SIMON RÉHEL

CATÉGORISATION AUTOMATIQUE DE TEXTES ET COOCCURRENCE DE MOTS PROVENANT DE DOCUMENTS NON ÉTIQUETÉS

Mémoire présenté
à la Faculté des études supérieures de l'Université Laval
dans le cadre du programme de maîtrise en informatique
pour l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES ET DE GÉNIE
UNIVERSITÉ LAVAL
QUÉBEC

JANVIER 2005

© Simon Réhel, 2005

Résumé

Ayant pour objectif de rendre un programme informatique capable d'assigner de façon autonome des documents textuels à leur classe d'appartenance, la catégorisation automatique de textes est rendue possible grâce à l'apprentissage supervisé. Un entraînement du programme est effectué sur un ensemble de documents auxquels des étiquettes de catégorie ont déjà été assignées par des humains. Or, la constitution de cet ensemble d'entraînement se révèle un processus long et coûteux. Ce mémoire propose une façon d'améliorer la capacité d'un classificateur à bien accomplir sa tâche dans des situations où un entraînement sur un nombre suffisant de textes n'aura pas été possible. L'approche suggérée consiste à étudier une forme d'association, la cooccurrence, entre les mots provenant d'un ensemble de textes libellés et ceux provenant d'un ensemble de textes non libellés, plus volumineux. On espère ainsi augmenter à faible coût le vocabulaire utile à la classification de textes, en minimisant le nombre de documents à étiqueter.

Abstract

Automated text categorization consists of developing computer programs able to autonomously assign texts to predefined categories, on the basis of their content. Such applications are possible thanks to supervised learning, which implies a training phase on manually labeled documents. However, the construction of a training set is long and expensive. This study suggests a way to assist text classifiers in the gathering of the vocabulary when the size of the training set is limited. So, it is proposed to analyze word cooccurrence inside a text collection of many non-labeled documents, to augment the vocabulary produced by the analysis of the labeled texts. The representation of new documents to classify can then be modified in order to better match the vocabulary used by the classifier. What is expected, of course, is an improvement of its ability to categorize texts.

Avant-Propos

Poursuivre des études au niveau de la maîtrise, c'est l'opportunité d'enrichir ses connaissances dans un domaine particulier du savoir et de s'initier à la recherche. Mais, avant tout, c'est une occasion exceptionnelle de pouvoir consacrer plusieurs mois de travail à un projet personnel majeur, un projet sous sa propre responsabilité.

Pourtant, malgré cette dimension individuelle, mener à bien une telle entreprise ne se fait pas sans la collaboration de plusieurs personnes. Je profite donc de ces quelques lignes pour adresser mes remerciements à ceux qui, de près ou de loin, m'ont supporté tout au long de mes travaux.

En premier lieu, je tiens à dire merci à mon directeur de recherche, M. Guy Mineau. D'abord, merci de m'avoir fait découvrir le monde de la recherche et de m'avoir convaincu d'emprunter ce chemin. Évidemment, merci aussi pour l'encadrement exceptionnel, les suggestions, la confiance et la liberté accordés tout au long de mon cheminement.

Impossible aussi de passer sous silence l'aide apportée par M. Pascal Soucy, qui m'a facilité la tâche de la définition du projet de recherche et qui a été généreux en conseils jusqu'à la fin. Surtout, il m'a donné accès à des outils logiciels et à des références utiles, qui m'ont fait épargner beaucoup de temps et d'efforts.

Également, je remercie la compagnie Copernic.com pour les ressources techniques allouées ainsi que le Fonds québécois de la recherche sur la nature et les technologies (FQRNT) pour le soutien financier accordé.

Enfin, mes derniers remerciements, mais non les moindres, sont adressés à tous ceux qui, sans avoir été impliqués directement dans le projet, ont toujours été d'un grand support : mes parents, ma famille, mes amis, mes confrères de classe, mes professeurs, mes collègues de travail … qui tous se reconnaîtront je l'espère sans que j'aie besoin de les nommer !

Table des matières

Résumé i
Abstract ii

Avant-Propos iii

Table des matières v
Liste des tableaux vii
Liste des figures viii

Chapitre 1 Catégorisation automatique de textes 6
1.1 Catégorisation automatique de textes 7
1.2 Représentation des documents 12
1.3 Réduction de la taille du vocabulaire 16
1.3.1 Sélection d'attributs 17
1.3.2 Extraction d'attributs 22

Chapitre 2 Algorithmes d'apprentissage automatique appliqués à la catégorisation de textes 24
2.1 Algorithmes d'apprentissage 24
2.1.1 Classificateur bayésien naïf 26
2.1.2 Algorithme des k-voisins les plus proches 28
2.1.3 Machines à support vectoriel 32
2.1.4 Autres types de classificateurs 34
2.2 Critères d'évaluation des classificateurs 35

Chapitre 3 Utilisation de connaissances extérieures à l'ensemble d'entraînement 40
3.1 Coût d'étiquetage des documents d'entraînement 41
3.2 Utilisation de connaissances extérieures 45
3.2.1 Utilisation d'un lexique 45
3.2.2 Utilisation de documents non libellés 48

Chapitre 4 Cooccurrence de mots 57
4.1 Définition 58
4.2 Mesures d'association 60
4.2.1 Test du χ^2 62
4.2.2 Ratio de vraisemblance 64
4.2.3 Information mutuelle 67

Chapitre 5 Étude de cooccurrence de mots dans des textes non libellés pour assister la catégorisation automatique de textes 71
5.1 Problématique 72
5.2 Description de l'approche proposée 78

Chapitre 6 Évaluation de l'approche proposée 88
6.1 Méthodologie 89
6.1.1 Corpus de textes 89
6.1.2 Caractéristiques des classificateurs utilisés 94
6.2 Résultats 96
6.3 Discussion 99

Conclusion 106

Bibliographie 112

Annexe A 116
Annexe B 118

Liste des tableaux

Tableau 2.1 - Table de contingence à la base de l'évaluation des classificateurs 36

Tableau 4.1 - Table de contingence pour mesurer le degré d'association entre deux mots ..62

Tableau 5.1 - Influence de la taille de l'ensemble d'entraînement sur le vocabulaire reconnu par un classificateur 76

Tableau 5.2 - Influence de la taille de l'ensemble d'entraînement sur le nombre de mots inconnus du classificateur survenant dans des documents à classer.....77

Tableau 5.3 - Réduction du nombre de mots inconnus à traiter en utilisant un seuil minimal d'occurrence.....84

Tableau 6.1 - Résumé des renseignements sur les corpus de textes utilisés.....93

Tableau 6.2 - Exemples de paires de mots jugés cooccurrents.....97

Tableau 6.3 - Évaluation de la méthode proposée sur le corpus RCV1 98

Tableau 6.4 - Évaluation de la méthode proposée sur le corpus WebKB 99

Tableau 6.5 - Évaluation de la méthode proposée sur le corpus Ohsumed 99

Tableau 6.6 - Comportement de la précision et du rappel sur le corpus RCV1 et sur un classificateur kNN.....103

Tableau 6.7 - Comparaison de notre approche avec deux autres travaux tirant profit de documents non étiquetés, sur le corpus WebKB 104

Liste des figures

Figure 1.1 - La tâche de classification 7

Figure 1.2 - Entraînement d'un système de classification automatique de textes 11

Figure 1.3 - Classification d'un nouveau document 12

Figure 2.1 - Maximisation de la marge avec les SVM 32

Figure 3.1 - Algorithme EM 51

Figure 3.2 - Concepts latents 53

Figure 3.3 - Indexation sémantique latente et catégorisation automatique de textes 54

Figure 3.4 - Utilisation de documents non libellés avec l'algorithme kNN 55

Figure 5.1 - Influence de la taille du corpus d'entraînement sur les performances d'un classificateur (corpus RCV1) 73

Figure 5.2 - Influence de la taille du corpus d'entraînement sur les performances d'un classificateur (corpus WebKB) 73

Figure 5.3 - Influence de la taille du corpus d'entraînement sur les performances d'un classificateur (corpus Ohsumed) 74

Figure 5.4 - Approche proposée 80

Figure 5.5 - Détermination des mots impliqués dans l'analyse de cooccurrence 85

Introduction

Jour après jour, la masse d'information qui nous est disponible augmente. L'Internet s'enrichit continuellement de nouveau contenu, les entreprises emmagasinent de plus en plus dans leurs bases de données, le courriel devient un moyen de communication extrêmement populaire et on rend disponible en format électronique des documents autrefois manuscrits. Mais toute cette information serait sans intérêt si notre capacité à y accéder efficacement n'augmentait pas elle aussi. Pour en profiter au maximum, on a besoin d'outils nous permettant de chercher, classer, conserver, mettre à jour et analyser les données accessibles. Il nous faut des outils nous aidant à trouver dans un temps raisonnable l'information désirée, effectuant certaines tâches à notre place, ou, à tout le moins, nous facilitant le travail. Le forage de données, les moteurs de recherche, l'indexation automatique et les systèmes de gestion de bases de données sont des exemples parmi tant d'autres de techniques mises au point à cette fin. Et la recherche est toujours active pour tenter d'apporter des améliorations aux solutions existantes.

Un de ces domaines prometteurs est la catégorisation[1] automatique de textes. Imaginons-nous en présence d'une banque considérable de textes, qui seraient plus aisément accessibles s'ils étaient répartis dans un ensemble de catégories en fonction de leur sujet. Évidemment, on pourrait demander à un humain de lire tous les textes et de les classer manuellement. Mais la tâche s'avère colossale s'il fait face à des centaines, voire des milliers de documents. Il apparaît alors très intéressant de pouvoir compter sur une application informatique qui, de façon automatique, assignerait ces textes à un ensemble prédéfini de catégories. C'est

[1] Dans la littérature, on ne fait en général pas de distinction entre «*catégorisation*» et «*classification*». Par conséquent, dans la suite de ce document, les deux termes seront employés indifféremment.

précisément là le but de la classification automatique de textes. On peut facilement concevoir l'utilité d'un tel processus et ses applications multiples. Par exemple, l'organisation des documents internes d'une compagnie, de documents tels les annonces classées qu'un journal a à gérer, les courriels reçus par un individu, les demandes de brevet acheminées à un office international. Pensons aussi au filtrage d'articles provenant d'une agence de presse ou à la détection de courriel indésirable. La classification automatique de textes pourrait aussi faciliter la recherche de pages Web par leur triage dans une hiérarchie de style Yahoo! Et la liste ne s'arrête pas ici, loin de là.

Les approches actuelles visant à créer des classificateurs automatiques de textes sont directement liées à l'apprentissage automatique, ce sous-domaine de l'intelligence artificielle qui s'intéresse à conférer aux machines la capacité de s'améliorer à l'accomplissement d'une tâche, en interagissant avec leur environnement. L'apprentissage automatique se divise principalement en deux façons d'apprendre : l'apprentissage supervisé et l'apprentissage non supervisé. C'est dans l'approche dite supervisée que s'inscrit la façon dont on aborde aujourd'hui le problème de la catégorisation automatique de textes. Dans ce paradigme, l'apprentissage s'effectue à partir d'un ensemble d'exemples où chacun d'eux est constitué d'un objet d'entrée et d'une valeur de sortie désirée pour cet objet. En connaissant les sorties prévues, l'algorithme peut généraliser les exemples afin d'identifier les différents attributs des objets qui justifient une sortie particulière, pour devenir en mesure de traiter de nouvelles données. Ce processus correspond exactement à ce qui est souhaité en catégorisation de textes : un apprentissage sur une banque de documents dont la classe d'appartenance est connue du classificateur (ayant été déterminée au préalable par un humain). Évidemment, une implication majeure en découle, soit le fait qu'un corpus de

textes d'entraînement libellés doit être constitué. Des ressources doivent être allouées à cette tâche qui, pour un corpus de grande taille, peut requérir un temps considérable.

La problématique que ce mémoire se propose d'affronter se situe à ce niveau. Sans remettre complètement en question la nécessité d'avoir en main un corpus d'entraînement, coûteux à bâtir, il apparaît intéressant de permettre au classificateur de pouvoir profiter d'une autre source de connaissances. Plutôt que de compter uniquement sur des documents libellés, pourquoi ne pas en plus baser son travail sur une grande banque de textes non libellés ? Selon le contexte, si l'accès à un corpus d'entraînement de taille raisonnable est déjà possible, ces nouvelles connaissances permettraient éventuellement d'augmenter le taux de réussite du classificateur. Mais, surtout, dans le cas où les circonstances restreignent le nombre de documents libellés disponibles, les nouvelles connaissances pourraient permettre de contrebalancer ce manque et d'atteindre un taux de réussite plus acceptable, sans requérir l'augmentation de la taille du corpus d'entraînement. Suite à cette proposition, il reste à déterminer le type de connaissances à extraire des textes non libellés. L'avenue qui a été explorée dans le cadre de nos travaux est celle de la cooccurrence entre les mots. Des termes cooccurrents peuvent être associés, et le pouvoir discriminant de l'un peut être inféré à partir du pouvoir discriminant de l'autre.

Diverses motivations poussent à travailler dans la direction qui vient d'être exposée. Parmi les principales, il y a entre autres la prise de conscience du coût élevé de constitution de corpus de textes étiquetés, qui sont nécessaires pour entraîner des classificateurs. Il est vrai que contrairement à ce qui se passe dans d'autres domaines, les chercheurs en catégorisation automatique de textes peuvent tout de même compter sur quelques collections de textes classifiés d'assez grande

envergure. Cependant, il ne faut pas oublier qu'en dehors du monde de la recherche, lors du développement d'une application réelle de catégorisation, un corpus plus spécifique au domaine en question doit être disponible. La nécessité de l'étiquetage manuel et les coûts associés sont donc réels. D'autre part, l'intérêt potentiel de tirer profit de documents non libellés vient de leur grande accessibilité et de leur disponibilité en grande quantité. Les résultats positifs de certains travaux en ce sens sont encourageants. En ce qui a trait à la cooccurrence de mots, il s'agit d'une association entre les mots dont la découverte peut être faite facilement à partir d'un ensemble de textes, de façon statistique. C'est une idée qui s'apparente donc bien avec le désir de tirer profit de documents non libellés. Par ailleurs, la connaissance sur la cooccurrence entre les mots semble avoir, au premier abord, le potentiel d'apporter un plus à un classificateur de textes. La cooccurrence reflète en effet la tendance qu'ont les mots à apparaître dans un même contexte, donc dans les mêmes textes, dans les mêmes catégories.

Bref, ce qui est proposé dans ce mémoire est le développement d'une méthode qui, pour faire de la catégorisation automatique de textes, mettrait aussi à contribution des documents non classés, disponibles en grande quantité, pour améliorer la précision des classificateurs actuels ou pour diminuer le nombre de documents libellés requis. Plus précisément, c'est la cooccurrence de mots qui est étudiée dans les textes non classés pour récupérer de l'information sur les relations qu'entretiennent les mots du lexique entre eux. Les objectifs du projet sont donc de trouver comment extraire de l'information utile et comment incorporer cette information dans le processus d'apprentissage et de classification.

Pour atteindre ces objectifs, la méthodologie privilégiée a été la suivante. D'abord, une revue bibliographique du domaine a permis de constater l'état actuel des choses en ce qui concerne la catégorisation automatique de textes, l'utilisation de

connaissances extérieures à l'ensemble d'entraînement et la cooccurrence de mots. Un travail d'analyse et de réflexion a donné naissance à une façon de combiner ces deux dernières notions qui serait profitable à un classificateur. Puis, des expérimentations ont été menées sur différents corpus de textes pour valider la pertinence de cette nouvelle approche et son utilité réelle.

Essentiellement, voici comment se structure la suite du mémoire. En premier lieu, le chapitre 1 vise à présenter différentes facettes de la catégorisation automatique de textes. En plus de préciser la nature du problème, il traite des principaux modes de représentation des documents utilisés et des principales techniques appliquées pour réduire la taille du vocabulaire pris en compte. Puis, le chapitre 2 expose les principaux algorithmes d'apprentissage ayant fait leurs preuves dans le domaine de la catégorisation de textes et les façons de les comparer entre eux. Pour poursuivre cet état de l'art, le chapitre 3 traite plus spécifiquement du problème de la constitution de corpus de textes libellés, du coût qui est associé à cette tâche et des façons qui ont été déjà proposées pour exploiter des connaissances extérieures à ces documents libellés. Au chapitre 4, il est plutôt question de la notion de cooccurrence entre des mots : en quoi cela consiste et comment on peut l'évaluer numériquement. Enfin, le chapitre 5 présente de façon détaillée l'approche nouvelle que nous proposons, tandis que le chapitre 6 expose la méthodologie et les résultats des expérimentations qui ont été menées pour valider cette approche, suivis d'une discussion à leur sujet.

Chapitre 1

Catégorisation automatique de textes

L'objet d'étude principal de ce mémoire, la catégorisation automatique de textes, est un problème qui intéresse les chercheurs depuis relativement longtemps. On retrouve des travaux portant sur ce sujet depuis au moins le début des années 1960. Même s'il est certain que des avancées importantes ont été observées depuis, la recherche dans ce domaine est toujours très pertinente, car les résultats obtenus aujourd'hui sont encore sujets à amélioration. Pour certaines tâches, les classificateurs automatiques performent presque aussi bien que les humains, mais pour d'autres, l'écart est encore grand. Sans compter que le besoin d'un traitement efficace de l'information est grandissant, car l'immense bassin de connaissances à notre portée est sans intérêt s'il n'est pas bien géré. Au premier abord, l'essentiel du problème est facile à saisir. D'un côté, on est en présence d'une banque de documents textuels et de l'autre, d'un ensemble prédéfini de catégories. L'objectif est de rendre une application informatique capable de déterminer de façon autonome dans quelle catégorie classer chacun des textes, à partir de leur contenu, tel qu'illustré à la figure 1.1. Malgré cette première définition très simple, la solution n'est pas immédiate et plusieurs facteurs sont à prendre en considération. Ce chapitre vise donc à définir plus en profondeur de quoi il en retourne. En premier lieu, il précisera la tâche de classification, les principales difficultés qu'elle

engendre et les principales façons d'aborder le problème. Le deuxième point qui sera présenté est le choix d'un mode de représentation adéquat des instances à traiter, en l'occurrence les textes. Il s'agit d'une étape incontournable en apprentissage automatique : on doit opter pour une façon uniforme et judicieuse d'abstraire les données avant de les soumettre à un algorithme. Comme l'apprentissage joue un rôle dans la catégorisation automatique de textes, le choix d'un mode de représentation y devient un enjeu. Par la suite, il sera question de la sélection d'attributs, presque toujours impliquée en catégorisation automatique de textes et par laquelle on élimine les attributs jugés inutiles à la classification.

