester les algorithmes. Après de nombreuses discussions avec des spécialistes en catégorisation de textes et dans la recherche documentaire multilingues (parmi lesquels David Lewis, Fabrizio Sebastiani, Douglas Oard, Carol Peters, et bien d'autres), nous avons conclu à l'absence de tels corpus. Nous avons donc créé un corpus multilingue à partir des collections proposées aux concurrents du concours CLEF de l'année 2002 (*The Cross-Language Evaluation Forum*, voir : `http://clef.iei.pi.cnr.it`). Les responsables du concours CLEF sont intéressés par le nouveau domaine d'application que je propose. Je vais donc développer une coopération avec eux pour contribuer à étendre CLEF à la classification multilingue.

Échantillonnage dans les textes

L'échantillonnage dans les textes est à explorer : au lieu de travailler sur la totalité d'un texte, on peut se limiter à travailler sur une partie. Actuellement, il y a peu de recherches sur ce sujet. Au moins deux problèmes se posent :

1. la détermination de "l'individu statistique" à échantillonner : un N-grammes, un mot, une phrase, un paragraphe, *etc.*

2. la méthode d'échantillonnage : échantillonnage aléatoire simple, ou bien sur-représentation de certaines parties des textes comme les titres, les résumés, les introductions, les conclusions, *etc.*

La recherche dans cette voie peut être fructueuse car un échantillon bien choisi peut être porteur de presque toute l'information du texte initial, et l'échantillonnage devrait accélérer les traitements (si la procédure d'échantillonnage elle-même n'est pas trop longue).

Applications à la veille technologique

La veille est l'une des clefs de succès des entreprises, comme des centres de recherche. Le problème central de la *veille* est de détecter *l'information cruciale* le plus tôt possible. La fouille de textes (ou *Text Mining*) est l'un des outils permettant d'analyser rapidement une grande quantité d'information. La nature multilingue des textes à analyser implique la généralisation des méthodes qui se développent en ce moment. La veille est particulièrement difficile car il faut détecter des « signaux faibles » dans un océan de « bruit ». Il y a donc un défi à relever pour les méthodes automatiques.

Approfondissement et généralisation du LSI (Latent Semantic Indexing) avec l'ensemble des méthodes d'analyse factorielle

Il nous semble que les techniques connues sous le nom de « LSI » peuvent être améliorées en utilisant l'ensemble des méthodes statistiques d'analyse des données multidimensionnelles, souvent mal connues de la communauté de « l'apprentissage automatique ». C'est l'une des voies à explorer.

Application en biologie sur le code génétique

Le code génétique est une sorte de texte, composé avec un alphabet de 4 caractères. Les codes des individus, ou des espèces sont comme de très longs textes, avec quelques fautes d'orthographe. Les méthodes d'analyse automatique de textes à partir du codage en n-grammes peuvent donc s'appliquer à la recherche sur les séquences génétiques.

Index des auteurs cités

Bibliographie

[Aas and Eikvil, 1999] Aas, K. and Eikvil, L. (1999). Text categorization : a survey. Technical report, Norwegian Computing Center. Available from World Wide Web : http://www.nr.no/documents/samba/research_areas/BAMG/Publications/tm_survey.ps.

[Adam et al., 2002] Adam, C. K., Ng, H. T., and Chieu, H. L. (2002). Bayesian online classifiers for text classification and filtering. In Beaulieu, M., Baeza-Yates, R., Myaeng, S. H., and Järvelin, K., editors, *Proceedings of SIGIR-02, 25th ACM International Conference on Research and Development in Information Retrieval*, pages 97–104, Tampere, FI. ACM Press, New York, US. Available from World Wide Web : http://doi.acm.org/10.1145/564376.564395.

[Afnor, 1993] Afnor (1993). Information et documentation. Principes généraux pour l'indexation des documents. NF Z 47-102. Paris.

[Aha, 1997] Aha, D. W. (1997). Editorial on lazy learning. *Artificial Intelligence Review*, 11 :7–10.

[Aha et al., 1991] Aha, D. W., Kibler, D., and Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6 :37–66.

[Amati and Crestani, 1999] Amati, G. and Crestani, F. (1999). Probabilistic learning for selective dissemination of information. *Information Processing and Management*, 35(5) :633–654. Available from World Wide Web : http://www.cs.strath.ac.uk/~fabioc/papers/99-ipem.pdf.

[Amini, 2001] Amini, M.-R. (2001). *Apprentissage automatique et Recherche d'information : application à l'extraction d'information de surface et au résumé de texte*. PhD thesis, Université Paris 6, Paris, France.

[Androutsopoulos et al., 2000] Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., and Spyropoulos, C. D. (2000). An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. In Belkin, N. J., Ingwersen, P., and Leong, M.-K., editors, *Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval*, pages 160–167, Athens, GR. ACM Press, New York, US. Available from World Wide Web : http://doi.acm.org/10.1145/345508.345569.

[Apté et al., 1994] Apté, C., Damerau, F. J., and Weiss, S. M. (1994). Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3) :233–251. Available from World Wide Web : http://www.acm.org/pubs/articles/journals/tois/1994-12-3/p233-apte/p233-apte.pdf.

[Ballesteros and Croft, 1998] Ballesteros, L. and Croft, W. B. (1998). Resolving ambiguity for cross-language retrieval. In Press, A., editor, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 64–71, Melbourne, Vic., Australia. Available from World Wide Web : http://citeseer.nj.nec.com/ballesteros98resolving.html.