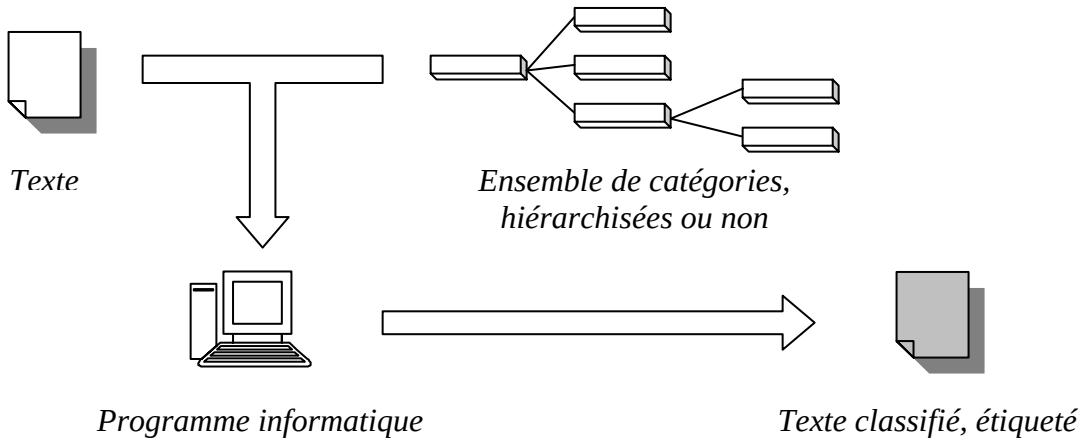

Figure 1.1 - La tâche de classification

1.1 Catégorisation automatique de textes

Comme il vient d'être mentionné, le but de la catégorisation automatique de textes est d'apprendre à une machine à classer un texte dans la bonne catégorie en se basant sur son contenu. Habituellement, les catégories font référence aux sujets des textes, mais pour des applications particulières, elles peuvent prendre d'autres

formes. En effet, on peut résoudre, par des techniques de catégorisation, des problèmes tels que l'identification de la langue d'un document, le filtrage de courrier électronique pertinent ou indésirable, ou encore la désambiguïsation de termes [Seb99]. Un autre aspect du problème qui varie selon les applications est la présence ou non d'une contrainte concernant le nombre de catégories assignables à un document donné. Il se peut qu'on désire qu'un même texte ne soit associé qu'à une seule catégorie ou bien on peut permettre que plusieurs catégories accueillent un même document. Aussi, une précision supplémentaire est à faire : dans le cadre de la catégorisation de textes, l'ensemble de catégories possibles est déterminé à l'avance. Il est à noter que le problème consistant à regrouper des documents selon leur similarité, mais lorsque les groupes à former sont a priori inconnus, en est un à part entière connu sous le nom de regroupement (*«clustering»)* de textes. Il n'en sera pas question ici.

Arrêtons-nous au cas où les catégories font référence aux sujets des textes : la classification s'apparente alors au problème de l'extraction de la sémantique d'un texte, puisque l'appartenance d'un document à une catégorie est étroitement liée à la signification de ce texte. C'est en partie ce qui rend la tâche difficile puisque le traitement de la sémantique d'un document écrit en langage naturel n'est pas encore solutionné. Une observation mérite aussi d'être faite concernant le fait que la nature des textes à traiter influence significativement la difficulté de la tâche de classification. Prenons l'exemple d'articles de journaux écrits généralement dans un style direct et contenant de l'information purement factuelle. Le vocabulaire utilisé s'avère précis et souvent relativement restreint pour faciliter la compréhension. À l'opposé, imaginons un corpus de textes d'un style plus littéraire, utilisant un vocabulaire très varié et imagé. On peut aisément prévoir que la classification automatique de ce dernier corpus présentera plus de

difficultés que pour l'autre [SM99]. Entre ces deux extrêmes, on peut aussi retrouver des textes scientifiques (où chaque catégorie aura potentiellement un vocabulaire caractéristique), des pages Web, du courrier électronique, etc. Chacun de ces types de textes possède des particularités qui rendent la tâche de classification plus ou moins ardue.

De façon générale, on distingue deux façons d'aborder le problème de la classification automatique [Seb99]. Jusqu'à la fin des années 1980, l'approche dominante pour le résoudre s'inscrivait dans une optique d'ingénierie des connaissances («*knowledge engineering*»). On construisait un système expert comportant un ensemble de règles définies manuellement, par des experts du domaine, et qui ensuite pouvait procéder automatiquement à la classification. Ces règles prenaient généralement la forme d'une implication logique où l'antécédent portait, typiquement, sur la présence ou l'absence de certains mots, et où le conséquent désignait la catégorie d'appartenance du texte. Cette approche peut s'avérer très efficace. Entre autres, le système CONSTRUE a démontré une précision de classification impressionnante, mais sur un corpus de textes relativement petit, ce qui ne permet pas de tirer des conclusions favorables [Yan99]. En fait, malgré de bons résultats sur des banques de textes bien précises, l'inconvénient de cette approche est que l'édition des règles de décision peut s'avérer très longue. Et surtout, si des catégories s'ajoutent ou si on désire utiliser le classificateur dans un autre domaine, on doit répéter l'exercice. C'est donc la pertinence de cet ensemble de règles, qui évolue dans le temps, qui mine l'intérêt envers cette façon de faire.

Avec le développement de techniques d'apprentissage automatique («*machine learning*»), on voit le problème d'un autre oeil. En fait, même en 1961, [Mar61] avait proposé un classificateur bayésien qui se distinguait de l'édition de règles en se

basant plutôt sur un calcul de probabilités. Cependant, ce n'est pas avant le début des années 1990 que cette approche a pris son envol.

Dans l'esprit de l'apprentissage automatique, on vise à construire des systèmes qui vont apprendre par eux-mêmes à classer les documents. On met l'accent sur l'automatisation, par apprentissage, de la création du classificateur. À l'aide d'un ensemble de textes déjà associés à des catégories, on entraîne le système. Cette banque de textes libellés doit être préalablement construite par un humain. Ensuite, la machine tente d'apprendre la tâche de classification en observant le travail fait par l'humain. Elle essaie de généraliser les liens entre les textes et les catégories en analysant des exemples. Après la phase d'entraînement, le classificateur peut procéder lui-même au classement de nouveaux textes. Souvent, on s'inspire de la stratégie bien connue «*diviser-pour-régner*» et on découpe le problème en plusieurs sous-problèmes plus faciles à résoudre. Plus précisément, on construit un classificateur binaire pour chaque catégorie, qui va déterminer si, oui ou non, un document en fait partie. En fait, l'apprentissage de la distinction de deux catégories s'avère plus facile que l'apprentissage sur l'ensemble des catégories. De plus, comme les catégories peuvent éventuellement se chevaucher, le document soumis à plusieurs classificateurs sera jugé par chacun d'eux et pourra être associé à plus d'une catégorie. En résumé, l'objectif global devient donc de créer un constructeur automatique de classificateurs. À prime abord, la tâche est plus complexe qu'avec la première approche, où elle relève d'un expert chargé de produire les règles de classification. Cependant, en bout de ligne, lorsqu'un tel constructeur de classificateurs est mis au point, il peut être transposé d'un corpus de textes à un autre, d'un domaine à un autre, sans nécessiter beaucoup de travail supplémentaire, mis à part peut-être l'ajustement de quelques paramètres. L'effort en recherche des dernières années semble avoir été mis plutôt de ce côté. La figure

1.2 présente le processus général d'entraînement d'un tel système, tandis que la figure 1.3 illustre le processus de classification d'un nouveau document. Les étapes de représentation des documents et de sélection d'attributs, préalables à l'algorithme d'apprentissage en tant que tel, seront expliquées dans les sections qui suivent.

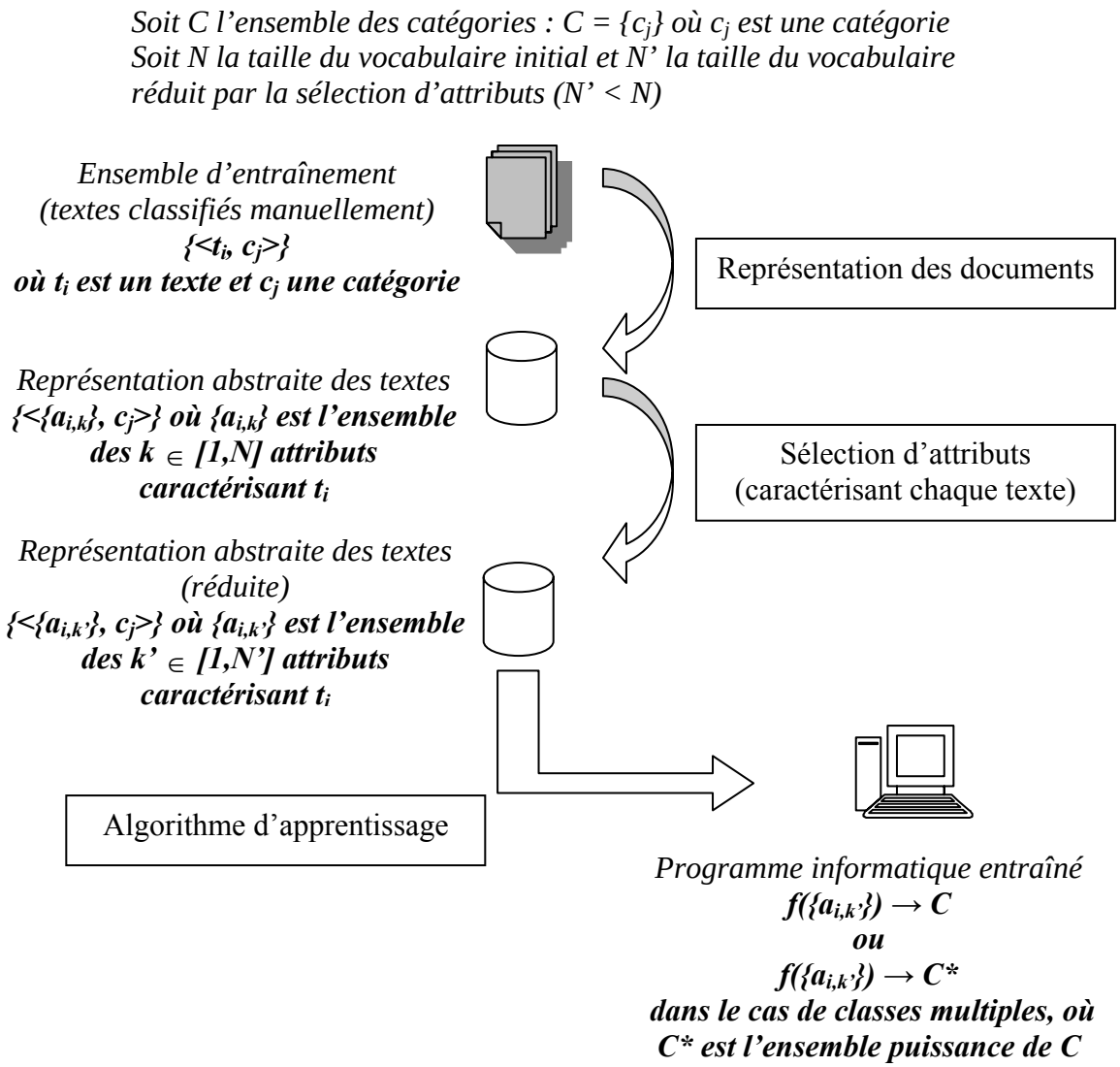

Figure 1.2 - Entraînement d'un système de classification automatique de textes

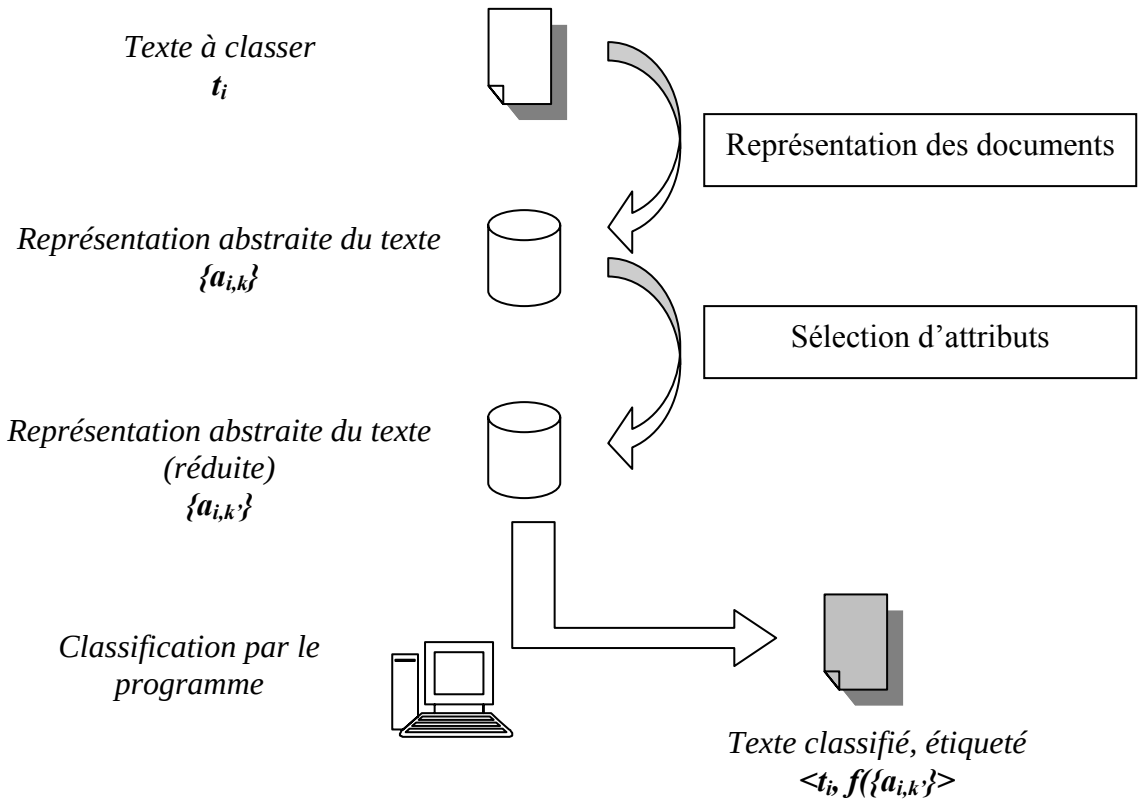

Figure 1.3 - Classification d'un nouveau document

1.2 Représentation des documents

À chaque fois qu'il est question de définir un problème de façon à assurer un traitement automatique, il est impossible de passer outre l'étape où il faut choisir la façon dont on va représenter le problème. Dans le cas de la classification automatique de textes, on doit opter pour une façon efficace de représenter les instances à traiter, soit les textes. Un grand nombre de chercheurs dans le domaine ont choisi d'utiliser une représentation vectorielle dans laquelle chaque texte est représenté par un vecteur de n termes pondérés. À la base, les n termes sont tout simplement les n différents mots apparaissant dans les textes de l'ensemble d'entraînement. Cette approche est aussi appelée «*bag-of-words*». On peut utiliser

d'autres types d'attributs pour caractériser les vecteurs dont certains seront présentés plus loin. Il existe aussi plusieurs façons d'associer un poids à un terme. Il peut être tout simplement binaire (1 si le mot est présent dans le texte, 0 sinon). Il peut aussi représenter le nombre d'occurrences du mot dans le texte. Cependant, en procédant ainsi, on donne une importance trop grande aux termes qui apparaissent très souvent à travers toutes les classes et qui sont peu représentatifs d'une classe en particulier.

Une façon largement utilisée de calculer le poids d'un terme est la fonction TFIDF (acronyme pour «*term frequency inverse document frequency*»). Issue du monde de la recherche d'information [SB88], celle-ci donne plus d'importance aux mots qui apparaissent souvent à l'intérieur d'un même texte, ce qui correspond bien à l'idée intuitive que ces mots sont plus représentatifs du document. Mais sa particularité est qu'elle donne également moins de poids aux mots qui appartiennent à plusieurs documents, pour refléter le fait que ces mots ont un faible pouvoir de discrimination entre les classes.

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

$$\mathrm{tfidf}(t_k, d_j) = \#(t_k, d_j) \cdot \log \frac{|Tr|}{\#(\mathrm{t_k})}$$

où

- $\#(t_k, d_j)$ est le nombre d'occurrences de t_k dans d_j
- $|Tr|$ est le nombre de documents d'entraînements
- $\#(t_k)$ est le nombre de documents d'entraînements dans lesquels t_k apparaît au moins une fois

Si on désire avoir des poids entre 0 et 1, on peut appliquer une normalisation, ce qui est souvent le cas. La fonction TFIDF a démontré une bonne efficacité dans des tâches de catégorisation de textes, et, en plus, son calcul est simple [Seb99]. Il faut savoir que d'autres méthodes plus sophistiquées peuvent aussi être utilisées [FGLS99].

Malgré la grande popularité de l'approche «*bag-of-words*» pour représenter les textes, bon nombre de chercheurs ont expérimenté l'utilisation de modes de représentation alternatifs. Une des motivations est que cette approche met de côté une quantité considérable d'information contenue dans le document original. Les paragraphes, les phrases et l'ordre des mots sont complètement écartés. Toute l'information sur les structures syntaxiques est perdue. La question est bien sûr de savoir si la qualité de la classification automatique des textes va être affectée négativement par cette façon d'agir. Est-ce que tout ce qui est mis à l'écart au profit d'une représentation «*bag-of-words*» est véritablement pertinent lors de l'assignation d'un texte à une catégorie ? La question à savoir si les mots pris séparément représentent toujours des unités sémantiques atomiques n'est plus à débattre. Il est clair que ce n'est pas le cas. Personne ne peut contredire le fait que les mots «*apprentissage*» et «*automatique*» ensemble ont une sémantique différente que lorsque pris séparément. On peut donc oser croire que des modes de représentation conservant l'information sur les liens entre les mots pourraient améliorer la classification de textes, sachant que cette tâche est directement liée au sens du message véhiculé par le texte, c'est-à-dire à sa sémantique.

En particulier, [SM99] ont essayé une dizaine de façons de représenter les textes sur un classificateur ayant comme base l'apprentissage de règles (RIPPER). Ils ont d'abord considéré les groupes nominaux, en extrayant des suites de noms et d'adjectifs pour construire les termes de l'espace vectoriel. L'utilisation d'une

application analysant la nature grammaticale des mots du texte a été nécessaire à la mise en place de ce mode de représentation. Puis, en utilisant une approche plutôt statistique que syntaxique, ils ont également considéré les groupes-clés, c'est-à-dire les groupes de mots ayant le plus de sens dans un document. En parallèle, ils ont aussi évalué l'impact de regrouper les mots synonymes en un même méta-attribut. Dans le même ordre d'idées, la notion d'hyperonymes[2] a été mise à l'épreuve pour regrouper des mots de la même façon. Malheureusement, aucun de ces modes alternatifs n'a produit de résultats équivalents ou supérieurs à l'approche «*bag-of-words*». Par contre, une partie de leurs expérimentations a porté sur l'utilisation de plusieurs classificateurs utilisant des modes de représentation différents, combinés à l'aide d'un vote à majorité. De cette façon, la classification s'est avérée meilleure, ce qui ouvre la porte à des investigations supplémentaires de ce côté.

Pour sa part, [Lew92] n'a pas non plus obtenu de résultats satisfaisants en représentant les textes à l'aide de groupes nominaux. L'explication trouvée à ces résultats décevants est que malgré la qualité sémantique supérieure des groupes nominaux par rapport aux mots individuels, ils possèdent une qualité statistique inférieure. L'approche «*bag-of-words*» est capable d'extraire des documents d'entraînement une information statistique sur les relations mot/catégorie qui s'avère très utile pour la classification et qui semble peser plus lourdement dans la balance que l'information sémantique fournie par les modes de représentation alternatifs. On peut en effet comprendre qu'un mot individuel est, en moyenne, amené à apparaître plus fréquemment dans un corpus de textes qu'un groupe nominal, fournissant ainsi au classificateur une meilleure image de la distribution statistique caractérisant le corpus.

[2] Un hyperonyme est un terme dont le sens inclut le sens d'autres termes, dits ses hyponymes. Par exemple, un chat est un animal donc «*animal*» est un hyperonyme de «*chat*».

1.3 Réduction de la taille du vocabulaire

S'attaquer au problème de la classification automatique de textes signifie aussi s'attaquer à des difficultés qui sont propres au traitement automatique de la langue naturelle. La taille impressionnante du vocabulaire peut s'avérer un obstacle à l'utilisation d'algorithmes plus complexes. Pourquoi ? Si l'on utilise directement le vocabulaire contenu dans les textes d'entraînement et que l'on crée un attribut pour chaque mot qu'il contient, on se retrouve avec un espace vectoriel ayant une dimension très élevée. Chacun des textes sera représenté par un vecteur ayant autant de termes qu'il y a de mots dans le vocabulaire. Le traitement d'un tel espace vectoriel demanderait beaucoup de mémoire et de temps de calcul et pourrait nous empêcher d'utiliser des algorithmes de classification plus complexes. Le problème pourrait dans certains cas devenir non soluble. Utiliser tous ces mots influencerait aussi négativement la précision de la classification. En effet, plusieurs mots, tant en anglais qu'en français, sont vides de sens, donc inutiles pour faire ressortir la sémantique d'un texte. Aussi, si un mot est présent dans un nombre élevé de documents, c'est donc qu'il ne permettra pas de départager l'appartenance d'un texte qui le contient à l'une ou l'autre des catégories. On dit alors que son pouvoir de discrimination est faible. C'est en bonne partie pour ces raisons que certaines techniques ont été mises en place pour réduire la dimension du vocabulaire, des techniques qui se divisent en deux grandes familles. La sélection d'attributs («*feature selection*») prend les attributs (ou mots) d'origine et conserve seulement ceux jugés utiles à la classification, selon une certaine fonction d'évaluation. Les autres sont rejetés. D'autres méthodes effectuent plutôt une extraction d'attributs («*feature extraction*»). À partir des attributs de départ, elles créent de nouveaux attributs, en faisant soit des regroupements ou des

transformations. On constate que la sélection d'attributs est meilleure quant à l'élimination d'attributs réellement inutiles ou même d'attributs erronés («*noisy*») (mots mal orthographiés par exemple) tandis que l'extraction d'attributs est plutôt axée sur la réduction du nombre d'attributs redondants.

1.3.1 Sélection d'attributs

Plusieurs techniques de sélection d'attributs ont été développées en vue de réduire la dimension de l'espace vectoriel. Chacune de ces techniques utilise des critères lui permettant de rejeter les attributs jugés inutiles à la tâche de classification. On obtient alors un vocabulaire réduit, des textes représentés par des vecteurs de moindre dimension, un temps de calcul plus abordable et même dans certains cas une précision de classification accrue.

Une bonne partie des chercheurs débutent le processus de sélection d'attributs par la suppression de mots vides de sens, communément appelés en anglais «*stop words*». Cette méthode nécessite l'utilisation d'une liste de ces mots qui ne contiennent aucune information sémantique, qui ne modifient pas le sens des mots qui les accompagnent. Par exemple, en français, on peut compter des mots comme «*le*», «*pour*», «*ne*». Il s'agit souvent de mots fonctionnels comme les prépositions. Évidemment, le contenu de cette liste dépend de la langue et possiblement du domaine des textes à classer. Mais cette étape, habituellement préalable à d'autres méthodes de sélection d'attributs, permet d'éliminer efficacement plusieurs mots inutiles à la tâche de classification. Cette pratique n'est toutefois pas universelle. Certains tendent à dire que ces mots peuvent contenir une certaine information sur la catégorie d'un texte. Par exemple, une catégorie donnée peut être écrite dans un langage faisant usage d'un grand nombre de mots de liaison, alors qu'une autre

peut être écrite dans un style plus direct n'en utilisant presque pas. Une autre illustration de ce phénomène : au sein d'une tâche de classification de pages Web, la présence d'adjectifs possessifs («*mon*», «*ma*», «*mes*», …) peut potentiellement aider à différencier les pages personnelles d'autres pages [NMTM00]. Les éliminer complètement empêcherait de tirer profit de cette information. Ceux qui s'opposent à l'utilisation d'une liste de mots vides de sens disent qu'en utilisant une méthode efficace de sélection d'attributs, les mots réellement inutiles sont de toute façon éliminés [Sou02].

La recherche de radical (*«stemming»)* et la lemmatisation sont d'autres processus qu'utilisent certains chercheurs pour créer un vocabulaire réduit. Leur but est de regrouper en un seul attribut les multiples formes morphologiques de mots qui ont une sémantique commune. Par exemple, «*étude*», «*étudiant*» et «*étudier*» pourraient être regroupés, et un seul attribut serait ajouté à l'espace vectoriel plutôt que trois. Ce faisant, la dimension du vocabulaire s'en trouverait réduite et on aurait capté la sémantique commune d'une famille de mots. Il existe plusieurs façons d'effectuer un tel traitement, certaines très simples, mais probablement moins efficaces, et d'autres plus sophistiquées. On peut penser à éliminer les n derniers caractères de tous les mots, mais dans ce cas, il faut déterminer n de façon très judicieuse. Des algorithmes existent aussi pour enlever les suffixes des mots pour ne conserver que la partie racine, en s'aidant de règles et de listes d'exceptions. Les algorithmes de «*stemming*» les plus connus sont sans doute ceux de Lovins [Lov68] et de Porter [Por80]. Tous ne s'entendent pas sur la distinction à faire entre les notions de lemmatisation et de «*stemming*», mais en général, la lemmatisation est associée à une approche plus complexe au problème qui implique d'abord l'étiquetage lexical des mots («*part-of-speech tagging*») pour différencier les différentes catégories syntaxiques, distinction à laquelle le «*stemming*» ne s'attarde pas. L'usage du

«stemming» et de la lemmatisation en catégorisation de textes est mitigé, moins répandu que la suppression de mots fonctionnels. Certains s'en méfient, probablement justement parce que leur efficacité varie beaucoup selon l'algorithme utilisé [Sou02].

Les deux processus précédents sont rarement utilisés seuls. Généralement, on les considère comme des traitements préalables à des techniques plus élaborées. Parmi ces critères de sélection d'attributs les plus rencontrés, notons [YP97] :

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

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

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

où

- Pr signifie la proportion des documents ayant la caractéristique entre parenthèses
- c_i signifie qu'un document fait partie de la catégorie c_i
- t signifie qu'un document possède le terme t
- $\bar{t}$ signifie qu'un document ne possède pas le terme t
- m est le nombre de catégories

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

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

- **La force du terme** (*«term strength»*) : Il s'agit d'une méthode plutôt différente des autres. Elle se propose d'estimer l'importance d'un terme en fonction de sa propension à apparaître dans des documents semblables. Une première étape consiste à former des paires de documents dont la similarité cosinusoïdale[3] est supérieure à un certain seuil. La force d'un terme est ensuite calculée à l'aide de la probabilité conditionnelle qu'il apparaisse dans le deuxième document d'une paire, sachant qu'il apparaît dans le premier.

Il faut remarquer qu'un point commun à toutes ces mesures est la nécessité de choisir un seuil. Peu importe le critère choisi pour la sélection, il faut déterminer à partir de quelle valeur on élimine ou on conserve un terme. Le choix de ce seuil est vraisemblablement aussi très déterminant pour la performance du classificateur. Une étude comparative a été menée [YP97] à propos de ces cinq méthodes de sélection des attributs, sur deux classificateurs différents[4]. Les résultats démontrent la supériorité du gain d'information et de la statistique du χ^2 sur les autres. Étonnamment, la performance de la sélection sur la base de la fréquence s'approche de la leur, malgré sa simplicité. Elle pourrait donc être utilisée

[3] Voir la section 2.1.2 pour le calcul de cette mesure de similarité.
[4] kNN et LLSF. Voir la section 2.1 pour le fonctionnement de ces classificateurs.

avantageusement lorsque le temps de calcul importe vraiment. Seulement, quand le niveau d'agressivité de la sélection augmente, elle tend à réussir un peu moins bien que le gain d'information et la statistique du χ^2. Mentionnons que bien d'autres idées ont été lancées, entre autres par [BGMM02] qui ont pour leur part proposé une méthode de sélection des attributs basée sur les classificateurs linéaires.

Un avantage supplémentaire à la réduction du vocabulaire est qu'elle réduirait le risque de sur-apprentissage («*overfitting*»). Il s'agit d'un phénomène qui se produit lorsqu'un classificateur devient trop spécifique aux données d'entraînement. Il devient très performant lorsqu'il s'agit de classer les documents de son ensemble d'entraînement, mais quand vient le temps de traiter d'autres textes, son efficacité diminue de beaucoup. En fait, on est en présence d'un manque de généralisation. Plutôt que de retenir seulement les caractéristiques générales d'une catégorie, le classificateur tient aussi compte des particularités des données d'entraînement et pense, à tort, qu'elles s'appliquent à l'ensemble des membres de la catégorie. Plutôt que de s'entraîner seulement sur les relations systématiques entre les valeurs des attributs et l'appartenance à une catégorie, il s'entraîne aussi sur les relations accidentelles. C'est évidemment une situation à éviter, car cela fait obstacle à une amélioration de la performance des classificateurs. Il semble qu'un espace vectoriel à dimension réduite serait moins propice au sur-apprentissage, probablement parce que les attributs conservés font réellement partie d'une bonne généralisation de la catégorie [Seb99].

Un phénomène supplémentaire à considérer est que la performance de certaines méthodes de sélection d'attributs varie selon le classificateur sur lequel elles sont testées. Entre autres, l'interaction entre les méthodes de sélection d'attributs et les modèles linéaires de classification a été explorée [BGMM02]. Les résultats laissent

croire qu'effectivement, les méthodes de sélection d'attributs et les types de classificateurs présentent divers degrés de compatibilité entre eux.

1.3.2 Extraction d'attributs

Une alternative à la sélection d'attributs pour la réduction de l'espace vectoriel est l'extraction d'attributs. Ce processus consiste à créer à partir des attributs originaux un ensemble d'attributs synthétiques qui maximisent l'efficacité de la classification. Le regroupement de termes («*term clustering*») fait partie de ces techniques et a pour but de grouper les mots qui ont une sémantique commune. Les groupes («*clusters*») ainsi créés deviennent les attributs d'un nouvel espace vectoriel. En ce sens, [BM98a] rapportent des résultats intéressants à propos d'un tel regroupement basé sur la distribution des étiquettes de classe associées à chacun des mots. Sans entrer dans les détails, mentionnons seulement que l'idée de départ est la suivante : si deux mots différents apparaissent dans les mêmes classes et dans les mêmes proportions, les réunir en un seul attribut qui affiche la distribution moyenne ne peut pas affecter négativement la performance de la classification. D'autres critères pour regrouper des termes sont imaginables. En plus, les techniques d'indexation sémantique latente que nous verrons plus en détail au chapitre 3 peuvent être considérées comme des techniques d'extraction d'attributs.

Résumé

Ce chapitre a permis de brosser un portrait global de la classification automatique de textes. Tout d'abord, il s'est attardé sur la nature du problème à résoudre, les difficultés rencontrées et les approches possibles pour le solutionner. Par la suite, il

a été question de différentes stratégies de représentation des documents traités par un classificateur. Le choix judicieux d'un mode de représentation des données est nécessaire, comme pour toute application de l'apprentissage automatique. Enfin, des techniques de sélection et d'extraction d'attributs ont été exposées, celles-ci visant à réduire la taille du vocabulaire à traiter pour que les algorithmes évoluent dans un espace vectoriel de dimension raisonnable. Maintenant que le problème étudié est mieux cerné, poursuivons au chapitre suivant avec la présentation d'algorithmes d'apprentissage automatique transposés dans ce domaine.

Chapitre 2

Algorithmes d'apprentissage automatique appliqués à la catégorisation de textes

Comme il a déjà été précisé, c'est principalement par apprentissage automatique que l'on tente de résoudre le problème de la catégorisation automatique de textes. Dans cette optique, plusieurs algorithmes mis au point pour des problèmes quelconques en apprentissage automatique ont été adaptés et appliqués dans notre domaine de recherche. Il sera question dans ce chapitre de ces algorithmes, d'abord de façon générale puis en détaillant trois d'entre eux. Dans un deuxième temps, on abordera les critères d'évaluation des classificateurs, essentiels pour mesurer la performance des algorithmes et surtout pour les comparer entre eux.

2.1 Algorithmes d'apprentissage

Dans le courant de l'apprentissage automatique, différents types de classificateurs ont été mis au point, toujours dans le but d'atteindre un degré maximal de précision et d'efficacité, chacun ayant ses avantages et inconvénients. Ils partagent toutefois des caractéristiques communes. La construction de la majorité d'entre eux comporte deux principales étapes. D'abord, la définition d'une fonction qui associe à un document une valeur entre 0 et 1 représentant son degré d'appartenance à la

catégorie. Cette fonction est appelée CSV pour «*Categorization Status Value*» et prend différentes formes selon le type de classificateur. La deuxième étape, mais non la moindre, est de déterminer un seuil qui va servir lors de la prise de décision finale, à savoir si oui ou non un document va être accepté ou rejeté de la catégorie. Si la fonction CSV retourne une valeur supérieure au seuil pour un document, alors on décide de l'associer à la catégorie. Plusieurs méthodes de détermination du seuil sont possibles et le choix de l'une d'entre elles peut influencer significativement la performance d'un classificateur. Cette importance du choix du seuil a été abordée plus en profondeur par [Yan01], qui présente entre autres les stratégies suivantes :

- Assigner à chaque document les *n* catégories pour lesquelles le classificateur lui aura donné les plus hauts scores
- Associer à chaque catégorie les *n* meilleurs documents, c'est-à-dire les documents pour lesquels le classificateur aura donné les plus hauts scores pour la catégorie en question
- Choisir un seuil différent pour chaque catégorie, pour maximiser la performance sur cette catégorie, évaluée à l'aide d'un ensemble de validation

Parmi la panoplie de classificateurs existants, on peut faire des regroupements et distinguer des grandes familles. Par exemple, on peut discerner les classificateurs probabilistes qui utilisent l'ensemble d'entraînement, c'est-à-dire les textes déjà classés, pour estimer les paramètres de la distribution de probabilité des mots par rapport aux catégories. C'est dans cette famille qu'on retrouve entre autres le classificateur bayésien naïf. On trouve aussi des classificateurs se basant sur un profil, les classificateurs linéaires. Dans ce contexte, le profil est un vecteur de termes pondérés construit pour chaque catégorie, dans le but de les représenter d'une façon générale. Ce vecteur est bien sûr construit à l'aide des données d'entraînement. Quant un nouveau texte doit être classé, il est alors comparé à ce vecteur «*type*». Un avantage de cette approche est qu'elle produit un classificateur

compréhensible par un humain, dans le sens où le profil de la catégorie peut être interprété assez facilement. Par contre, l'inconvénient principal de tous les classificateurs linéaires est que l'espace est divisé en seulement deux portions, ce qui peut être restrictif, car tous les problèmes ne sont pas nécessairement linéairement séparables. Parmi les nombreux membres de cette famille, nous retrouvons Rocchio, Widrow-Hoff et EG [LSCP96]. Les machines à support vectoriel s'apparentent aux classificateurs linéaires, dans le sens où elles tentent de séparer l'espace en deux, mais certaines manipulations mathématiques les rendent adaptables à des problèmes non linéaires. Il y a aussi une famille de classificateurs qui se basent sur l'exemple. On parle alors d'apprentissage à base d'instances. Les nouveaux textes à classer sont comparés directement aux documents de l'ensemble d'entraînement. L'algorithme des k-voisins les plus proches est sans doute le plus connu de cette famille.

Dans les pages qui suivent, trois de ces algorithmes seront exposés plus en détail. D'abord, le classificateur bayésien naïf qui, même s'il a été surpassé par d'autres algorithmes, est souvent utilisé comme point de référence à cause de sa simplicité et du fait qu'il est connu depuis assez longtemps. Puis, les machines à support vectoriel et l'algorithme des k-voisins les plus proches, qui représentent vraisemblablement à l'heure actuelle les deux meilleurs choix en catégorisation de textes [YL99] [Seb99] [Joa98a].

2.1.1 Classificateur bayésien naïf

Comme son nom l'indique, ce classificateur se base sur le théorème de Bayes permettant de calculer les probabilités conditionnelles. Dans un contexte général, ce théorème fournit une façon de calculer la probabilité conditionnelle d'une cause

sachant la présence d'un effet, à partir de la probabilité conditionnelle de l'effet sachant la présence de la cause ainsi que des probabilités a priori de la cause et de l'effet. On peut résumer son utilisation lorsqu'il est appliqué à la classification de textes ainsi : on cherche la classification qui maximise la probabilité d'observer les mots du document. Lors de la phase d'entraînement, le classificateur calcule les probabilités qu'un nouveau document appartienne à telle catégorie à partir de la proportion des documents d'entraînement appartenant à cette catégorie. Il calcule aussi la probabilité qu'un mot donné soit présent dans un texte, sachant que ce texte appartient à telle catégorie. Par la suite, quant un nouveau document doit être classé, on calcule les probabilités qu'il appartienne à chacune des catégories à l'aide de la règle de Bayes et des chiffres calculés à l'étape précédente [Mit97].