[Beesley, 1988] Beesley, K. (1988). Language Identifier : A Computer Program for Automatic Natural Language Identification on On-Line Text. In *Proceedings of the 29th Annual Conference of the American Translators Association*, pages 47–54.

[Benzecri, 1973] Benzecri, J. P. (1973). *L'Analyse des Données*, volume 1. Dunod, Paris.

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

[Biskri and Delisle, 2001] Biskri, I. and Delisle, S. (2001). Les n-grams de caractères pour l'extraction de connaissances dans des bases de données textuelles multilingues. In *TALN*, pages 93–102. Available from World Wide Web : http://www.uqtr.ca/~biskri/Personnel/articles/taln2001.doc.

[Borko and Bernick, 1964] Borko, H. and Bernick, M. (1964). Automatic document classification. part ii : additional experiments. *Journal of the Association for Computing Machinery*, 11(2) :138–151. Available from World Wide Web : http://www.acm.org/pubs/articles/journals/jacm/1964-11-2/p138-borko/p138-borko.pdf.

[Breiman et al., 1984] Breiman, L., Friedman, J., Olshen, R. A., and Stone, C. J. (1984). *Classification and regression trees*. Wadsworth International Group, Belmont, CA.

[Broomhead and Loewe, 1988] Broomhead, D. and Loewe, D. (1988). Multivariate functional inter- polation and adaptive networks. *Complex Systems*, (2) :321–355.

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

[Buckley et al., 1994] Buckley, C., Salton, G., Allan, J., and Singhal, A. (1994). Automatic query expansion using SMART : TREC 3. In *Text REtrieval Conference*. Available from World Wide Web : http://citeseer.nj.nec.com/37681.html.

[Burges, 1998] Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2) :121–167.

[Caropreso et al., 2001] Caropreso, M. F., Matwin, S., and Sebastiani, F. (2001). A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In Chin, A. G., editor, *Text Databases and Document Management : Theory and Practice*, pages 78–102. Idea Group Publishing, Hershey, US. Available from World Wide Web : http://faure.iei.pi.cnr.it/~fabrizio/Publications/TD01a.pdf.

[Carreras and Márquez, 2001] Carreras, X. and Márquez, L. (2001). Boosting trees for anti-spam email filtering. In *Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing*, Tzigov Chark, BG. Available from World Wide Web : http://www.lsi.upc.es/~carreras/pub/boospam.ps.

[Cavnar and Trenkle, 1994] Cavnar, W. B. and Trenkle, J. M. (1994). N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, US. Available from World Wide Web : http://www.nonlineardynamics.com/trenkle/papers/sdair-94-bc.ps.gz.

[Chai et al., 2002] Chai, K. M., Ng, H. T., and Chieu, H. L. (2002). Bayesian online classifiers for text classification and filtering. In Beaulieu, M., Baeza-Yates, R., Myaeng, S. H., and Järvelin, K., editors, *Proceedings of SIGIR-02, 25th ACM International Conference on Research and Development in Information Retrieval*, pages 97–104, Tampere, FI. ACM Press, New York, US. Available from World Wide Web : http://doi.acm.org/10.1145/564376.564395.

[Chandioux, 1998] Chandioux, J. (1998). Les promesses de la traduction automatique. *Terminogramme*, (84-85).

[Chauchat, 1975] Chauchat, J. H. (1975). Analyse de données sur un problème de transport : zonage d'après une matrice de déplacements. *Publication Econométriques*, VIII(2) :91–117.

[Clech et al., 2003] Clech, J., Rakotomalala, R., and Jalam, R. (2003). Séléction multivariée de termes. In *SFDS03*. à apparaître.

[Cleverdon, 1984] Cleverdon, C. (1984). Optimizing convenient online access to bibliographic databases. *Information Services and Use*, 4 :37–47.

[Cohen, 1996] Cohen, W. W. (1996). Learning rules that classify e-mail. In *The 1996 AAAI Spring Symposium on Machine Learning in Information Access*, pages 18–25. Available from World Wide Web : http://www-2.cs.cmu.edu/~wcohen/postscript/aaai-ss-96.ps.

[Cohen and Hirsh, 1998] Cohen, W. W. and Hirsh, H. (1998). Joins that generalize : text classification using WHIRL. In Agrawal, R., Stolorz, P. E., and Piatetsky-Shapiro, G., editors, *Proceedings of KDD-98, 4th International Conference on Knowledge Discovery and Data Mining*, pages 169–173, New York, US. AAAI Press, Menlo Park, US. Available from World Wide Web : http://www.research.whizbang.com/~wcohen/postscript/kdd-98.ps.

[Cohen and Singer, 1999] Cohen, W. W. and Singer, Y. (1999). Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems*, 17(2) :141–173. Available from World Wide Web : http://www.acm.org/pubs/articles/journals/tois/1999-17-2/p141-cohen/p141-cohen.pdf.

[Cormier, 1992] Cormier, M. C. (1992). L'évaluation des systèmes de traduction automatique. *Meta*, 37(2).

[Cornuéjols, 2002] Cornuéjols, A. (2002). Une nouvelle méthode d'apprentissage : Les svm. séparateurs à vastes marges. *Bulletin de l'AFIA*, 51 :14–23.

[Cover and Hart, 1967] Cover, T. M. and Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13 :21–27.

[Cowie et al., 1998] Cowie, J., Ludovik, E., and Zacharski, R. (1998). An Autonomous, Web-based, Multilingual Corpus Collection Tool. In *Proceeding of Natural Language Processing and Industrial Applications*, pages 142–148, Moncton, Canada.