La probabilité à estimer est donc : $P(c_j \mid a_1, a_2, a_3, \ldots, a_n)$

où

- c_j est une catégorie
- a_i est un attribut

À l'aide du théorème de Bayes, on obtient : $\frac{P(a_1,a_2,a_3,\ldots,a_n|c_j)P(c_j)}{P(a_1,a_2,a_3,\ldots,a_n)}$

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

En guise de simplification, on calcule $P(a_1, a_2, a_3, \ldots, a_n \mid c_j)$ ainsi : $\prod_{i=1}^{n} P(a_i|c_j)$.

On assume, en simplifiant les choses ainsi, que tous les mots sont indépendants sachant la classe du document[5]. Autrement dit, on assume que la probabilité qu'un

[5] C'est à cause de cette simplification qu'on qualifie ce classificateur de *«naïf»*.

mot apparaisse dans un texte est indépendante de la présence des autres mots du texte. On sait que cela est faux. Par exemple, la probabilité de présence du mot «*artificielle*» dépend partiellement de la présence du mot «*intelligence*». Pourtant, cette supposition n'empêche pas un tel classificateur de présenter des résultats satisfaisants. Et surtout, elle réduit de beaucoup les calculs nécessaires. Sans elle, il faudrait tenir compte de toutes les combinaisons possibles de mots dans un texte, ce qui d'une part impliquerait un nombre important de calculs, mais aussi réduirait la qualité statistique de l'estimation, puisque la fréquence d'apparition de chacune des combinaisons serait très inférieure à la fréquence d'apparition des mots pris séparément.

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

$$\frac{n_k+1}{n+|vocabulaire|}$$

où

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

2.1.2 Algorithme des k-voisins les plus proches

L'algorithme des k-voisins les plus proches («*k-nearest neighbors*» ou kNN) est une méthode d'apprentissage à base d'instances. Il ne comporte pas de phase

d'entraînement en tant que telle. Les documents faisant partie de l'ensemble d'entraînement sont seulement emmagasinés. Lorsqu'un nouveau document à classer arrive, il est comparé aux documents d'entraînement à l'aide d'une mesure de similarité. Ses k plus proches voisins sont alors considérés : on observe leur catégorie et celle qui revient le plus parmi les voisins est assignée au document à classer. C'est là une version de base de l'algorithme que l'on peut raffiner. Souvent, on pondère les voisins par la distance qui les sépare du nouveau texte. On accorde plus de poids, lors de la prise de décision, aux documents plus similaires.

La valeur de k est un des paramètres à déterminer lors de l'utilisation de ce type de classificateur. Souvent, on procède à une optimisation de ce paramètre à l'aide de tests sur une portion des documents. Il est à noter que si l'on ne pondère pas les voisins, la valeur que l'on choisit pour k va être plus critique, plus déterminante en rapport avec la performance du classificateur. Par contre, si l'on décide de pondérer les voisins, l'importance du paramètre k va être atténuée. On peut se permettre de considérer un plus grand nombre de voisins, sachant que plus ils diffèrent du document à classer, moins ils ont d'impact sur la prise de décision. Cependant, il demeure nécessaire de limiter le nombre de voisins pour s'en tenir à un temps de calcul raisonnable.

Une des caractéristiques fondamentales de ce type de classificateur est l'utilisation d'une mesure de similarité entre les documents. Les textes étant représentés sous forme vectorielle, donc comme des points dans un espace à n dimensions, on peut au premier abord penser à déterminer les voisins les plus proches en calculant la distance euclidienne entre ces points.

D'autres façons de calculer la similarité des documents sont utilisées, comme la similarité cosinusoïdale, et sont préférables en classification de textes pour

plusieurs raisons. Premièrement, la similarité cosinusoïdale permet de comparer des textes de longueurs différentes en normalisant leur vecteur. De plus, elle met l'accent plutôt sur la présence de mots que sur l'absence de mots. Justement, la présence de mots est probablement plus représentative de la catégorie du texte que l'absence de mots.

Distance euclidienne entre deux documents a et b : $\sqrt{\sum_{t \in T} (p_t(a) - p_t(b))^2}$

où

- T est l'ensemble des attributs
- $p_t(a)$ est le poids du terme t dans le document a
- $p_t(b)$ est le poids du terme t dans le document b

Similarité cosinusoïdale entre deux documents a et b : $\sum_{t \in T} \frac{p_t(a) \cdot p_t(b)}{\sqrt{\sum_{t \in T} p_t(a)^2 \cdot \sum_{t \in T} p_t(b)^2}}$

où

- T est l'ensemble des attributs
- $p_t(a)$ est le poids du terme t dans le document a
- $p_t(b)$ est le poids du terme t dans le document b

Regardons à l'aide d'un exemple pourquoi la deuxième mesure est préférable à la première. Nous allons considérer le vocabulaire et les trois documents suivants :

- Vocabulaire : <classification, automatique, similarité, document>
- Document 1 : <1, 1, 1, 0>

- Document 2 : <0, 0, 1, 0>
- Document 3 : <0, 0, 0, 0>

La mesure de distance euclidienne indique que le document 3 est moins distant (plus similaire) du document 2 (distance de 1) que le document 2 ne l'est du document 1 (distance de 1.41). Pourtant, les documents 2 et 3 n'ont aucun mot en commun, alors que les documents 1 et 2 partagent le mot «*similarité*». La mesure de similarité cosinusoïdale, pour sa part, retourne une similarité nulle entre les documents 2 et 3, mais positive (0.53) entre les documents 1 et 2. Ces derniers résultats sont plus représentatifs de la réalité, puisque les documents 1 et 2 partagent au moins un mot, donc présentent une certaine similarité et sont plus susceptibles d'être classés dans la même catégorie que les documents 2 et 3 [Sou02].

Comme il a déjà été mentionné, le classificateur kNN n'implique pas de phase d'entraînement en tant que telle. Cela nous amène à discuter d'une classe d'«*apprentisseurs*» qu'on peut qualifier de paresseux («*lazy learners*»). La seule opération préalable est le stockage des exemples d'entraînement. L'apprentissage est repoussé au moment où un nouveau document à classer arrive. Par le fait même, la plus grosse part de l'effort requis en termes de temps de calcul est fourni au moment même de la classification.

Une des caractéristiques de l'apprentissage à base d'instances est qu'il n'y a pas de construction d'une description explicite de la fonction à apprendre (dans notre cas, l'appartenance à une catégorie). L'avantage est qu'on n'estime pas qu'une seule fois la fonction pour tout l'espace, mais on l'estime plutôt localement et différemment pour chaque nouvelle instance. L'inconvénient est que bien que le coût d'entraînement du classificateur soit faible, car il ne fait que mémoriser les

exemples d'entraînement, le coût de classification de nouvelles instances peut être élevé, puisque c'est à ce moment que tout le calcul se fait. Cependant, une bonne indexation des exemples aide beaucoup à pallier ce problème [Mit97].

2.1.3 Machines à support vectoriel

Les machines à support vectoriel (*«Support Vector Machines»* ou SVM) [Vap95] forment une classe d'algorithmes d'apprentissage qui peuvent s'appliquer à tout problème qui implique un phénomène f et qui, à partir d'un jeu d'entrées x, produit une sortie $y = f(x)$, et où le but est de retrouver f à partir de l'observation d'un certain nombre de couples entrée/sortie. Le problème revient à trouver une frontière de décision qui sépare l'espace en deux régions, à trouver l'hyperplan qui classifie correctement les données et qui se trouve le plus loin possible de tous les exemples. On dit qu'on veut maximiser la marge, la marge étant la distance du point le plus proche de l'hyperplan.

Figure 2.1 - Maximisation de la marge avec les SVM

(a) une frontière possible séparant les données

(b) la frontière entraînant une marge maximale

Dans la figure 2.1, la partie de droite représente la frontière recherchée entre les données. Il s'agit de la ligne en trait continu. La marge, soit la distance entre cette ligne et les lignes en trait pointillé, y est maximale. La partie de gauche présente une situation où la marge serait plus petite. Les points situés sur les lignes en trait pointillé sont ceux qu'on appelle les vecteurs de support.

Des techniques de programmation quadratique permettent de résoudre ce problème, un algorithme efficace étant présenté entre autres dans [Pla99]. Une propriété intéressante des SVM est que la surface de décision est déterminée uniquement par les points qui en sont les plus près, les vecteurs de support. En présence de ces seuls exemples d'entraînement, la même fonction serait apprise. C'est une différence par rapport à des algorithmes comme kNN avec lesquels tous les exemples d'entraînement sont utilisés lors de l'apprentissage [YL99].

Même s'ils cherchent l'hyperplan séparant l'espace vectoriel en deux, l'avantage des SVM est qu'ils s'adaptent facilement aux problèmes non linéairement séparables. Avant de procéder à l'apprentissage de la meilleure séparation linéaire, on transforme les vecteurs d'entrée en vecteurs de caractéristiques de dimension plus élevée. De cette façon, un séparateur linéaire trouvé par un SVM dans ce nouvel espace vectoriel devient un séparateur non linéaire dans l'espace original. Cette transformation des vecteurs se fait à l'aide de noyaux («*kernels*»).

Dans le cas de la classification de textes, les entrées sont des documents et les sorties sont des catégories. En considérant un classificateur binaire, on voudra lui faire apprendre l'hyperplan qui sépare les documents appartenant à la catégorie et ceux qui n'en font pas partie. Selon [Joa98a], les SVM conviennent bien pour la classification de textes parce que, premièrement, une dimension élevée ne les affecte pas puisqu'ils se protègent contre le sur-apprentissage. Dans le même sens,

il affirme que peu d'attributs sont totalement inutiles à la tâche de classification et que les SVM permettent d'éviter une sélection agressive qui aurait comme résultat une perte d'information. On peut se permettre de conserver plus d'attributs. Également, une caractéristique des documents textuels est que lorsqu'ils sont représentés par des vecteurs, une majorité des entrées sont nulles. Or, les SVM conviennent bien à des vecteurs dits clairsemés. Un autre aspect positif des SVM est qu'aucun ajustement de paramètres manuel n'est requis, car ils ont l'habileté de trouver automatiquement des paramètres adéquats.

2.1.4 Autres types de classificateurs

En plus du classificateur bayésien naïf, de l'algorithme des k-voisins les plus proches et des SVM, plusieurs autres types de classificateurs ont été l'objet d'expérimentations, mais ne seront pas vus en détail ici. Parmi ceux-ci, notons les réseaux de neurones, les arbres de décisions, les «*apprentisseurs*» de règles en forme normale disjonctive, LLSF («*Linear Least Squares Fit*») basé sur une régression par les moindres carrés, etc.

Une approche qui peut sembler séduisante au premier abord est celle des comités de classificateurs. Intuitivement, il est légitime de croire que devant une tâche nécessitant la connaissance d'un expert, plusieurs experts travaillant ensemble et combinant adéquatement leurs jugements pourraient être plus performants que le jugement d'un seul expert pris séparément. Une méthode envisagée pour classifier automatiquement des textes est donc d'interroger différents classificateurs et de combiner leurs décisions de classification, soit par un vote à majorité, soit en pondérant chaque classificateur selon sa performance testée sur des exemples de validation semblables aux documents en question. Théoriquement, plus les

classificateurs sont indépendants de par leur conception, leur entraînement et leur façon de classifier, plus la performance du comité devrait être bonne. Jusqu'à maintenant, il semble que cette approche ait présenté des résultats partagés, mais qu'aucune expérience d'envergure vraiment significative n'ait été menée. Il est clair qu'une telle approche est très coûteuse sur le plan informatique étant donné qu'il faut entraîner et faire fonctionner les différents classificateurs, puis combiner leurs résultats. La porte est ouverte de ce côté pour vérifier si le jeu en vaut vraiment la chandelle [Seb99]. Dans cet ordre d'idées, l'algorithme AdaBoost [SSV00] a, quant à lui, permis d'obtenir des résultats intéressants. Son fonctionnement général revient à entraîner un comité de classificateurs en passant par plusieurs itérations, en donnant à chaque étape plus de poids aux exemples incorrectement classifiés à l'itération précédente et en retenant pour les étapes subséquentes les classificateurs les plus prometteurs.

2.2 Critères d'évaluation des classificateurs

Face à toutes ces différentes méthodes de classification, comment les comparer ? Sur la base de quels critères peut-on dire que tel classificateur est meilleur qu'un autre ? Comme c'est le cas avec les systèmes de recherche d'information («*information retrieval*»), on se base sur l'expérience pour évaluer les classificateurs de textes, plutôt que de procéder analytiquement. La raison en est simple : pour pouvoir évaluer un système analytiquement, on doit posséder une définition formelle du problème à résoudre. Il faudrait pouvoir spécifier à quoi correspondent exactement la rectitude et la complétude. En fait, c'est l'appartenance d'un document à une catégorie, sa pertinence au sein d'une catégorie, qu'il faut définir. Cependant, un caractère très subjectif ressort de ce

concept, qui relève aussi de la sémantique d'un texte. Donc, pour l'instant, ce n'est pas formalisable. On se replie alors sur une évaluation empirique des classificateurs [Seb99].

Presque toujours, on divise la banque de textes déjà classés et disponibles en deux ensembles : l'ensemble d'entraînement sur lequel le classificateur fait son apprentissage et l'ensemble de test sur lequel on peut évaluer sa performance. Parfois, on peut former un troisième ensemble, l'ensemble de validation, qu'on utilise pour l'optimisation de certains paramètres. Donc, l'ensemble de test contient des documents dont on connaît à l'avance les catégories auxquelles ils devraient appartenir. On pourra ainsi comparer les décisions prises par le classificateur automatique à celles des experts humains et calculer un score de performance. Ce calcul peut se faire de diverses façons, dont voici les principales.

On se considère ici en présence d'un classificateur binaire pour chaque catégorie, indiquant si oui ou non un document y appartient. Pour mieux illustrer les différentes mesures qui vont suivre, on prend pour point de départ la table de contingence illustrée par le tableau 2.1, distincte pour chaque classe.

	Documents appartenant à la catégorie	Documents n'appartenant pas à la catégorie
Documents assignés à la catégorie par le classificateur	*a*	*b*
Documents rejetés de la catégorie par le classificateur	*c*	*d*

Tableau 2.1 - Table de contingence à la base de l'évaluation des classificateurs

On définit à partir des statistiques de cette table les mesures suivantes :

- **précision** (*«precision»*) : a / (a + b), soit le nombre d'assignations correctes sur le nombre total d'assignations
- **rappel** (*«recall»*) : a / (a + c), soit le nombre d'assignations correctes sur le nombre d'assignations qui auraient dû être faites
- **exactitude** (*«accuracy»*) : (a + d) / (a + b + c + d)
- **erreur** (*«error»*) : (b + c) / (a + b + c + d)

Les deux dernières mesures, bien que couramment utilisées en apprentissage automatique, sont jugées moins bien adaptées à la tâche de classification de textes. Elles incluent dans leur définition le nombre total de documents. Or, comme un document appartient généralement à un petit nombre de catégories sur l'ensemble, un classificateur qui rejetterait tous les documents présenterait seulement un faible taux d'erreur et une exactitude quand même très élevée. Entraîner un classificateur sur la base de l'optimisation d'un de ces deux critères tendrait à créer un programme qui n'accepte aucun document dans sa catégorie. C'est la raison pour laquelle la précision et le rappel sont les mesures les plus rencontrées dans la littérature.

Ce qui est recherché, c'est un score global et non un score pour chaque catégorie. Il y a deux principales façons de faire une moyenne pour obtenir un score global. La macro-moyenne calcule d'abord les scores pour chaque catégorie et fait ensuite une moyenne sur ces scores. La micro-moyenne regroupe d'abord les données de chaque catégorie dans une même table de contingence globale et calcule ensuite les scores à partir de celle-ci. La distinction à faire entre ces deux méthodes est que la macro-moyenne donne une importance égale à toutes les catégories, tandis que la micro-moyenne donne une importance égale à tous les documents. Les avis sont partagés quant à savoir laquelle des deux mesures est préférable. Les tenants de la macro-moyenne affirment que la micro-moyenne est moins représentative parce que les catégories plus fréquentes ont plus de poids que les autres. Et c'est justement cet aspect de la micro-moyenne que ses défenseurs apprécient [Seb99].

Bref, il semble préférable, lors de la présentation des résultats d'une expérience, d'identifier clairement la mesure utilisée. La comparaison avec d'autres expériences sera ainsi plus facile.

Lors de l'évaluation de la performance d'un classificateur, on ne peut tenir compte de la précision ou du rappel séparément. Effectivement, on pourrait mettre en place un système qui rejetterait tous les textes : il obtiendrait une précision de 100%, mais un rappel de 0%. À l'inverse, un système qui accepterait tous les textes aurait un rappel de 100%, mais une précision de 0%. On voit donc que ce vers quoi il faut tendre est un classificateur qui fait le compromis idéal entre ces deux facteurs.

Dans cet ordre d'idées, on peut calculer le «*break-even point*», c'est-à-dire le point où la précision et le rappel sont égaux. On a souvent utilisé cette mesure pour comparer la performance de classificateurs. Plus ce point se rapproche de 100%, plus le classificateur est performant à la fois en précision et en rappel.

Aussi, la mesure F1 est beaucoup utilisée. Elle est définie ainsi :

$$\mathrm{F1} = \frac{2rp}{(r+p)}$$

où

- r est le rappel
- p est la précision

C'est une fonction qui est maximisée quand la précision et le rappel sont proches. On cherche généralement à l'optimiser lors de l'ajustement du seuil. On peut aussi utiliser une forme plus générale de la mesure F en ajoutant un paramètre qui pondère l'importance relative des deux critères. Il se peut en effet qu'une

application particulière nécessite une précision élevée, mais puisse se permettre un rappel un peu moins bon et vice versa [Yan99].

À la vue de cette variété de critères, un des points importants à considérer lorsqu'il s'agit d'évaluer en parallèle différents classificateurs est la comparabilité des mesures de performance utilisées. Chaque chercheur doit donc se faire un devoir de mentionner précisément la façon dont il a procédé pour évaluer son classificateur. Ainsi, la comparaison de différentes techniques de classification automatique, qui apparaît déjà assez complexe, serait un peu plus facile.

Résumé

Suite à la présentation de la classification automatique de textes au chapitre 1, il a été question ici des algorithmes d'apprentissage mis à l'œuvre pour traiter ce problème. On a pu constater la variété de techniques d'apprentissage pouvant amener une application informatique à classer des textes avec autonomie. De façon détaillée, on comprend maintenant mieux le fonctionnement du classificateur bayésien naïf, de l'algorithme des k-voisins les plus proches et des machines à support vectoriel. En plus, la deuxième partie du chapitre a fait la lumière sur le processus d'évaluation des classificateurs. Le chapitre suivant introduira un axe de recherche particulier dans lequel s'inscrivent nos travaux, à savoir l'étude de moyens permettant de faire profiter les classificateurs de connaissances extérieures à l'ensemble d'entraînement.

Chapitre 3

Utilisation de connaissances extérieures à l'ensemble d'entraînement

À la lumière de ce qui a été introduit dans les deux chapitres précédents sur la catégorisation automatique de textes, on se rend compte que plusieurs éléments sont impliqués dans le processus, que ce soit le mode de représentation des documents, la méthode de sélection d'attributs ou l'algorithme d'apprentissage mis en place. Indépendamment de ces facteurs, une chose ne change pourtant pas : un corpus d'entraînement est essentiel. Et comme il en a déjà été question, une des difficultés rencontrées lors de la création d'un classificateur automatique de textes est la cueillette d'un grand nombre de documents déjà classés pour constituer ce corpus d'entraînement. Il se trouve que plus ils sont entraînés sur des corpus de textes imposants, plus les classificateurs ont l'opportunité de bien performer lors de la classification de nouveaux documents, c'est-à-dire de produire une classification qui ressemble à celle qu'aurait produite un humain. Parallèlement à ce besoin de documents libellés qui sont pourtant coûteux à obtenir, les textes non étiquetés, non classés, sont disponibles en beaucoup plus grand nombre. Alors, pourquoi ne pas essayer d'utiliser ces derniers pour améliorer la performance des classificateurs ? Car on sait que, même si ces textes non étiquetés ne permettent pas directement de retracer les liens entre les mots et les catégories, ils contiennent de

l'information pertinente sur les liens qu'entretiennent les mots entre eux. Dans la même optique, à défaut d'utiliser des documents non libellés, une autre option envisageable est de fournir à un classificateur des connaissances extraites d'un lexique. Bref, ce chapitre apportera d'abord une réflexion sur le coût d'étiquetage des documents pour ensuite discuter de quelques propositions faites dans le passé pour qu'un classificateur puisse tirer profit de connaissances extérieures à son ensemble d'entraînement, soit à partir d'un lexique ou directement à partir de textes non libellés.

3.1 Coût d'étiquetage des documents d'entraînement

Dans un monde idéal, il serait à coup sûr très intéressant de pouvoir déterminer avec précision le coût d'étiquetage des documents d'entraînement, le coût de la constitution du corpus d'entraînement. De combien de temps a besoin un humain pour associer un texte à une catégorie ? En pratique, il s'agit d'une question difficile à répondre. Assurément, plusieurs variables influencent le phénomène et les lignes qui suivent porteront sur certaines d'entre elles.

Sans doute qu'une grande partie du temps consommé pour catégoriser un texte est consacré à sa lecture, puis possiblement à sa relecture. On peut aussi imaginer que la longueur des textes à catégoriser est assez déterminante du temps qui va être requis pour cette activité, et que la vitesse de lecture varie d'une personne à une autre. Une fois cette étape accomplie, il faut déterminer à quelle(s) catégorie(s) ce texte appartient. Au temps de réflexion qui s'impose s'ajoute, si nécessaire, le temps de se référer à la description des catégories et éventuellement de consulter d'autres textes préalablement associés à certaines classes, pour valider la décision. D'autres facteurs interviennent également comme, entre autres, l'homogénéité des

textes appartenant à une même catégorie. En effet, si tous les textes portant une étiquette donnée se ressemblent beaucoup (du point de vue de la forme ou du vocabulaire par exemple), il devient plus facile de les reconnaître rapidement. Également, on peut supposer que le style littéraire des textes influence à un certain niveau le temps requis pour en saisir le sens et en déterminer la catégorie. Il faut finalement remarquer que l'ensemble de catégories possibles fait une différence : plus il y a de classes différentes, autrement dit plus il y a d'étiquettes possibles pour un texte donné, plus il est difficile de faire un choix parmi celles-ci. Aussi, plus la sémantique des catégories est précise, fine, détaillée, plus il faut faire attention avant d'y associer un document. À cet égard, pensons à une situation où il faudrait différencier des textes appartenant soit à la catégorie «*sciences naturelles*», soit à la catégorie «*sciences humaines*». Cette tâche est probablement plus aisée que celle de distinguer des textes appartenant à l'une ou l'autre des catégories «*biochimie*», «*biologie*» et «*médecine*».

Une fois qu'un premier étiquetage a été effectué, il peut arriver, dépendamment de l'importance et de l'aspect critique de l'application, qu'une étape de révision soit mise en place, menant éventuellement lieu à des corrections. Pour être plus pertinente, la révision implique vraisemblablement un nouvel intervenant. Voilà une étape dont le temps s'additionne aux temps précédents. Mentionnons pour terminer cette discussion un autre facteur influençant le temps requis pour étiqueter des documents : le degré d'expertise de l'humain impliqué. Effectivement, les premiers textes qu'une personne classe vont probablement lui prendre plus de temps, demanderont une plus grande attention lors de la révision et contiendront potentiellement plus d'erreurs.

Il ne faut pas oublier non plus que lors de la création d'un corpus pour une nouvelle application, à partir de zéro, il est nécessaire de consacrer du temps à

l'élaboration de la liste de catégories possibles pour les documents. Il s'agit d'une étape assez importante qui s'apparente à l'ingénierie d'une ontologie[6]. Le tout se doit d'être cohérent, de couvrir tous les cas possibles, de se situer à un niveau de détail adéquat par rapport à l'application visée. Bref, cette tâche consistant à choisir l'ensemble de catégories consomme un temps qui vient s'ajouter à celui de l'étiquetage en tant que tel.

En somme, on voit qu'un grand nombre de variables influencent le coût d'étiquetage. Il apparaît donc difficile de l'évaluer a priori, analytiquement. Par contre, après coup, une fois qu'un corpus a été mis au point, il est possible de regarder en arrière pour constater le coût effectif qui a été engendré. À ce sujet, les données publiées sont rares. L'agence de presse Reuters, qui classe les nouvelles qu'elle diffuse, a rendu disponible pour la recherche des collections de textes catégorisés. Plusieurs versions de ces collections ont été utilisées par un grand nombre de chercheurs dans le domaine de la catégorisation automatique de textes. Certains chiffres intéressants ont été obtenus de l'organisation et sont présentés dans [LYRL04]. Dans les années 1996-1997, Reuters a produit un peu plus de 800 000 nouvelles en anglais par année. Si l'on ajoute aux articles écrits par les journalistes de l'agence ceux provenant d'autres sources, on arrive à un total de 5.5 millions de textes anglais par année à catégoriser. À un moment, l'organisation employait 90 personnes dédiées à l'étiquetage de ces documents. Pour le corpus RCV1[7], qui comprend les 800 000 documents de l'agence, l'estimation du coût de l'étiquetage est de 12 années-personnes.

---

[6] Une ontologie permet de modéliser un domaine en fournissant un vocabulaire de termes et de relations caractéristiques qui peut être partagé par différents systèmes. C'est un concept très important en ingénierie des connaissances.

[7] Des détails sur ce corpus seront ajoutés au chapitre 6.

La discussion qui précède permet de constater qu'il est en effet coûteux de construire un corpus d'entraînement. Force est aussi d'admettre qu'il y a un besoin pour ceux-ci. En effet, les collections de textes qui existent déjà comportent certaines faiblesses, comme le manque de documents, des assignations de catégories incomplètes ou incohérentes et une disponibilité limitée. De toute façon, même si les corpus étaient parfaits, il y aurait toujours un intérêt pour en avoir des nouveaux. Le phénomène de sur-apprentissage qui se produit quand un algorithme d'apprentissage modifie ses paramètres en fonction des propriétés accidentelles d'un ensemble d'entraînement peut se produire aussi à une autre échelle. Une communauté de chercheurs peut sur-apprendre en raffinant les algorithmes en les roulant constamment sur les mêmes ensembles de données existants. La vérification qu'un véritable progrès est acquis ne peut être vraiment faite qu'en effectuant des tests sur de nouvelles collections de test [LYRL04].

Ces remarques font plutôt référence à des considérations relatives à la recherche dans le domaine : des corpus de textes libellés sont nécessaires pour mettre au point et pour tester de nouveaux algorithmes d'apprentissage. C'est un premier aspect, mais le problème se fait peut-être encore plus sentir lorsque vient le temps d'appliquer dans une situation concrète ces algorithmes. Il est alors nécessaire de soumettre au classificateur des exemples d'entraînement représentatifs de l'application concernée. Un corpus est encore une fois nécessaire. Dépendamment des circonstances, les ressources ne sont pas toujours disponibles pour permettre la création d'un corpus suffisamment consistant pour conférer au classificateur la capacité d'accomplir sa tâche de façon satisfaisante.

3.2 Utilisation de connaissances extérieures

Donc, pour toutes les raisons mentionnées précédemment, une question importante mérite d'être étudiée : quelles autres sources d'information pourraient réduire le besoin de textes libellés ? Plusieurs travaux ont porté sur des tentatives d'intégrer des connaissances linguistiques ou lexicales à des classificateurs automatiques de textes. Des techniques de résumé automatique ont été appliquées pour évaluer l'importance des phrases [KPS02]. Aussi, des techniques de lissage, développées dans le cadre des modèles de langue n-grammes, ont été employées pour calculer la pondération d'attributs [Aiz01]. Par ailleurs, [BMP01] ont profité de l'étiquetage lexical, produisant de l'information concernant les parties du discours, pour caractériser les attributs des textes. Il s'agit là de quelques exemples parmi une multitude de tentatives qui, en général, n'ont pas obtenu le succès escompté. Certains travaux ont permis de hausser légèrement la performance d'un classificateur sur une tâche donnée, mais, globalement, aucune approche réellement concluante ne s'est démarquée.

3.2.1 Utilisation d'un lexique

Dans cette lignée, un certain nombre de chercheurs se sont penchés sur le lexique WordNet[8] et ont cherché une façon efficace de l'exploiter pour améliorer les performances de classificateurs de textes. WordNet est une ontologie d'héritages lexicaux en ligne dans laquelle les mots sont regroupés au sein de groupes de synonymes appelés «*synsets*». Chacun de ces groupes représente un concept lexical sous-jacent. Ces groupes sont reliés entre eux par différents types de liens

[8] Accessible à l'adresse : http://www.cogsci.princeton.edu/~wn/

(hyperonymie[9], méronymie[10], etc.), le tout formant un graphe. Cet outil, ayant pour objectif de représenter des aspects sémantiques d'un lexique, a vu le jour il y a plusieurs années et a été développé petit à petit, en partie manuellement, en partie de façon automatisée. Comme il contient beaucoup d'informations et différents types d'informations (près de 150 000 mots à l'heure actuelle), on peut imaginer plusieurs façons de mettre à profit WordNet en catégorisation automatique de textes. À la base, l'idée est que plus un système est informé, mieux il devrait performer.

Une des propositions qui a été faite pour intégrer les connaissances lexicales de WordNet est la suivante : construire deux classificateurs indépendants, l'un d'eux étant tout simplement un classificateur linéaire bâti à partir de l'information puisée dans WordNet et l'autre étant appris à partir d'un ensemble d'entraînement et d'un algorithme d'apprentissage classique. Un classificateur final est entraîné sur un autre ensemble d'entraînement, recevant en entrée les sorties des deux précédents classificateurs. Il s'agit en fait d'une adaptation d'un schéma de méta-apprentissage appelé «*Stacked Generalization*» [GRLVV02]. Dans ce cadre, le classificateur basé sur WordNet fonctionne de la manière suivante. D'abord, les documents sont représentés par des vecteurs de poids TFIDF. Ensuite, pour chaque nom (ou étiquette) de catégorie dans l'ensemble d'entraînement, une série de synonymes est extraite de WordNet manuellement. Le classificateur assigne un document à une catégorie si la similarité cosinusoïdale[11] entre les deux (entre la représentation du document et la liste de synonymes du nom de la catégorie)

---

[9] Hyperonymie : un hyperonyme est un synonyme d'un niveau de généralité immédiatement supérieur (exemple : «*animal*» est un hyperonyme de «*chat*»)

[10] Méronymie : relation hiérarchique existant entre deux concepts ou deux signes linguistiques, dans laquelle le premier est une partie d'un tout que constitue le second [Office de la langue française, 2002] (exemple : «*doigt*» et «*main*»)

[11] Telle que présentée au chapitre 2.

dépasse un certain seuil, appris par validation croisée sur l'ensemble d'entraînement. La logique derrière la «*Stacked Generalization*» est que différents algorithmes d'apprentissage fournissent des explications différentes, mais complémentaires à propos des données. Avec cette approche, la combinaison des prédictions des classificateurs est faite selon le degré de confiance envers chacun. En d'autres mots, le classificateur de second niveau apprend à quel degré il peut faire confiance à tel ou tel classificateur de premier niveau, selon les entrées reçues. Les expérimentations pour tester cette approche n'ont pas permis de démontrer clairement son efficacité. Des hausses de quelques points en termes de mesure F1 ont été obtenus suite à l'intégration de Wordnet à certains types de classificateurs, mais, pour d'autres, rien de positif n'est ressorti. Une observation qui a été faite : les informations provenant de WordNet ont été particulièrement utiles dans les situations où une catégorie présentait peu de documents d'entraînement.

Plusieurs autres façons d'utiliser l'information contenue dans le lexique Wordnet pour assister la classification automatique de textes ont été imaginées. Par exemple, on a appliqué l'«*apprentisseur*» de règles RIPPER sur des textes représentés par une liste de «*synsets*» construite de la façon suivante. Pour chaque nom ou verbe du texte, on fait une requête dans Wordnet pour obtenir une liste globale des «*synsets*» correspondants et de leurs hyperonymes. Une fois la liste constituée, on conserve seulement les «*synsets*» les plus fréquents dans le corpus. Finalement, on associe une valeur numérique à ces «*synsets*» calculée à partir de leur densité, c'est-à-dire leur nombre d'occurrences dans un document par rapport à la taille du document. L'algorithme RIPPER tente ensuite d'apprendre des règles de classification à partir de cette représentation. Un paramètre problématique dans cette méthode est la hauteur jusqu'à laquelle on doit remonter dans la chaîne d'hyperonymes, jusqu'à quel niveau de généralisation on doit se rendre. Cette

valeur semble être difficile à fixer puisque dépendamment des domaines, la pertinence d'hyperonymes très généraux varie [Cha01].

Une autre approche, qui s'éloigne un peu de l'apprentissage automatique, ne comporte pas de phase d'entraînement et se base encore davantage sur le lexique Wordnet. Dans le même sens que la précédente, elle commence par générer toutes les chaînes d'hyperonymes possibles pour chaque nom du texte. Ensuite, toutes les chaînes qui partagent un ou plusieurs «*synsets*» communs sont fusionnées. Celles restantes sont enfin évaluées selon leur fréquence dans le document et leur degré de généralité (leur profondeur dans la chaîne). Les catégories associées au texte sont finalement choisies parmi les concepts ayant obtenu les plus hauts scores [Cha01].

Les deux dernières stratégies présentées et mettant à profit le lexique Wordnet ont, selon leurs créateurs, donné de bons résultats, mais il semble que les expériences menées n'étaient pas à très grande échelle [Cha01]. De plus, ces méthodes n'ont pas été évaluées de façon à pouvoir les comparer clairement et facilement à d'autres. Malgré cela, elles prouvent que l'utilisation des connaissances extérieures aux documents à classer peut être bénéfique pour la classification automatique de textes. Elles donnent également des pistes pour intégrer le lexique Wordnet à ce processus.

3.2.2 Utilisation de documents non libellés

Wordnet représente une source possible d'information extérieure aux documents d'entraînement. Mais une autre source potentielle de connaissances utiles à la classification réside dans les textes non libellés. Il est certain que ces documents ne fournissent pas d'indices au classificateur lui permettant de généraliser encore

mieux les liens existants entre les mots et la catégorie d'un texte. De toute façon, l'idée n'est pas de passer en mode d'apprentissage non supervisé ou de faire du regroupement («*clustering*») de textes en ne tenant plus du tout compte des documents libellés. La proposition est plutôt de mettre à profit le nombre limité de textes d'entraînement disponibles, mais d'ajouter à ces documents les textes non classés pour obtenir de l'information supplémentaire. Ce supplément de connaissances est de nature différente. Un texte non associé à une étiquette de catégorie contient tout de même des mots agencés d'une façon semblable à ceux contenus dans les textes libellés. On peut donc y recueillir de l'information sur la distribution de probabilité jointe des mots [NMTM00].

Dans la plupart des cas, recueillir des textes non libellés constitue une tâche très peu coûteuse à réaliser. On sait que de façon générale, le nombre de documents électroniques à notre disposition augmente sans cesse. Il ne suffit que de les rassembler pour les présenter à un classificateur. Notamment, le Web peut constituer facilement une source de tels documents. Par la suite, il suffit de trouver une façon utile d'en tirer profit.

L'approche de co-apprentissage («*co-training*») présentée dans [BM98b] classe les documents non libellés à l'aide d'une version préliminaire du classificateur et les ajoute à l'ensemble d'entraînement qui sera exploité pour sa phase finale d'apprentissage. La particularité du co-apprentissage est qu'il est conçu pour s'appliquer quand les données peuvent être représentées sous différents angles. Dans le cas des pages Web par exemple, elles peuvent être décrites par les mots qu'elles contiennent, mais aussi par les hyperliens pointant vers elles. L'idée est donc de créer au départ deux hypothèses apprises respectivement avec l'une et l'autre des représentations des documents, de classer les textes non libellés avec ces

hypothèses et d'ajouter les nouveaux exemples à l'ensemble d'entraînement de l'autre hypothèse.

Une autre expérience intéressante a été menée par [NMTM00] pour tenter d'exploiter les documents non libellés. On a entraîné un classificateur bayésien naïf sur des textes étiquetés et non étiquetés, à l'aide de l'algorithme itératif EM («*Expectation-Maximization*»). Cet algorithme d'estimation de maximum de vraisemblance par itération successive de deux étapes a été introduit par [DLR78]. Il est particulièrement adapté pour les problèmes avec des données incomplètes, ce qui est précisément le cas dans la situation à laquelle on s'intéresse, puisque les textes non libellés sont dépourvus d'étiquettes de classe. De la même façon, un classificateur de type bayésien naïf se prête bien à l'application de l'algorithme EM, étant donné que la tâche d'apprentissage avec ce classificateur devient une estimation des paramètres de la distribution de probabilités ayant généré les données. Sommairement, voici comment se déroule la création de ce classificateur, qui s'effectue en deux temps. La figure 3.1 illustre ce processus. Une première phase d'estimation («*expectation*») consiste à générer un classificateur bayésien naïf habituel à partir des documents libellés disponibles. Par la suite, lors de la phase de maximisation («*maximization*»), on assigne une classe aux documents non libellés en utilisant ce dernier. Puis, on entraîne un nouveau classificateur en utilisant cette fois les deux ensembles de documents. On répète les deux dernières étapes selon le principe de l'escalade («*hill-climbing*»), jusqu'à ce que le classificateur cesse de s'améliorer. Les résultats obtenus sont positifs : cette méthode a réduit de beaucoup le nombre de documents libellés requis pour obtenir une précision donnée [NMTM00].