[Creecy et al., 1992] Creecy, R. M., Masand, B. M., Smith, S. J., and Waltz, D. L. (1992). Trading MIPS and memory for knowledge engineering : classifying census returns on the Connection Machine. *Communications of the ACM*, 35(8) :48–63. Available from World Wide Web : http://www.acm.org/pubs/articles/journals/cacm/1992-35-8/p48-creecy/p48-creecy.pdf.

[Damashek, 1995] Damashek, M. (1995). Gauging similarity with N-grams : Language-independent categorization of text. *Science*, 267(5199) :843–848.

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

[Deerwester et al., 1990] Deerwester, S., Dumais, S., Landauer, T., Furnas, G., and Harshman, R. (1990). Indexing by latent semantic anlysis. *Journal of the American Society of Information science*, 416(6) :391–407.

[Déjean, 1998] Déjean, H. (1998). Morphemes as Necessary Concept for Structures Discovery from Untagged Corpora. In *Workshop on Paradigms and Grounding in Natural Language Learning*, pages 295–299, Adélaide, Australie.

[Dumais, 1991] Dumais, S. (1991). Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments & Computers*, 23(2) :229–236.

[Dumais et al., 1998] Dumais, S., Platt, J., Heckerman, D., and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In Gardarin, G., French, J. C., Pissinou, N., Makki, K., and Bouganim, L., editors, *Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management*, pages 148–155, Bethesda, US. ACM Press, New York, US. Available from World Wide Web : http://robotics.stanford.edu/users/sahami/papers-dir/cikm98.pdf.

[Dunning, 1994] Dunning, T. (1994). Statistical Identification of Languages. Technical Report MCCS 94-273, Computing Research Laboratory.

[Elisseeff, 2000] Elisseeff, A. (2000). *Etude de la colmplexité et contrôle de la capacité des systèmes d'apprentissage : SVM multi-classe, réseaux de régularisation et réseaux de neurones multicouches*. PhD thesis, Ecole normale supérieure de Lyon, Lyon, France.

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

[Escudero et al., 2000] Escudero, G., Màrquez, L., and Rigau, G. (2000). Boosting applied to word sense disambiguation. In de Mántaras, R. L. and Plaza, E., editors, *Proceedings of ECML-00, 11th European Conference on Machine Learning*, pages 129–141, Barcelona, ES. Springer Verlag, Heidelberg, DE. Available from World Wide Web : http://www.lsi.upc.es/~escudero/recerca/ecml00.pdf. Published in the "Lecture Notes in Computer Science" series, number 1810.

[Fix and Hodges, 1951] Fix, E. and Hodges, J. L. (1951). Discriminatory analysis –nonparametric discrimination : Consistency properties. Technical Report 21-49-004, report no. 04, USAF School of Aviation Medicine, Randolph Field, Texas.

[Fix and Hodges, 1952] Fix, E. and Hodges, J. L. (1952). Discriminatory analysis –nonparametric discrimination : Small sample performance. Technical Report 21-49-004, report no. 11, USAF School of Aviation Medicine, Randolph Field, Texas.

[Fluhr, 1995] Fluhr, C. (1995). Multilingual information retrieval. In Cole, R. A., Mariani, J., Uszkoreit, H., Zaenen, A., and Zue, V., editors, *Survey of the State of the Art in Human Language Technology*.

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

[Fürnkranz, 1998] Fürnkranz, J. (1998). A Study Using n-gram Features for Text Categorization. Technical Report OEFAI-TR-98-30, Austrian Research Institute for Artificial Intelligence, Austria.

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

[Fuhr et al., 1991] Fuhr, N., Hartmann, S., Knorz, G., Lustig, G., Schwantner, M., and Tzeras, K. (1991). AIR/X – a rule-based multistage indexing system for large subject fields. In Lichnerowicz, A., editor, *Proceedings of RIAO-91, 3rd International Conference "Recherche d'Information Assistee par Ordinateur"*, pages 606–623, Barcelona, ES. Elsevier Science Publishers, Amsterdam, NL. Available from World Wide Web : http://www.darmstadt.gmd.de/~tzeras/FullPapers/gz/Fuhr-etal-91.ps.gz.

[Fuhr and Knorz, 1984] Fuhr, N. and Knorz, G. (1984). Retrieval test evaluation of a rule-based automated indexing (AIR/PHYS). In van Rijsbergen, C. J., editor, *Proceedings of SIGIR-84, 7th ACM International Conference on Research and Development in Information Retrieval*, pages 391–408, Cambridge, UK. Cambridge University Press.

[Giguet, 1998] Giguet, E. (1998). *Méthode pour l'analyse automatique de structures formelles sur documents multilingues*. PhD thesis, Université de Caen, France.

[Gilleron and Tommasi, 2000] Gilleron, R. and Tommasi, M. (2000). Découverte de connaissances à partir de données. Cours donné à l'IUP MIAGE troisième année à Lille 1.

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

[Good, 1965] Good, I. (1965). *The Estimation of Probabilities : An Essay on Modern Bayesian Methods*. MIT Press.

[Grefenstette, 1995] Grefenstette, G. (1995). Comparing Two Language Identification Schemes. In *Proceedings of the 3rd International Conference on the Statistical Analysis of Textual Data (JADT'95)*, Rome, Italy.

[Guermeur et al., 2000] Guermeur, Y., Elisseeff, A., and Paugam-Moisy, H. (2000). A new multiclass SVM based on a uniform convergence result.

[Hastie et al., 2001] Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer–Verlag.