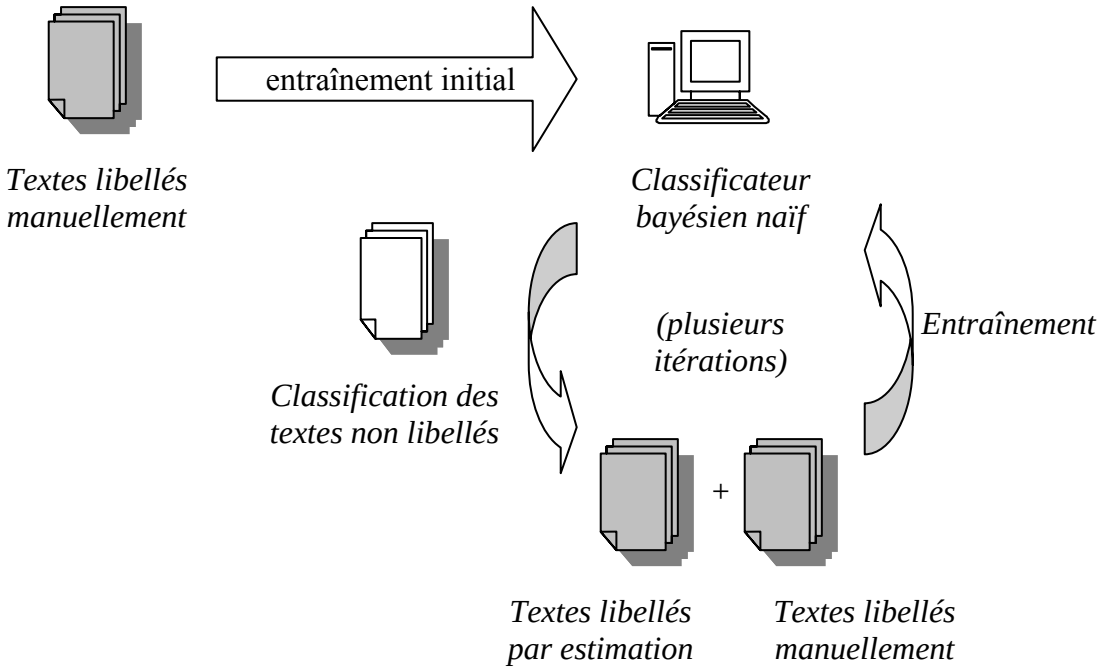

Figure 3.1 - Algorithme EM

Un exemple simple va illustrer plus concrètement cette idée. Prenons comme point de départ le fait que les documents d'entraînement d'un classificateur lui ont permis d'associer la présence du mot «*guitare*» à la catégorie «*musique*». Par la suite, l'estimation de la classe d'un grand nombre de documents non libellés permet de constater la présence du mot «*mandoline*» dans cette même catégorie. Un nouveau classificateur peut alors être considéré, et va dorénavant utiliser ces deux mots comme indicateurs de la catégorie «*musique*». Même si le mot «*mandoline*», plus rare, n'apparaissait pas dans les documents d'entraînement (ou pas assez significativement), une grande banque de textes non libellés a permis de le relier à une catégorie.

Une autre proposition visant à tirer profit de connaissances extérieures aux documents d'entraînement est d'utiliser la technique qu'est l'indexation

sémantique latente. Celle-ci se base sur l'idée qu'il existe une structure sémantique sous-jacente aux données textuelles et que les relations entre les termes et les documents d'un corpus peuvent être décrites dans cette forme de structure sémantique. La figure 3.2 illustre ce phénomène. À la base, on considère une matrice deux dimensions terme/document où chaque entrée correspond à l'absence ou la présence d'un mot dans un document. L'absence d'un mot est représentée par un 0 et la présence d'un mot par une valeur positive correspondant au poids du mot. Cette valeur peut être calculée par diverses méthodes de pondération, semblables à celles utilisées pour la représentation vectorielle[12]. L'objectif de l'indexation sémantique latente est de réduire cette matrice de grande dimension en une matrice qui capturerait mieux les relations entre les documents. La première étape est de calculer la décomposition en valeurs singulières[13] de cette matrice pour obtenir un produit de trois autres matrices TSD^T, où la matrice S est diagonale. On ordonne les éléments de la diagonale en ordre croissant et on élimine ensuite les *k* plus petites valeurs. Les colonnes de T et D correspondantes sont aussi éliminées. Le produit de ces trois nouvelles matrices nous donne une nouvelle matrice représentant les liens entre les termes et les documents, mais dans une dimension moindre. Cette représentation modifiée des documents peut être la base d'un algorithme de classification. Cette matrice concerne les documents d'entraînement, mais il suffit de transposer les nouveaux documents à classer dans cet espace vectoriel en les multipliant par T et S^{-1}. On peut ensuite comparer leur similarité avec les textes d'entraînement pour procéder à la classification. L'indexation sémantique latente a été d'abord développée pour pallier les problèmes de synonymie et de polysémie souvent rencontrés lors du traitement de

[12] Voir la section 1.2 à ce sujet.

[13] Technique bien connue en algèbre linéaire permettant de résoudre certains problèmes et d'exposer des propriétés intéressantes de la matrice originale. Autres applications intéressantes : compression d'images, réduction de bruit dans les signaux sonores, etc.

la langue naturelle. Elle a permis d'obtenir de bons résultats en classification automatique de textes. Cependant, l'inconvénient principal est que la nouvelle représentation des documents devient difficile à interpréter par un humain par la suite [ZH01].

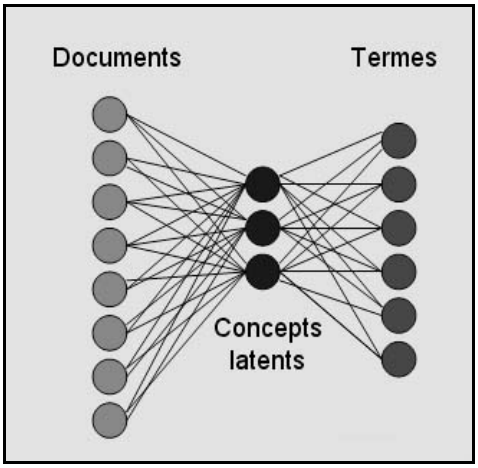

Figure 3.2 - Concepts latents

Cette technique peut facilement être étendue pour tirer profit de documents non classés. La stratégie proposée par [ZH01] (et illustrée à la figure 3.3) est de se servir de l'indexation sémantique latente pour créer un modèle du domaine à l'aide des textes non libellés. En fait, ceux-ci sont utilisés pour construire la matrice réduite, cette étape étant complètement indépendante des étiquettes de catégories. Ensuite, on classe les nouveaux documents en les représentant dans ce nouvel espace vectoriel enrichi puis en les comparant aux documents d'entraînement. L'expérience a donné des résultats supérieurs à l'utilisation de l'indexation sémantique latente classique et une observation intéressante est que les connaissances extérieures se sont avérées plus utiles quand l'ensemble d'entraînement était réduit.

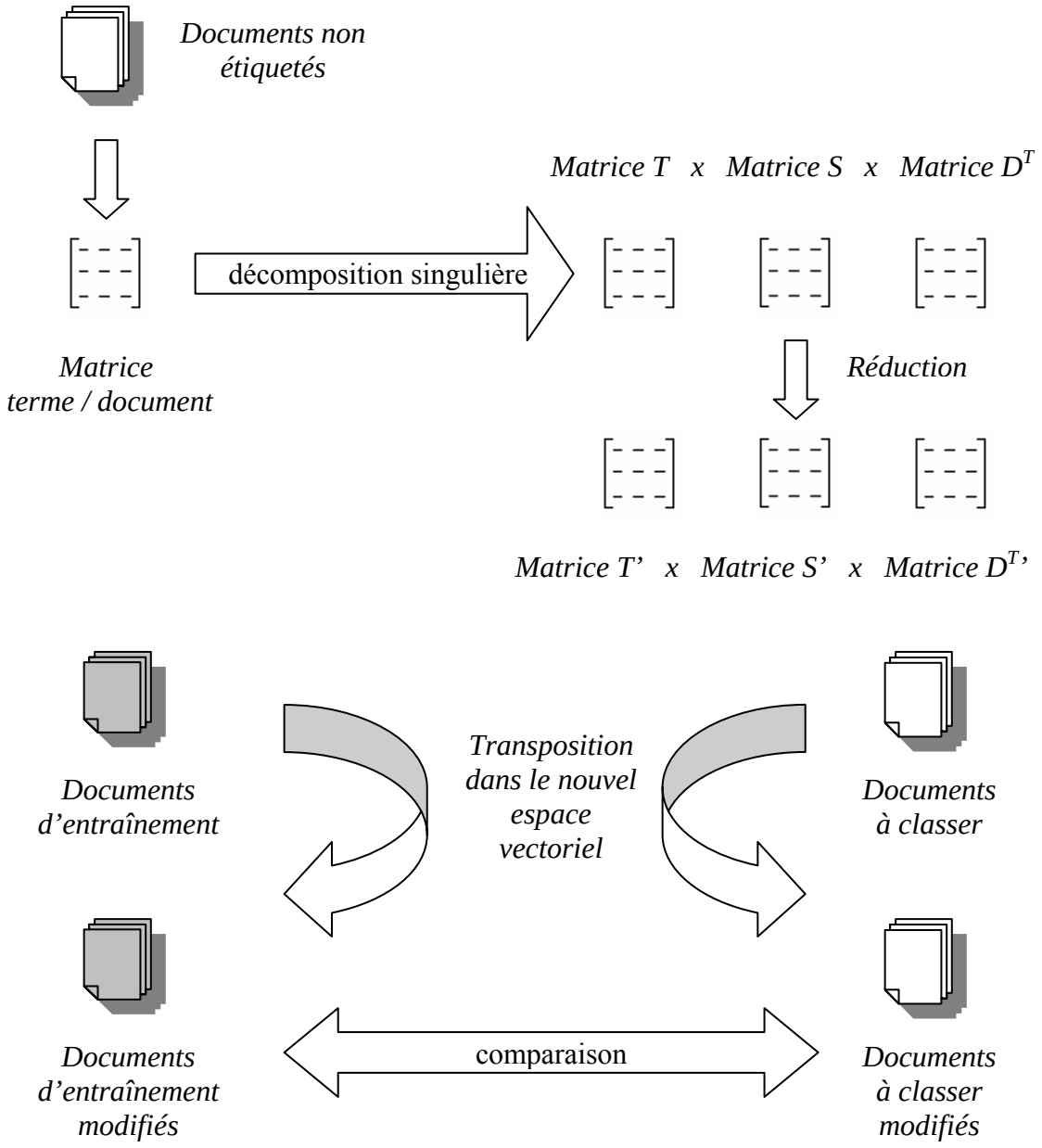

Figure 3.3 - Indexation sémantique latente et catégorisation automatique de textes

Enfin, voici une façon supplémentaire de profiter de documents non libellés. Au chapitre 2, on a vu comment fonctionnait l'algorithme des k-voisins les plus proches : il compare un texte à classer avec les documents d'entraînement, calcule leur similarité et prend en considération les voisins les plus semblables. Dans un

tel contexte, [ZH02] ont proposé d'utiliser les textes non étiquetés en tant qu'intermédiaires entre les textes d'entraînement et les nouveaux textes à classer (voir figure 3.4). Autrement dit, un exemple d'entraînement va être placé parmi les voisins les plus proches d'un texte à classer si ce dernier est jugé similaire à un texte non libellé et que ce texte non libellé est jugé similaire à l'exemple d'entraînement. C'est une approche dite de second ordre : on procède en quelque sorte par transitivité, car les documents ne sont pas comparés directement, mais en passant par des intermédiaires.

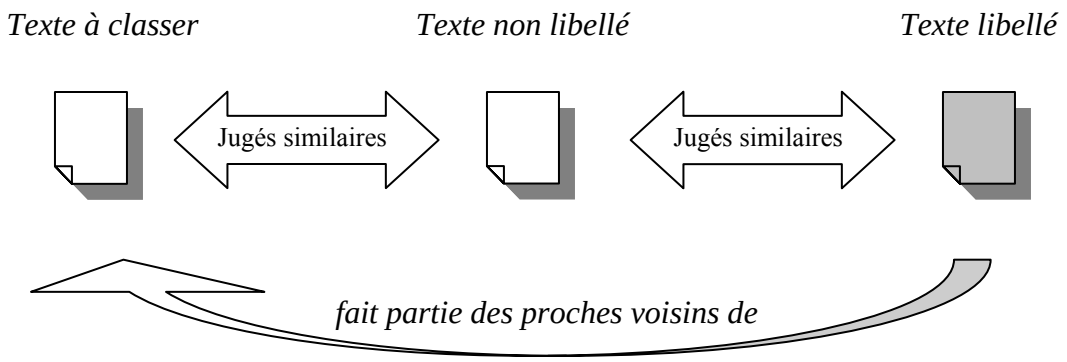

Figure 3.4 - Utilisation de documents non libellés avec l'algorithme kNN

Résumé

Ce chapitre a donc mis en évidence le coût élevé associé à la construction d'un ensemble d'entraînement ainsi que la nécessité d'un tel ensemble. Il en résulte que des projets de développement d'applications de catégorisation automatique de textes peuvent voir leur faisabilité compromise. On a pu voir que face au problème, diverses solutions ont été proposées en vue de tirer profit de connaissances extérieures à cet ensemble d'entraînement. Deux principales sources de connaissances ont été exposées : les lexiques et les documents non libellés. Nos

travaux se sont penchés sur la deuxième option, avec l'intention d'étudier dans ces textes non étiquetés une forme d'association entre les mots : la cooccurrence. Le chapitre suivant se consacre à la présentation de ce concept.

Chapitre 4

Cooccurrence de mots

Au chapitre précédent, il a été question de l'utilisation de documents non libellés pour assister la catégorisation automatique de textes. L'une des motivations invoquées était le fait que bien que ceux-ci ne soient associés à aucune étiquette de catégorie, ils contiennent une certaine quantité d'information à propos des liens qu'entretiennent entre eux les mots du lexique. Mais qu'en est-il de ces liens ? Ceux-ci peuvent prendre plusieurs formes, qu'on pense aux relations de synonymie, d'antonymie, d'hyperonymie, de méronymie, etc. Ces types de liens représentent des associations relativement fortes entre les mots. Un autre genre d'association, de nature plus faible, mais quand même très pertinente, est la cooccurrence. Le présent chapitre a pour objectif de définir cette idée dans le contexte de nos travaux, ce qui sera fait en présentant également la notion de collocation et la distinction entre les deux concepts. En deuxième partie, quelques façons de mesurer numériquement le degré de cooccurrence seront présentées, des techniques qui sont applicables à différentes situations. Ce n'est qu'au chapitre suivant que nous présenterons comment nous proposons d'utiliser cette information sur la cooccurrence entre les mots dans le contexte de la catégorisation automatique de textes.

4.1 Définition

La notion de cooccurrence fait référence au phénomène général par lequel des mots sont susceptibles d'être utilisés dans un même contexte [MS99]. Autrement dit, on considère qu'il y a cooccurrence lorsque la présence d'un mot dans un texte donne une indication sur la présence d'un autre mot. Pour mieux comprendre cette définition, pensons à un exemple simple comme «*avion*» et «*aéroport*», deux mots qui, selon toute vraisemblance, sont utilisés la plupart du temps dans un contexte commun, celui de l'aviation. Souvent, si on retrouve un de ces mots dans un texte, on peut prévoir que le deuxième va aussi être présent. Ces deux mots ne sont ni synonymes ni antonymes. Le sens de l'un n'est pas inclus dans le sens de l'autre (donc pas d'hyperonymie) et l'un n'est pas une partie de l'autre (donc pas de méronymie). Pourtant, il est évident que ces deux mots partagent quelque chose. On choisit donc de dire qu'ils sont cooccurrents. D'autres exemples sont très faciles à trouver : «*médecin*» et «*infirmière*», «*étudiant*» et «*professeur*», etc. Bien que dans ce mémoire, cette connaissance au sujet des mots soit mise à profit pour faire de la catégorisation automatique de textes, elle pourrait éventuellement être exploitée au sein d'autres applications comme la traduction automatique de textes ainsi que la recherche d'information, pour ne nommer que ceux là. Tout ce qui a trait de près ou de loin à la sémantique des mots pourrait profiter du type de connaissance que la cooccurrence véhicule.

Pour peut-être mieux comprendre en quoi consiste la cooccurrence, elle sera comparée à un autre genre d'association entre les mots, la collocation. Une distinction mérite d'être faite entre les deux notions, puisqu'elles partagent certains points en commun, mais elles diffèrent tout de même. Une collocation est plutôt

définie dans [Cho88] comme étant : *«une séquence de deux ou plusieurs mots consécutifs qui sont caractérisés par une unité syntaxique et sémantique et dont la signification ou la connotation ne peut pas être dérivée de la signification de chacune de ses composantes individuelles»*. Typiquement, c'est un groupe de deux ou plusieurs mots qui correspondent à une certaine façon conventionnelle de dire les choses. Les groupes nominaux (ex. : *«armes de destruction massive»*), les groupes verbaux (ex. : *«tenir compte de»*) ou autres expressions courantes (ex. : *«riches et célèbres»*) sont considérés comme étant des collocations. C'est le cas de patrons particuliers de mots, qui sont subtiles et difficiles à expliquer, qui reflètent un choix spécifique connu des gens dont c'est la langue maternelle (ex. : on utilise l'expression *«intelligence artificielle»* et non *«intelligence synthétique»*). Une des caractéristiques importantes des collocations est leur compositionnalité limitée. Une expression en langage naturel est compositionnelle quand le sens global de l'expression peut être prédit par le sens de chacune de ses parties (ex. : *«maison rouge»*). En ce qui concerne les collocations, elles ne sont pas complètement compositionnelles parce qu'un élément sémantique est ajouté à leur combinaison. Les exemples les plus concrets de non compositionnalité demeurent bien entendu les expressions idiomatiques, où souvent le sens véhiculé par l'expression est seulement lié indirectement aux sens véhiculés par les mots individuels la composant, par le biais de références historiques ou sociales. Pensons aux expressions *«être tiré à quatre épingles»* ou *«tomber dans les pommes»*, qui n'ont pas pour but de faire référence à des *«épingles»* ou à des *«pommes»*, au contraire. On a mentionné plus haut que les collocations présentaient une compositionnalité limitée, dans le sens où la modification de la sémantique, provenant de la combinaison de mots, peut être plus subtile que dans les exemples précédents (ex. : *«vin blanc»*). En plus de la non compositionnalité, d'autres indices peuvent indiquer la présence d'une collocation. Lorsqu'on ne peut substituer un mot par un autre qui est presque un

synonyme (ex. : *«vin jaune»* au lieu de *«vin blanc»*) ou qu'on ne peut modifier l'expression en ajoutant des mots (ex. : *«tomber dans les petites pommes»*) ou en appliquant des transformations grammaticales. Également, si l'on ne peut traduire mot à mot une expression, c'est probablement parce qu'il s'agit d'une collocation (ex. : *«prendre une décision»* et *«make a decision»*). Encore une fois, de nombreuses applications peuvent tirer profit de l'information sur les collocations : la génération de langage (pour que le texte généré soit naturel et que les erreurs relatives au choix des mots soient évitées), l'analyse syntaxique (en cas d'ambiguïté, pour donner la préférence aux séquences ayant une collocation naturelle), etc.