[Hayes and Weinstein, 1990] Hayes, P. and Weinstein, S. P. (1990). Construe/Tis : a system for content-based indexing of a database of news stories. In Rappaport, A. and Smith, R., editors, *Proceedings of IAAI-90, 2nd Conference on Innovative Applications of Artificial Intelligence*, pages 49–66. AAAI Press, Menlo Park, US.

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

[Hovy et al., 2002] Hovy, E., King, M., and Popescu-Belis, A. (2002). An introduction to mt evaluation. In King, M. e., editor, *Workbook of the LREC 2002 Workshop on Machine Translation Evaluation : Human Evaluators Meet Automated Metrics*, pages 1–7, Las Palmas de Gran Canaria, Spain.

[Hull, 1994] Hull, D. A. (1994). Improving text retrieval for the routing problem using latent semantic indexing. In Croft, W. B. and van Rijsbergen, C. J., editors, *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 282–289, Dublin, IE. Springer Verlag, Heidelberg, DE. Available from World Wide Web : http://www.acm.org/pubs/articles/proceedings/ir/188490/p282-hull/p282-hull.pdf.

[Hull, 1996] Hull, D. A. (1996). Stemming algorithms : A case study for detailed evaluation. *Journal of the American Society of Information Science*, 47(1) :70–84.

[Hull and Grefenstette, 1996] Hull, D. A. and Grefenstette, G. (1996). Querying across languages : A dictionary-based approach to multilingual information retrieval. In Frei, H.-P., Harman, D., Schäuble, P., and Wilkinson, R., editors, *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'96, August 18-22, 1996, Zurich, Switzerland (Special Issue of the SIGIR Forum)*, pages 49–57. ACM.

[Iyer et al., 2000] Iyer, R. D., Lewis, D. D., Schapire, R. E., Singer, Y., and Singhal, A. (2000). Boosting for document routing. In Agah, A., Callan, J., and Rundensteiner, E., editors, *Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management*, pages 70–77, McLean, US. ACM Press, New York, US. Available from World Wide Web : http://www.cs.huji.ac.il/~singer/papers/rankboost.ps.gz.

[Jalam and Chauchat, 2002] Jalam, R. and Chauchat, J.-H. (2002). Pourquoi les n-grammes permettent de classer des textes ? Recherche de mots-clefs pertinents à l'aide des n-grammes caractéristiques. In Morin, A. and Sébillot, P., editors, *6th International Conference on Textual Data Statistical Analysis*, volume 1, pages 381–390, St. Malo France. IRISA, INRIA.

[Jalam and Teytaud, 2001] Jalam, R. and Teytaud, O. (2001). Identification de la langue et catégorisation de textes basées sur les n-grammes. *ECA*, 1(1-2) :227–238.

[Joachims, 1997] Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In Fisher, D. H., editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 143–151, Nashville, US. Morgan Kaufmann Publishers, San Francisco, US. Available from World Wide Web : http://www-ai.cs.uni-dortmund.de/DOKUMENTE/joachims_97a.ps.gz.

[Joachims, 1998] Joachims, T. (1998). Text categorization with support vector machines : learning with many relevant features. In Nédellec, C. and Rouveirol, C., editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, DE. Springer Verlag, Heidelberg, DE. Available from World Wide Web : http://www-ai.cs.uni-dortmund.de/DOKUMENTE/joachims_98a.ps.gz. Published in the "Lecture Notes in Computer Science" series, number 1398.

[Joachims, 1999] Joachims, T. (1999). Transductive inference for text classification using support vector machines. In Bratko, I. and Dzeroski, S., editors, *Proceedings of ICML-99, 16th International Conference on Machine Learning*, pages 200–209, Bled, SL. Morgan Kaufmann Publishers, San Francisco, US. Available from World Wide Web : http://www-ai.cs.uni-dortmund.de/DOKUMENTE/joachims_99c.ps.gz.

[Joachims, 2000] Joachims, T. (2000). Estimating the generalization performance of a SVM efficiently. In Langley, P., editor, *Proceedings of ICML-00, 17th International Conference on Machine Learning*, pages 431–438, Stanford, US. Morgan Kaufmann Publishers, San Francisco, US. Available from World Wide Web : http://www-ai.cs.uni-dortmund.de/DOKUMENTE/joachims_00a.pdf.

[Joachims, 2001] Joachims, T. (2001). A statistical learning model of text classification with support vector machines. In Croft, W. B., Harper, D. J., Kraft, D. H., and Zobel, J., editors, *Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval*, pages 128–136, New Orleans, US. ACM Press, New York, US. Available from World Wide Web : http://www.cs.cornell.edu/People/tj/publications/joachims_01a.pdf.

[Kim et al., 2000] Kim, Y.-H., Hahn, S.-Y., and Zhang, B.-T. (2000). Text filtering by boosting naive Bayes classifiers. In Belkin, N. J., Ingwersen, P., and Leong, M.-K., editors, *Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval*, pages 168–75, Athens, GR. ACM Press, New York, US. Available from World Wide Web : http://www.acm.org/pubs/articles/proceedings/ir/345508/p168-kim/p168-kim.pdf.

[King, 1992] King, M. (1992). L'évaluation des systèmes de traduction automatique dans le cadre d'un service de traduction. *Meta*, 37(4).

[Kodratoff, 2001] Kodratoff, Y. (2001). *Machine Learning and Its Applications*, chapter Comparing Machine Learning and Knowledge Discovery in DataBases : An Application to Knowledge Discovery in Texts, pages 1–21. Springer Verlag LNAI 2049.

[Kohavi, 1995] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, pages 1137–1145, Montréal. Available from World Wide Web : http://citeseer.nj.nec.com/kohavi95study.html.

[Koller and Sahami, 1997] Koller, D. and Sahami, M. (1997). Hierarchically classifying documents using very few words. In Fisher, D. H., editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 170–178, Nashville, US. Morgan Kaufmann Publishers, San Francisco, US. Available from World Wide Web : http://robotics.stanford.edu/users/sahami/papers-dir/ml97-hier.ps.

[Lam et al., 1999] Lam, W., Ruiz, M. E., and Srinivasan, P. (1999). Automatic text categorization and its applications to text retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 11(6) :865–879. Available from World Wide Web : http://www.cs.uiowa.edu/~mruiz/papers/IEEE-TKDE.ps.

[Larkey, 1998] Larkey, L. S. (1998). Automatic essay grading using text categorization techniques. In Croft, W. B., Moffat, A., van Rijsbergen, C. J., Wilkinson, R., and Zobel, J., editors, *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 90–95, Melbourne, AU. ACM Press, New York, US. Available from World Wide Web : http://cobar.cs.umass.edu/pubfiles/ir-121.ps.

[Larkey, 1999] Larkey, L. S. (1999). A patent search and classification system. In Fox, E. A. and Rowe, N., editors, *Proceedings of DL-99, 4th ACM Conference on Digital Libraries*, pages 179–187, Berkeley, US. ACM Press, New York, US. Available from World Wide Web : http://cobar.cs.umass.edu/pubfiles/ir-162.ps.

[Larkey and Croft, 1996] Larkey, L. S. and Croft, W. B. (1996). Combining classifiers in text categorization. In Frei, H.-P., Harman, D., Schäuble, P., and Wilkinson, R., editors, *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pages 289–297, Zürich, CH. ACM Press, New York, US. Available from World Wide Web : http://cobar.cs.umass.edu/pubfiles/1combo.ps.gz.

[Lebart, 2001] Lebart, L. (2001). Classification of textual data. Séminaire de recherche à the School of Computer Science and Information Systems, Birkbeck College. Available from World Wide Web : http://www.dcs.bbk.ac.uk/seminars/autumn01.html.

[Lebart and Salem, 1994] Lebart, L. and Salem, A. (1994). *Statisque textuelle*. Dunod, Paris.

[Lefèvre, 2000] Lefèvre, P. (2000). *La rechreche d'information - du texte intégral au thésaurus*. Hermès Science, Paris.

[Lelu and Hallab, 2000] Lelu, A. and Hallab, M. (2000). Consultation "floue" de grandes listes de formes lexicales simples et composées : un outil préparatoire pour l'analyse de grands corpus textuels. In Rajmann, M. and Chappelier, J. C., editors, *JADT'2000*, volume 1, pages 317–324, Lausanne.

[Lewis, 1992a] Lewis, D. D. (1992a). An evaluation of phrasal and clustered representations on a text categorization task. In Belkin, N. J., Ingwersen, P., and Pejtersen, A. M., editors, *Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval*, pages 37–50, Kobenhavn, DK. ACM Press, New York, US. Available from World Wide Web : http://www.research.att.com/~lewis/papers/lewis92b.ps.

[Lewis, 1992b] Lewis, D. D. (1992b). *Representation and learning in information retrieval*. PhD thesis, Department of Computer Science, University of Massachusetts, Amherst, US. Available from World Wide Web : http://www.research.att.com/~lewis/papers/lewis91d.ps.

[Lewis, 1995] Lewis, D. D. (1995). Evaluating and optmizing autonomous text classification systems. In Fox, E. A., Ingwersen, P., and Fidel, R., editors, *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, pages 246–254, Seattle, US. ACM Press, New York, US. Available from World Wide Web : http://www.research.att.com/~lewis/papers/lewis95b.ps.

[Lewis, 1998] Lewis, D. D. (1998). Naive (Bayes) at forty : The independence assumption in information retrieval. In Nédellec, C. and Rouveirol, C., editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 4–15, Chemnitz, DE. Springer Verlag, Heidelberg, DE. Available from World Wide Web : http://www.research.att.com/~lewis/papers/lewis98b.ps. Published in the "Lecture Notes in Computer Science" series, number 1398.

[Lewis and Catlett, 1994] Lewis, D. D. and Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In Cohen, W. W. and Hirsh, H., editors, *Proceedings of ICML-94, 11th International Conference on Machine Learning*, pages 148–156, New Brunswick, US. Morgan Kaufmann Publishers, San Francisco, US. Available from World Wide Web : http://www.research.att.com/~lewis/papers/lewis94e.ps.

[Lewis and Ringuette, 1994] Lewis, D. D. and Ringuette, M. (1994). A comparison of two learning algorithms for text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, Las Vegas, US. Available from World Wide Web : http://www.research.att.com/~lewis/papers/lewis94b.ps.

[Lewis et al., 1996] Lewis, D. D., Schapire, R. E., Callan, J. P., and Papka, R. (1996). Training algorithms for linear text classifiers. In Frei, H.-P., Harman, D., Schäuble, P., and Wilkinson, R., editors, *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pages 298–306, Zürich, CH. ACM Press, New York, US. Available from World Wide Web : http://www.research.att.com/~lewis/papers/lewis96d.ps.