Avec cette définition de la collocation en perspective, on peut voir en quoi se distingue la cooccurrence : on ne considère pas que les mots concernés doivent apparaître dans une unité grammaticale commune ou dans un ordre particulier. Deux mots considérés cooccurrents sont associés fortement l'un à l'autre, mais le contexte commun dans lequel ils apparaissent est plus large que dans le cas des collocations : il peut s'agir d'un paragraphe, d'un texte ou d'une collection de textes selon l'application, selon l'usage prévu pour cette information.

4.2 Mesures d'association

Dans le cadre de l'approche statistique du traitement de la langue naturelle, plusieurs façons d'attribuer un score d'association à une paire de mots ont été élaborées. Parmi ces mesures, certaines se basent sur de solides fondements théoriques, alors que d'autres relèvent plutôt du domaine de l'heuristique. Certaines sont directement tirées de la discipline des statistiques alors que d'autres sont nées dans le domaine de la recherche d'information. Nous avons choisi d'étudier plus en profondeur trois de ces mesures. Elles seront exposées en détail

dans la présente section. Le choix s'est arrêté sur celles-ci en fonction de leur popularité et de leur bonne performance dans certaines applications. Ces trois mesures sont celles qui ont fait l'objet d'expérimentations au cours de nos travaux.

Avant tout, une remarque importante à faire est que, en général, les scores d'association obtenus à partir de mesures différentes ne peuvent pas être comparés directement. Dans plusieurs cas aussi, la valeur numérique en tant que telle ne peut pas être interprétée. Typiquement, leur utilisation a plutôt pour but d'ordonnancer des paires de mots candidats en fonction de leur degré d'association. Tous les traitements et comparaisons effectués par la suite se basent sur des listes de n meilleurs mots ou sur des seuils, et la valeur numérique en tant que telle n'est plus considérée [Eve04].

Généralement, les mesures d'association peuvent s'appliquer autant aux collocations qu'aux cooccurrences, à peu de choses près. Pour chacune des mesures qui seront présentées ici (et pour presque toutes les autres), le calcul se base sur les données du tableau 4.1, qui contient les fréquences d'apparition des paires de mots. C'est à partir des quatre valeurs présentes dans cette table que l'on va calculer le degré d'association entre deux mots : (a) le nombre de fois où les deux mots apparaissent ensemble, (b) le nombre de fois où le premier mot est présent sans que le deuxième ne le soit, (c) le nombre de fois où le deuxième mot est présent sans que le premier ne le soit, et (d) le nombre de fois où aucun des deux mots n'apparaît.

	Mot 2 présent	Mot 2 absent
Mot 1 présent	*a*	*b*
Mot 1 absent	*c*	*d*

Tableau 4.1 - Table de contingence pour mesurer le degré d'association entre deux mots

C'est l'interprétation que l'on donne aux valeurs de cette table qui spécifie le phénomène mesuré. Le contexte commun dans lequel doivent figurer deux mots pour qu'on considère qu'ils apparaissent ensemble doit être défini. Si on veut mesurer des collocations, on va analyser la présence des mots à l'intérieur d'un même bigramme ou d'un même trigramme, et l'ordre d'apparition va revêtir une certaine importance. Si, par contre, on veut mesurer des cooccurrences, on va analyser la présence des mots à l'intérieur d'une même phrase, d'un même paragraphe, d'un même texte ou d'une même collection de textes, selon le cas. Dans le contexte de nos travaux, la cooccurrence entre deux mots est vue comme étant la tendance de deux mots à apparaître dans un même document. La table de contingence à la base de nos calculs recense donc le nombre de documents contenant ou non les deux mots dont on étudie la cooccurrence. Dans la suite du mémoire, c'est cette interprétation qui est associée à la cooccurrence, mais on peut garder en tête que chacune des mesures d'association discutées se transpose très facilement dans un autre contexte.

4.2.1 Test du χ^2

Une première façon d'assigner une valeur numérique au degré de cooccurrence entre deux mots est le test du χ^2 [MS99]. Cette mesure a été imaginée dans le cadre du problème classique en statistique qu'est la validation d'hypothèses. Voici le problème à surmonter : le fait que deux mots présentent une fréquence élevée d'occurrence ensemble ne reflète pas nécessairement une association entre ceux-ci.

Il peut s'agir d'un phénomène aléatoire. La question est alors de déterminer si la cooccurrence se produit plus souvent que par hasard. La façon classique de procéder commence par la formulation d'une hypothèse nulle H_0 selon laquelle il n'y aurait pas du tout d'association entre les mots. On calcule la probabilité qu'une cooccurrence se produise si H_0 était vraie et on rejette cette hypothèse si la probabilité s'avère trop faible. Sinon, on retient l'hypothèse d'indépendance. L'idée générale de la mesure du χ^2 est de comparer les fréquences observées dans un corpus avec les fréquences attendues s'il y avait indépendance. Si la différence entre les deux est grande, on peut rejeter l'hypothèse d'indépendance. Basée sur le tableau 4.1, l'équation qui traduit cette notion est la suivante :

$$\chi^2 = \sum_{ij} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

où

- i couvre les rangées du tableau 4.1
- j couvre les colonnes du tableau 4.1
- O_{ij} est la valeur observée pour la cellule (i, j)
- E_{ij} est la valeur attendue pour la cellule (i, j)

La statistique fait donc la somme des différences au carré entre les valeurs observées et attendues pour chaque cellule de la table, chaque différence étant normalisée par la valeur attendue. On peut la voir comme une sorte d'erreur quadratique moyenne. Les valeurs observées sont tout simplement calculées à partir des textes du corpus. Quant aux valeurs attendues, elles sont calculées à l'aide des probabilités marginales, c'est-à-dire à partir des totaux pour les lignes et pour les colonnes de la table, convertis en proportions. Par exemple, la probabilité attendue E_{11}, qui est la probabilité que le mot 1 et le mot 2 soient tous deux présents dans un texte, se calcule de la façon suivante : on multiplie la probabilité marginale que le mot 1 apparaisse avec la probabilité marginale que le mot 2

apparaisse, ainsi qu'avec le nombre de documents. Sachant que N représente le nombre total de documents, on peut faire les calculs suivants :

- Probabilité que le mot 1 apparaisse : $P_1 = (a + b) / N$
- Probabilité que le mot 2 apparaisse : $P_2 = (a + c) / N$
- Probabilité que le mot 1 et le mot 2 apparaissent tous les deux : $E_{11} = P_1 \cdot P_2 \cdot N$

Après quelques manipulations algébriques et simplifications, on arrive à l'expression suivante qui nous permet d'obtenir la valeur de χ^2 :

$$\chi^2 = \frac{N(ad - bc)^2}{(a+b)(a+c)(b+d)(c+d)}$$

Quand la taille de l'échantillon est suffisamment grande, le test du χ^2 suit une distribution χ^2. On consulte une table de cette distribution pour connaître le degré de confiance associé à la valeur obtenue par le calcul. Par exemple, en haut de la valeur critique $\chi^2 = 3.841$, on peut rejeter l'hypothèse d'indépendance avec une confiance de 95%. L'avantage principal du test du χ^2 par rapport au t-test[14] réside dans le fait que ce dernier présuppose que les données suivent une distribution normale, alors que le test du χ^2 ne fait pas de présupposition du genre. Par contre, si les valeurs attendues dans chaque cellule sont très petites, il y a des risques que la mesure retourne des résultats imprécis.

4.2.2 Ratio de vraisemblance

Dans le même ordre d'idées que le test du χ^2, une autre façon d'effectuer des tests d'hypothèses est le ratio de vraisemblance («*likelihood ratio*»). Ce test est réputé

[14] Le t-test, développé par Student, est un autre test d'hypothèses très utilisé en statistiques [MS99].

pour être plus approprié avec les données clairsemées. Également, il présente un autre avantage : la valeur numérique qui est calculée peut être plus facilement interprétée qu'avec le test du χ^2. En effet, il s'agit d'un nombre reflétant dans quelle proportion une hypothèse est plus probable qu'une autre. On considère les deux hypothèses présentées plus loin et le ratio indique laquelle est la plus probable. L'hypothèse 1 reflète le cas où il y aurait indépendance complète entre les deux mots : les probabilités conditionnelles que le mot 2 soit présent connaissant respectivement la présence ou l'absence du mot 1 sont les mêmes. Par contre, l'hypothèse 2 voulant que les deux mots soient dépendants implique que ces probabilités conditionnelles soient différentes.

Hypothèse 1 : P(mot 2 | mot 1 présent) = p = P(mot 2 | mot 1 absent) (indépendance)

Hypothèse 2 : P(mot 2 | mot 1 présent) = p1 ≠ p2 = P(mot 2 | mot 1 absent) (dépendance)

Partant de ces hypothèses, on désire évaluer la vraisemblance de chacune d'elles, étant donné les fréquences observées dans le corpus concernant l'apparition de mots, pour ensuite faire le rapport de ces valeurs de vraisemblance. Le tout se fait en supposant une distribution binomiale.

On fait d'abord un estimé du maximum de vraisemblance pour les probabilités suivantes, à partir des données de notre table de contingence initiale :

- $p = (a + c) / N$
- $p_1 = a / (a + b)$
- $p_2 = c / (c + d)$

Ensuite, on considère l'expression *L(H)* comme étant la vraisemblance d'une hypothèse *H*. Le logarithme du ratio de vraisemblance des deux hypothèses en question se calcule donc comme suit :

$$\log \lambda = \log \frac{L(H1)}{L(H2)} = \log \frac{B(a, a+b, p) \cdot B(c, c+d, p)}{B(a, a+b, p1) \cdot B(c, c+d, p2)}$$

où $B(k, n, x) = \binom{n}{k} x^k (1 - x)^{n-k}$ (distribution binomiale)

En effet, la vraisemblance de H1, par exemple, correspond à la probabilité d'avoir *a* fois le mot 2 présent quand le mot 1 est présent (*a* + *b* fois) multipliée par la probabilité d'avoir *c* fois le mot 2 présent quand le mot 1 est absent (*c* + *d* fois). Un raisonnement similaire peut être fait pour la deuxième hypothèse.

Une simplification permet d'arriver à l'expression suivante :

$$\log \lambda = \log L(a, a + b, p) + \log L(c, c + d, p) - \log L(a, a + b, p_1) - \log L(c, c + d, p_2)$$

où $L(k, n, x) = x^k (1 - x)^{n-k}$

Comme il a déjà été mentionné, le score obtenu par le rapport des vraisemblances peut être interprété comme tel. Mais une propriété intéressante de cette mesure est que la quantité -2 log λ suit une distribution χ^2 asymptotiquement. Donc, il est possible de faire des tests d'hypothèses en consultant la table de distribution appropriée.

4.2.3 Information mutuelle

Une troisième mesure d'association possible est l'information mutuelle (*«pointwise mutual information»* ou PMI). C'est une mesure qui est née de motivations provenant du domaine de la théorie de l'information. Elle mesure la quantité d'information apportée par la présence d'un mot au sujet de la présence d'un autre mot. L'équation suivante permet de calculer cette mesure et met en jeu, au numérateur, la proportion des documents où figurent les deux mots puis, au dénominateur, les proportions respectives des documents où figure un seul des deux mots.

$$\text{I (mot 1, mot 2)} = \log_2 \frac{P(mot1 \,\&\, mot2)}{P(mot1)P(mot2)}$$

Intuitivement, il est facile de voir que plus deux mots ont tendance à apparaître ensemble, plus le score sera élevé. En effet, si les deux mots étaient totalement indépendants, le rapport des probabilités serait de 1, donnant donc une valeur finale de 0. À l'opposé, si les deux mots ont une forte tendance à apparaître ensemble, alors la probabilité au numérateur dépassera l'autre, faisant croître le score d'information mutuelle. Toujours à partir de la table de contingence et après certaines manipulations, on arrive à la façon simplifiée de calculer la PMI :

$$\text{I (mot 1, mot 2)} = \log_2 \frac{Na}{(a+c)(a+b)}$$

L'information mutuelle est considérée comme étant une bonne mesure de l'indépendance entre deux mots. Sa valeur devient nulle si les deux mots sont parfaitement indépendants. Mais pour des paires de mots parfaitement dépendants (qui apparaissent toujours ensemble), la valeur de l'information mutuelle augmente plus ceux-ci sont rares, la raison étant que les probabilités au

dénominateur diminuent. Ce phénomène est en fait un inconvénient lié à l'utilisation de cette mesure puisqu'il devient plus difficile de comparer les scores obtenus par des mots fréquents avec ceux obtenus par des mots rares.

Cette façon de calculer l'association entre des mots a entre autres été exploitée par [Tur01] au sein d'un algorithme d'apprentissage ayant pour but de reconnaître des synonymes. Le problème à résoudre était le suivant : devant un mot donné et une liste de mots alternatifs, il s'agissait de trouver parmi cette liste le mot dont le sens se rapprochait le plus du mot d'origine. La solution proposée consistait à calculer le degré de cooccurrence entre le mot d'origine et chacun des choix de réponse, puis à choisir comme meilleur synonyme le candidat ayant obtenu le plus haut score de cooccurrence. L'intuition à la base de cette façon de faire était qu'un mot est caractérisé par les mots qui l'entourent. Pour recueillir les statistiques nécessaires au calcul des scores, il a été proposé de tirer profit d'un corpus de textes accessible à tous : le Web. Pour chaque paire de mots, il suffit d'exécuter deux requêtes sur un moteur de recherche (en l'occurrence Altavista) et de faire un calcul à partir du nombre de documents retournés par chacune d'elles.

$$\text{Score(choix)} = \log \frac{P(problème\&choix)}{P(problème)P(choix)}$$

où

- *choix* représente le choix de réponse considéré
- *problème* représente le mot dont on cherche le synonyme

Sachant que le but est de maximiser ce score, on peut laisser tomber le log et P(problème) qui restent constants. Puis, en utilisant de la façon la plus simple un moteur de recherche pour calculer les valeurs restantes, on obtient la formule suivante :

$$\text{Score(choix)} = \frac{hits(problèmeANDchoix)}{hits(choix)}$$

où

- *hits* représente le nombre de pages retournées par la requête entre parenthèses

On a aussi raffiné cette équation en utilisant des opérateurs comme NEAR, en tentant d'éviter de favoriser les antonymes et en utilisant des mots du contexte, lorsque le problème le permettait.

Évalué à l'aide de questions sur les synonymes tirées d'un test d'anglais langue seconde, cette méthode a généré des résultats supérieurs à 70%, ce qui dépasse la performance moyenne des étudiants qui passent ce test, qui est de 65%. Cela dépasse aussi la performance de l'indexation sémantique latente, qui, lors d'une expérience précédente, avait généré des résultats 10% inférieurs à ceux obtenus par l'approche basée sur l'information mutuelle. Cette dernière tire probablement sa force de la taille importante du corpus de textes utilisé, soit l'ensemble des pages Web indexées par Altavista. Comme ces données sont accessibles assez facilement et rapidement, pourquoi ne pas en profiter ? On peut entrevoir d'autres applications à cette technique comme l'expansion de requêtes en recherche d'information ou l'extraction de mots-clés.

Résumé

Ce chapitre aura permis de préciser en quoi consiste la notion de cooccurrence entre des mots, le phénomène général par lequel ceux-ci sont susceptibles d'être utilisés dans un même contexte. Il nous a également éclairé sur l'évaluation quantitative de la force de cette relation. Après une introduction générale sur les

différentes mesures d'association, les mesures particulières que sont le test du χ^2, le ratio de vraisemblance et l'information mutuelle ont été présentées. Rappelons que l'objectif premier de ce mémoire est de proposer une façon pour un classificateur de textes de tirer avantage de l'information sur la cooccurrence de mots provenant de documents non étiquetés. Le chapitre 3 ayant introduit l'idée d'utiliser des documents non étiquetés, et la notion de cooccurrence venant d'être abordée, tout est en place pour que le chapitre suivant puisse énoncer clairement notre proposition.

Chapitre 5

Étude de cooccurrence de mots dans des textes non libellés pour assister la catégorisation automatique de textes

Plusieurs thèmes gravitant autour de la catégorisation automatique de textes ont été abordés jusqu'à maintenant dans ce mémoire. Il a été question aux chapitres 1 et 2 des particularités du problème en tant que tel. On a vu en quoi consistait la catégorisation automatique de textes, quels traitements elle impliquait et de quelle façon on réussissait jusqu'à présent à la résoudre. Au chapitre 3, on a jeté un regard sur comment certains chercheurs ont réussi à tirer profit de documents non classés pour enrichir l'entraînement de classificateurs, afin de lui apporter des connaissances supplémentaires permettant d'obtenir de meilleurs résultats de classification. Parallèlement, le chapitre 4 a permis de faire la lumière sur la notion de cooccurrence de mots et sur différentes façons de la mesurer à partir d'un ensemble de textes. Le présent chapitre a maintenant pour objectif de présenter de façon détaillée l'approche nouvelle que nos travaux proposent. Celle-ci, combinant en quelque sorte les idées des chapitres 3 et 4, vise à enrichir un classificateur automatique de textes à l'aide d'information sur la cooccurrence de mots étudiée dans un corpus de textes non étiquetés. Avant toute chose, la problématique que

nous avons ciblée sera exposée. Puis, une description des détails de la solution suggérée suivra.