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

[Liddy et al., 1994] Liddy, E. D., Paik, W., and Yu, E. S. (1994). Text categorization for multiple users based on semantic features from a machine-readable dictionary. *ACM Transactions on Information Systems*, 12(3) :278–295. Available from World Wide Web : http://www.acm.org/pubs/articles/journals/tois/1994-12-3/p278-liddy/p278-liddy.pdf.

[Liu and Motoda, 1998] Liu, H. and Motoda, H. (1998). *Feature selection for knowledge discovery and data mining*, volume 454 of *The kluwer international series in engineering and computer science*. Kluwer.

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

[Manning and Schütze, 1999] Manning, C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*, chapter 16 : Text Categorization, pages 575–608. The MIT Press, Cambridge, US.

[Maron, 1961] Maron, M. (1961). Automatic indexing : an experimental inquiry. *Journal of the Association for Computing Machinery*, 8(3) :404–417. Available from World Wide Web : http://www.acm.org/pubs/articles/journals/jacm/1961-8-3/p404-maron/p404-maron.pdf.

[Miller et al., 1999] Miller, E., Shen, D., Liu, J., and C.Nicholas (1999). Performance and Scalability of a Large-Scale N-gram Based Information Retrieval System. *Journal of Digital Information*, 1(5).

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

[Mladenić, 1998] Mladenić, D. (1998). Feature subset selection in text learning. In Nédellec, C. and Rouveirol, C., editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 95–100, Chemnitz, DE. Springer Verlag, Heidelberg, DE. Available from World Wide Web : http://www-ai.ijs.si/DunjaMladenic/papers/PWW/pwwECML98.ps.gz. Published in the “Lecture Notes in Computer Science” series, number 1398.

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

[Moody and Darken, 1989] Moody, J. and Darken, C. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(2) :281–294.

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

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

[Moulinier, 1997] Moulinier, I. (1997). Feature selection : a useful preprocessing step. In Furner, J. and Harper, D., editors, *Proceedings of BCSIRSG-97, the 19th Annual Colloquium of the British Computer Society Information Retrieval Specialist Group*, Electronic Workshops in Computing, Aberdeen, UK. Springer Verlag, Heidelberg, DE. Available from World Wide Web : http://www.ewic.org.uk/ewic/workshop/fetch.cfm/IRR-97/Moulinier/Moulinier.ps.

[Moulinier and Ganascia, 1996] Moulinier, I. and Ganascia, J.-G. (1996). Applying an existing machine learning algorithm to text categorization. In Wermter, S., Riloff, E., and Scheler, G., editors, *Connectionist, statistical, and symbolic approaches to learning for natural language processing*, pages 343–354. Springer Verlag, Heidelberg, DE. Available from World Wide Web : http://www-poleia.lip6.fr/~moulinie/wijcai.ps.gz. Published in the “Lecture Notes in Computer Science” series, number 1040.

[Muhlenbach, 2002] Muhlenbach, F. (2002). *Evaluation de la qualité de la représentation en fouille de données*. PhD thesis, Laboratoire ERIC, Université Lumière Lyon 2, France.

[Mustonen, 1965] Mustonen, S. (1965). Multiple Disriminant Analysis in Linguistic Problems. *Statistical Methods in Linguistics*. Page visitée le 15 juin 2000 à l'adresse http ://www.nodali.sics.se/bibliotek/kval/smil.

[Ng et al., 1997] Ng, H. T., Goh, W. B., and Low, K. L. (1997). Feature selection, perceptron learning, and a usability case study for text categorization.

In Belkin, N. J., Narasimhalu, A. D., and Willett, P., editors, *Proceedings of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval*, pages 67–73, Philadelphia, US. ACM Press, New York, US. Available from World Wide Web : http://www.acm.org/pubs/articles/proceedings/ir/258525/p67-ng/p67-ng.pdf.

[Nunberg, 2000] Nunberg, G. (2000). Will the internet speak english ? *The American Prospect*.

[Oard, 2002] Oard, D. (2002). When you come to a fork in the road, take it : Multiple futures for clir research. In *Cross-Language Information Retrieval : A Research Roadmap Workshop at SIGIR-2002*, Tampere Finland. ACM.

[Oard and Dorr, 1996] Oard, D. and Dorr, B. (1996). A survey of multilingual text retrieval. Technical report, University of Maryland.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). Bleu : a method for automatic evaluation of machine translation. In *ACL-02*, pages 311–318. University of Pennsylvania.

[Peters and Sheridan, 2001] Peters, C. and Sheridan, P. (2001). Accès multilingue aux systèmes d'information. In *67th IFLA Council and General Conference*.

[Poggio and Girosi, 1989] Poggio, T. and Girosi, F. (1989). A theory of networks for approximation and learning. Technical Report AIM-1140. Available from World Wide Web : http://citeseer.nj.nec.com/poggio89theory.html.

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

[Powell, 1987] Powell, M. (1987). *Algorithms for Approximation*, chapter Radial basis functions for multivariable interpolation : A review, pages 143–167. Clarendon Press, New York.

[Péry-Woodley, 1995] Péry-Woodley, M. P. (1995). Quels corpus pour quels traitements atomatiques ? *Traitement Automatique des Langues*, 36(1-2) :213–232.

[Quinlan, 1986] Quinlan, J. (1986). Induction of decision trees. *Machine Learning*, 1(1) :81–106.

[Quinlan, 1993] Quinlan, J. (1993). *C4.5 : Programs for Machine Learning*. San Mateo, CA.

[Rajman and Lebart, 1998] Rajman, M. and Lebart, L. (1998). Similarités pour données textuelles. In *4th International Conference on Statistical Analysis of Textual Data (JADT'98)*, pages 545–555, Nice, France.

[Robertson and Harding, 1984] Robertson, S. and Harding, P. (1984). Probabilistic automatic indexing by learning from human indexers. *Journal of Documentation*, 40(4) :264–270.

[Robertson and K.Sparck-Jones, 1976] Robertson, S. and K.Sparck-Jones (1976). Relevance weighting of search terms. In *JASIS*, number 27, pages 129–146.

[Robertson et al., 1996] Robertson, S., Walker, S., Hancock-Beaulieu, M., Gull, A., and Lau, M. (1996). Okapi at TREC. In *Text REtrieval Conference*, pages 21–30. Available from World Wide Web : http://trec.nist.gov/pubs/trec2/papers/ps/city.ps.gz.

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

[Sable and Hatzivassiloglou, 2000] Sable, C. L. and Hatzivassiloglou, V. (2000). Text-based approaches for non-topical image categorization. *International Journal of Digital Libraries*, 3(3) :261–275. Available from World Wide Web : http://www.cs.columbia.edu/~sable/research/ijodl00.pdf.

[Sahami, 1998] Sahami, M., editor (1998). *Proceedings of the 1998 Workshop on Learning for Text Categorization*, Madison, US. Available as Technical Report WS-98-05.

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

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

[Saporta, 1990] Saporta, G. (1990). *Probabilités, Analyse des données et Statistique*. Édition Technip, Paris, France.

[Savoy, 2002] Savoy, J. (2002). Report on clef-2002 experiements : Combining multiple sources of evidence. Results of the CLEF-2002, cross-language evaluation forum. University of Neuchatel.

[Schapire and Singer, 2000] Schapire, R. E. and Singer, Y. (2000). BoosTexter : a boosting-based system for text categorization. *Machine Learning*, 39(2/3) :135–168. Available from World Wide Web : http://www.research.att.com/~schapire/papers/SchapireSi98b.ps.Z.

[Schapire et al., 1998] Schapire, R. E., Singer, Y., and Singhal, A. (1998). Boosting and Rocchio applied to text filtering. In Croft, W. B., Moffat, A., van Rijsbergen, C. J., Wilkinson, R., and Zobel, J., editors, *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 215–223, Melbourne, AU. ACM Press, New York, US. Available from World Wide Web : http://www.research.att.com/~schapire/cgi-bin/uncompress-papers/SchapireSiSi98.ps.

[Schmid, 1994] Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, Manchester, UK.

[Schütze et al., 1995] Schütze, H., Hull, D. A., and Pedersen, J. O. (1995). A comparison of classifiers and document representations for the routing problem. In Fox, E. A., Ingwersen, P., and Fidel, R., editors, *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, pages 229–237, Seattle, US. ACM Press, New York, US. Available from World Wide Web : ftp://parcftp.xerox.com/pub/qca/papers/sigir95.ps.gz.

[Sebastiani, 1999] Sebastiani, F. (1999). A tutorial on automated text categorisation. In Amandi, A. and Zunino, R., editors, *Proceedings of ASAI-99, 1st Argentinian Symposium on Artificial Intelligence*, pages 7–35, Buenos Aires, AR. Available from World Wide Web : http://faure.iei.pi.cnr.it/~fabrizio/Publications/ASAI99.pdf. An extended version appears as [Sebastiani, 2002].

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

[Sebastiani et al., 2000] Sebastiani, F., Sperduti, A., and Valdambrini, N. (2000). An improved boosting algorithm and its application to automated text categorization. In Agah, A., Callan, J., and Rundensteiner, E., editors, *Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management*, pages 78–85, McLean, US. ACM Press, New York, US. Available from World Wide Web : http://faure.iei.pi.cnr.it/~fabrizio/Publications/CIKM00.pdf.

[Shannon, 1948] Shannon, C. (1948). The Mathematical Theory of Communication. *Bell System Technical Journal*, 27 :379–423 and 623–656.

[Shortliffe, 1976] Shortliffe, E. H. (1976). *Computer-Based Medical Consultation : MYCIN*. Elsevier North, Holland.

[Sibun and Reynar, 1996] Sibun, P. and Reynar, J. (1996). Language Identification : Examining the Issues. In *Symposium on Document Analysis and Information Retrieval*, pages 125–135, Las Vegas.

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

[Soergel, 1997] Soergel, D. (1997). Multilingual thesauri in cross-language text and speech retrieval. In *AAAI Symposium on Cross-Language*

Text and Speech Retrieval. American Association for Artificial Intelligence. Available from World Wide Web : http://citeseer.nj.nec.com/soergel97multilingual.html.

[Souter et al., 1994] Souter, C., Churcher, G., Hayes, P., Hughes, J., and Johnson, S. (1994). Natural Language Identification Using Corpus-Based Models. *Hermes Journal of Linguistics*, 13 :183–203.

[Stone, 1974] Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, B(36) :111–147.

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

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

[Thiel, 2001] Thiel, E. (2001). *Géométrie des distances de chanfrein*. Habilitation à diriger des recherches, Université de la Méditerranée, Aix-Marseille 2.

[Tzeras and Hartmann, 1993] Tzeras, K. and Hartmann, S. (1993). Automatic indexing based on Bayesian inference networks. In Korfhage, R., Rasmussen, E., and Willett, P., editors, *Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval*, pages 22–34, Pittsburgh, US. ACM Press, New York, US. Available from World Wide Web : http://www.darmstadt.gmd.de/~tzeras/FullPapers/gz/Tzeras-Hartmann-93.ps.gz.

[Uren, 2000] Uren, V. (2000). An evaluation of text categorisation errors. In *Proceedings of the One-day Workshop on Evaluation of Information Management Systems*, pages 79–87, London, UK. Queen Mary and Westfield College. Available from World Wide Web : http://kmi.open.ac.uk/people/victoria/eval200.pdf.

[Van Rijsbergen, 1979] Van Rijsbergen, C. J. (1979). *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow. Available from World Wide Web : http://citeseer.nj.nec.com/vanrijsbergen79information.html.

[Vapnik, 1995] Vapnik, V. (1995). *The Nature of Statistical Learning*. Springer.

[Vapnik, 1998] Vapnik, V. (1998). *Statistical Learning Theory*. John Wiley, NY.

[Viennet and Fernandez, 1999] Viennet, E. and Fernandez, R. (1999). Machines à vecteurs de support et réseaux de neurones : comparaisons expérimentales pour l'indentification de visages. In *Conférence sur l'Apprentissage CAP'99*, Palaiseau, France.

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

[Volk, 1997] Volk, M. (1997). Probing the lexicon in evaluating commercial mt systems. In *Proc. of ACL/EACL Joint Conference*, pages 112–119, Madrid.

[Volk, 1998] Volk, M. (1998). The Automatic Translation of Idioms. Machine Translation vs. Translation Memory Systems. *Machine Translation : Theory, Applications, and Evaluation. An assessment of the state of the art.*

[Weiss et al., 1999] Weiss, S. M., Apté, C., Damerau, F. J., Johnson, D. E., Oles, F. J., Goetz, T., and Hampp, T. (1999). Maximizing text-mining performance. *IEEE Intelligent Systems*, 14(4) :63–69. Available from World Wide Web : http://www.research.ibm.com/dar/papers/pdf/ieee99_mtmp.pdf.

[Wiener, 1995] Wiener, E. D. (1995). A neural network approach to topic spotting in text. Master's thesis, Department of Computer Science, University of Colorado at Boulder, Boulder, US. Available from World Wide Web : http://www.stern.nyu.edu/~aweigend/Research/Papers/TextCategorization/Wiener_Thesis95.ps.

[Wiener et al., 1995] Wiener, E. D., Pedersen, J. O., and Weigend, A. S. (1995). A neural network approach to topic spotting. In *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 317–332, Las Vegas, US. Available from World Wide Web : http://www.stern.nyu.edu/~aweigend/Research/Papers/TextCategorization/Wiener.Pedersen.Weigend_SDAIR95.ps.

[Yang, 1994] Yang, Y. (1994). Expert network : effective and efficient learning from human decisions in text categorisation and retrieval. In Croft, W. B. and van Rijsbergen, C. J., editors, *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 13–22, Dublin, IE. Springer Verlag, Heidelberg, DE. Available from World Wide Web : http://www.acm.org/pubs/articles/proceedings/ir/188490/p13-yang/p13-yang.pdf.

[Yang, 1999] Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1/2) :69–90. Available from World Wide Web : http://www.cs.cmu.edu/~yiming/papers.yy/irj99.ps.

[Yang and Chute, 1992] Yang, Y. and Chute, C. (1992). A Linear Least Squares Fit Method for Terminology Mapping. In *Proceedings of Fifteenth International Conference on Computational Linguistics (COLING'92)*, volume 2, pages 447–453.

[Yang and Chute, 1994] Yang, Y. and Chute, C. G. (1994). An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems*, 12(3) :252–277. Available from World Wide Web : http://www.acm.org/pubs/articles/journals/tois/1994-12-3/p252-yang/p252-yang.pdf.

[Yang and Liu, 1999] Yang, Y. and Liu, X. (1999). A re-examination of text categorization methods. In Hearst, M. A., Gey, F., and Tong, R., editors, *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pages 42–49, Berkeley, US. ACM Press, New York, US. Available from World Wide Web : http://www.cs.cmu.edu/~yiming/papers.yy/sigir99.ps.

[Yang and Pedersen, 1997] Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In Fisher, D. H., editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420, Nashville, US. Morgan Kaufmann Publishers, San Francisco, US. Available from World Wide Web : http://www.cs.cmu.edu/~yiming/papers.yy/ml97.ps.

[Zhou and Guan, 2002] Zhou, S. and Guan, J. (2002). Chinese documents classification based on N-grams. In Gelbukh, A. F., editor, *Proceedings of CICLING-02, 3rd International Conference on Computational Linguistics and Intelligent Text Processing*, pages 405–414, Mexico City, MX. Springer Verlag, Heidelberg, DE. Published in the “Lecture Notes in Computer Science” series, number 2276.

[Zighed, 1985] Zighed, D. A. (1985). *Méthodes et outils pour les processus d'intérrogation non arborescents.* PhD thesis, Université Claude Bernard, Lyon 1, Lyon, France.

[Zighed et al., 1992] Zighed, D. A., Auray, J., and Duru, G. (1992). *SIPINA : Méthode et logiciel.* Editions A. Lacassagne, Lyon, France.

[Zighed and Rakotomalala, 2000] Zighed, D. A. and Rakotomalala, R. (2000). *Graphes d'induction. Apprentissage et Data Mining.* Hermes Science Publication, Paris.

[Zighed and Rakotomalala, 2002] Zighed, D. A. and Rakotomalala, R. (2002). Extraction de connaissances à partir de données (ECD). In *Techniques de l'Ingénieur*, volume HA.