5.1 Problématique

À la base, la problématique dont ce mémoire traite est celle de la constitution d'un ensemble d'entraînement. Pour permettre à un classificateur de pouvoir travailler de façon automatique, il est essentiel de le soumettre en premier lieu à une phase d'entraînement au cours de laquelle il va observer des documents classés manuellement pour pouvoir ensuite apprendre les caractéristiques de chaque catégorie. Pour s'assurer que l'apprentissage soit vraiment efficace, il faut fournir au système une banque assez importante de ces documents déjà libellés. Le problème est que la création d'un tel ensemble d'entraînement par des humains est coûteuse.

Au premier abord, il peut sembler évident que plus l'ensemble d'entraînement disponible contient d'exemples, plus le classificateur résultant de l'apprentissage peut réaliser sa tâche convenablement. Les figures 5.1, 5.2 et 5.3 permettent de constater cette influence très forte du nombre de documents d'entraînement sur le score de micro-F1 obtenu par un classificateur sur différents corpus de textes[15].

[15] Des détails sur ces corpus se retrouvent au chapitre suivant.

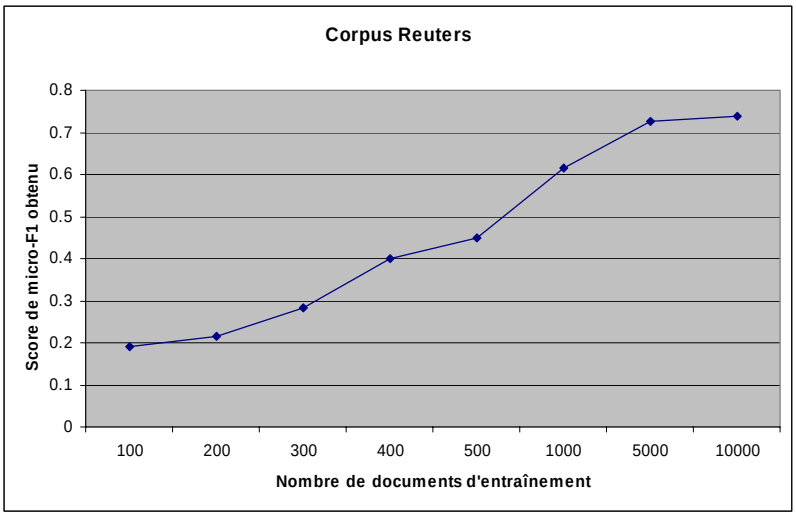

Figure 5.1 - Influence de la taille du corpus d'entraînement sur les performances d'un classificateur (corpus RCV1)

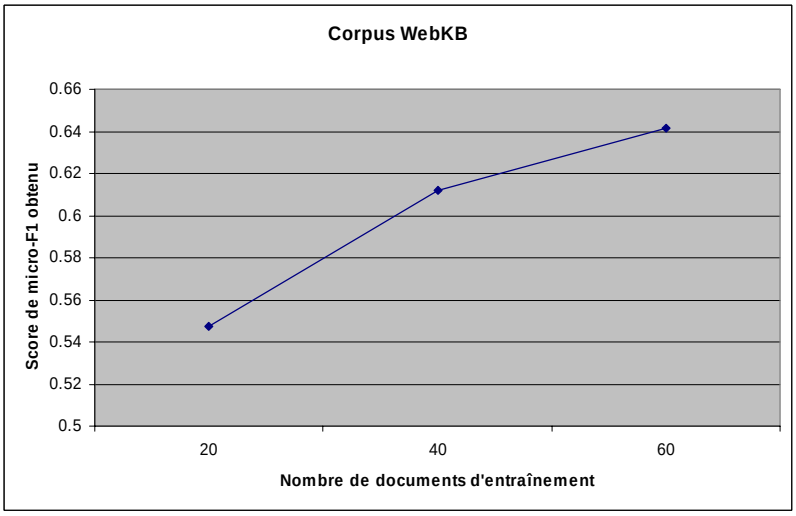

Figure 5.2 - Influence de la taille du corpus d'entraînement sur les performances d'un classificateur (corpus WebKB)

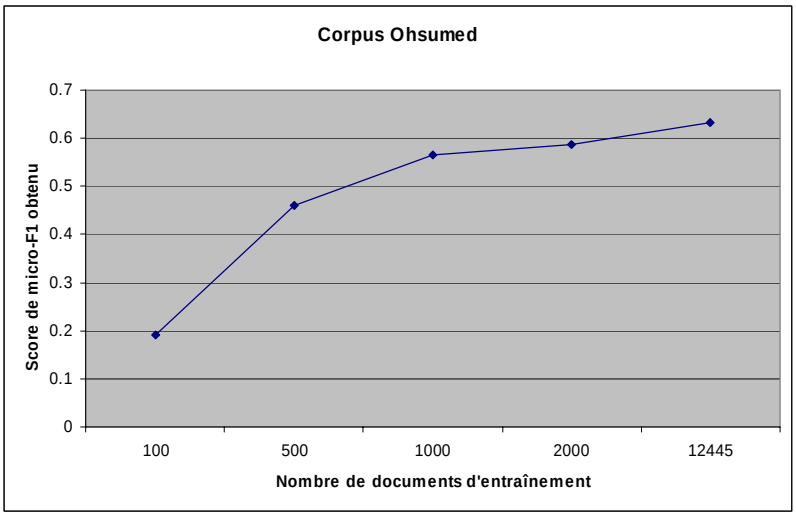

Figure 5.3 - Influence de la taille du corpus d'entraînement sur les performances d'un classificateur (corpus Ohsumed)

Cependant, il est intéressant de tenter de comprendre plus en profondeur ce phénomène. Pourquoi le fait de restreindre l'ensemble d'entraînement d'un classificateur entraîne une baisse de son efficacité à bien classer de nouveaux documents ? Pourquoi le fait de réduire le nombre d'exemples sur lesquels il s'entraîne fait chuter son habileté à réaliser la tâche qu'on attend de lui ?

L'idée derrière l'apprentissage automatique supervisé est qu'en observant les exemples d'entraînement, la machine tente d'établir des relations entre les caractéristiques des instances et les sorties attendues. Surtout, elle tente de généraliser de manière à isoler seulement les caractéristiques vraiment utiles pour identifier la bonne sortie. L'objectif est de mettre de côté les caractéristiques qui pourraient survenir de façon accidentelle et isolée dans seulement quelques exemples. En catégorisation de textes, le classificateur a donc à établir des relations entre les mots des documents et leur catégorie d'appartenance. L'ensemble d'entraînement doit être suffisamment grand pour être représentatif de la majorité des textes susceptibles d'appartenir aux catégories. Si le nombre d'exemples est trop petit, le classificateur va éventuellement apprendre des attributs spécifiques à

ceux-ci et ne sera pas en mesure de procéder à une bonne généralisation. Une autre façon de voir l'apprentissage est de l'associer à une estimation d'un espace vectoriel à partir d'une portion de cet espace, soit les exemples d'entraînement. Plus le classificateur a accès à une grande portion de l'espace pour appuyer son estimation, plus celle-ci est susceptible d'être juste.

Cette explication s'applique à l'apprentissage automatique en général. Si l'on examine de plus près la tâche de catégorisation de textes, une constatation qui peut être faite est que le nombre de documents auxquels un classificateur a accès pendant sa phase d'entraînement influence grandement la richesse du vocabulaire qu'il reconnaît, c'est-à-dire la taille de ce vocabulaire. Lorsqu'il a été question au premier chapitre de la représentation des documents et du modèle vectoriel, on a pu constater que le nombre d'attributs qui composent chaque vecteur représentant un document textuel est en fait le nombre de mots différents rencontrés parmi l'ensemble des exemples d'entraînement (moins ceux qui ont été rejetés par le processus de sélection d'attributs). Voilà ce à quoi correspond le vocabulaire du classificateur. Le tableau 5.1 démontre qu'effectivement, ce vocabulaire croît avec le nombre d'exemples d'entraînement.

Reuters	
Documents d'entraînement	Taille du vocabulaire
100	1086
200	1855
300	2320
400	2891
500	3139
1000	4704
5000	11791
10000	17843

WebKB	
Documents d'entraînement	Taille du vocabulaire
20	177
40	337
60	539

Ohsumed	
Documents d'entraînement	Taille du vocabulaire
100	562
500	1998
1000	3165
2000	4545
12445	10812

Tableau 5.1 - Influence de la taille de l'ensemble d'entraînement sur le vocabulaire reconnu par un classificateur

Une fois que l'apprentissage est complété et qu'un classificateur est appelé à traiter un nouveau document, ce dernier se retrouve associé à un vecteur aux mêmes attributs que les vecteurs représentant les documents d'entraînement. Tous les mots ne faisant pas partie du vocabulaire du classificateur sont complètement ignorés par le processus de classification. La représentation du document en fait totalement abstraction et ces mots ne participent donc pas du tout à la détermination de la catégorie à laquelle appartient le document. Or, on se rappelle que moins l'ensemble d'entraînement d'un classificateur contient de documents, moins son vocabulaire contient de mots, conséquemment, lorsqu'on lui soumet de nouveaux textes, plus il y a de mots qui lui sont inconnus et dont il ne peut tenir compte pour classifier. Le tableau 5.2 donne un aperçu de cet impact en présentant le nombre de mots survenant dans les ensembles de test (de taille fixe) de nos expérimentations, mais étant absents du vocabulaire reconnu.

Reuters	
Documents d'entraînement	Mots inconnus[16]
100	48286
200	47525
300	47063
400	46493
500	46246
1000	44704
5000	38037
10000	33032

WebKB	
Documents d'entraînement	Mots inconnus[17]
20	35916
40	35756
60	35556

Ohsumed	
Documents d'entraînement	Mots inconnus[18]
100	15486
500	14051
1000	12887
2000	11528
12445	6381

Tableau 5.2 - Influence de la taille de l'ensemble d'entraînement sur le nombre de mots inconnus du classificateur survenant dans des documents à classer

Ce phénomène peut être appuyé par l'analogie suivante. Imaginons un individu francophone qui apprend une langue seconde comme l'anglais. Au fil de son apprentissage, sa connaissance du vocabulaire anglais s'améliorera : il connaîtra de plus en plus de mots anglais (c'est ce qui se produit avec un classificateur auquel on soumet de plus en plus de documents d'entraînement). Si notre individu se trouve en conversation avec un anglophone et qu'il doit comprendre le sens de ce que son interlocuteur lui dit, les seuls mots qui vont l'aider à saisir le message sont évidemment les mots qu'il connaît déjà. Les termes qui lui sont inconnus au moment de la conversation ne lui seront d'aucun secours : il devra se faire une idée du sens du message sur la seule base des mots faisant partie de son vocabulaire. De la même façon, un classificateur devant associer un document à une catégorie prend sa décision uniquement en considérant les termes qu'il a déjà rencontrés en entraînement.

[16] Dans un ensemble de test comptant 10000 documents.
[17] Dans un ensemble de test comptant 3094 documents.
[18] Dans un ensemble de test comptant 3632 documents.

Face à cela, il apparaît donc que dans la situation où un classificateur est entraîné sur un faible nombre de textes libellés manuellement, une grande quantité d'information est mise de côté lors de la classification de nouveaux documents, puisqu'un nombre considérable de mots apparaissant dans ces textes sont inconnus du classificateur et il n'en tient pas compte pour estimer la classe d'appartenance des documents.

C'est à ce niveau que nos travaux se situent. Nous proposons une façon de prendre en considération, au sein de la représentation des nouveaux documents à classer, les mots ne faisant pas partie a priori du vocabulaire reconnu par le classificateur. L'approche suggérée est de les mettre en relation avec d'autres mots qui, pour leur part, font partie de ce vocabulaire. L'exploitation d'une collection importante de textes non libellés rend possible cette mise en correspondance des mots, car comme cette banque de documents peut être facilement acquise, elle peut compter un nombre plus grand de textes. Par le fait même, elle a plus de chances de contenir les mots absents du corpus d'entraînement, qui est restreint. À l'intérieur de cette collection de textes, il est possible d'étudier le comportement mutuel des mots faisant partie ou non du vocabulaire. Dans le cadre de notre projet, la cooccurrence entre ces mots constitue la propriété qui est analysée ainsi. La section suivante introduit les détails de cette approche.

5.2 Description de l'approche proposée

La modification au processus de catégorisation que nous proposons intervient dans les faits au niveau de la représentation des nouveaux textes à classer. Il n'est pas question de modifier l'apprentissage du classificateur, mais plutôt de faire subir un traitement supplémentaire à un nouveau document à classer avant de le

soumettre au classificateur. Le processus de catégorisation d'un nouveau document, présenté au chapitre 1, est toujours valide. L'étape ajoutée par notre approche s'incorpore dans la phase de représentation du document.

Au cours du déroulement habituel d'une application de classification automatique, lorsqu'un nouveau texte doit être classé, une représentation de celui-ci sous forme de vecteur est produite en se basant sur le vocabulaire extrait des documents d'entraînement, puis épuré par la sélection d'attributs. Les mots ne faisant pas partie de ce vocabulaire sont mis de côté. Dans l'approche proposée ici, ces mots inconnus sont comparés aux mots faisant partie du vocabulaire, ou plutôt aux mots d'un sous-ensemble du vocabulaire, comme on le verra plus loin. La cooccurrence des mots inconnus par rapport aux mots du vocabulaire est calculée à l'intérieur d'une collection de textes non libellés. Comme celle-ci est vraisemblablement beaucoup plus grande que l'ensemble d'entraînement, les chances sont élevées pour que les mots absents du vocabulaire des documents d'entraînement apparaissent dans ce corpus. Il devient alors possible d'étudier leur comportement par rapport aux autres mots. À la suite de ces calculs de cooccurrence, on se trouve devant une liste de paires de mots auxquelles sont associées des valeurs de cooccurrence. Pour un mot inconnu donné, on considère la paire ayant obtenu le plus haut score et s'il advient que ce score dépasse un seuil minimal de cooccurrence, c'est alors que l'on modifie la représentation du document à classer. On assume que les deux mots sont très cooccurrents et donc que la présence de l'un est une bonne indication de la présence de l'autre. Par conséquent, l'action proposée est de simuler la présence du mot connu dans les textes contenant le mot inconnu. L'effet de cette action est que ce terme va alors avoir une certaine influence sur la classification de ces textes. Autrement dit, il va agir à titre de «*représentant*» du mot inconnu. Comme il a un certain pouvoir puisqu'il fait partie,

lui, du vocabulaire, il va «*exercer le droit de vote*» du mot inconnu dans la prise de décision sur la catégorisation du document. Étant donné que ces deux mots ont été jugés très cooccurrents, ils ont une certaine propension à apparaître dans les mêmes documents, et, ce faisant, dans les mêmes catégories. C'est ce qui porte à croire que l'impact en sera positif. La figure 5.4 résume bien le processus qui vient d'être décrit.

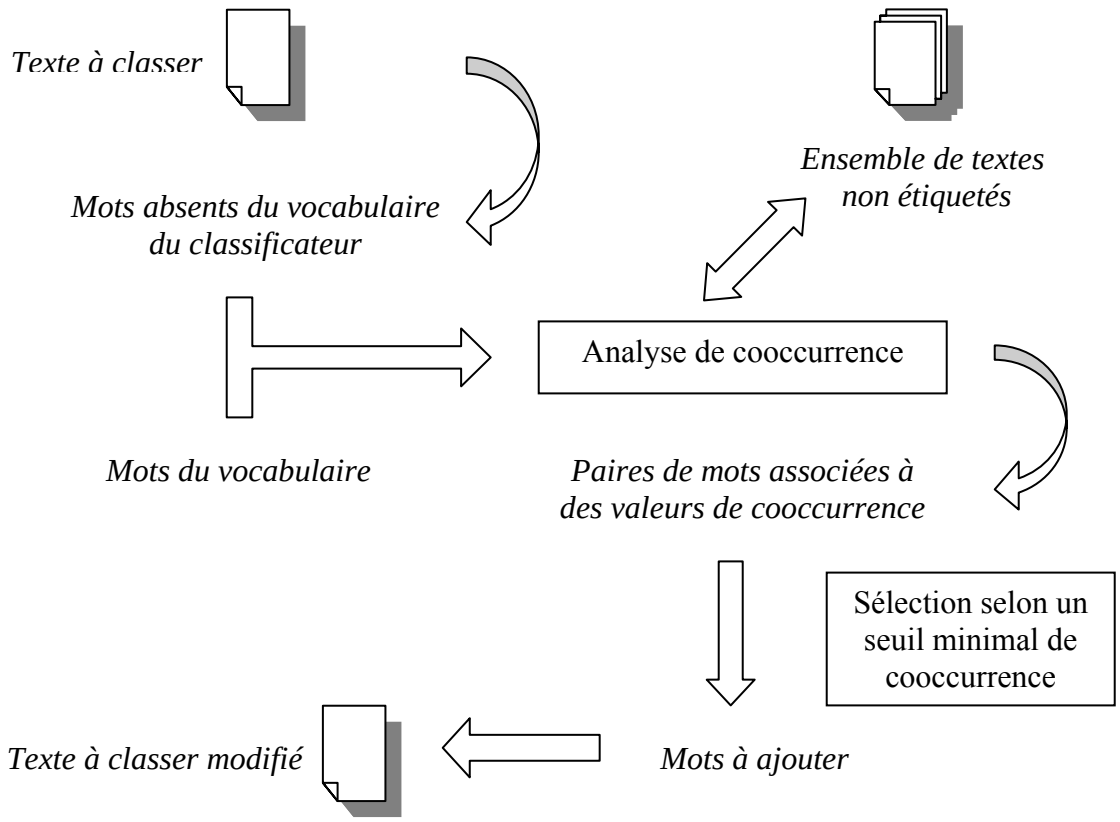

Figure 5.4 - Approche proposée

Un exemple concret étant toujours plus facile à saisir, regardons comment tout cela pourrait se traduire dans la situation suivante :

- Texte à classer : *L'objet d'étude principal de ce **mémoire**, la catégorisation **automatique** de **textes**, est un **problème** qui intéresse les chercheurs depuis relativement longtemps.*

- Mots du vocabulaire : {*apprentissage*, *artificielle*, ***automatique***, *classification*, *intelligence*, ***mémoire***, ***problème***, ***textes***, *vocabulaire*}

- Représentation vectorielle du texte à classer (en supposant une pondération binaire): < 0, 0, **1**, 0, 0, **1**, **1**, **1**, 0 >

On remarque que la présence du mot «*catégorisation*» est complètement ignorée dans la représentation vectorielle du document, parce qu'il ne fait pas partie du vocabulaire. Pourtant, pour déceler le sens de ce texte, il s'agit d'un terme important. En supposant qu'il y ait une catégorie faisant référence au thème de la classification (ou catégorisation) automatique de textes, il est possible que ce document n'y soit pas associé à cause de ce phénomène. Or, si des documents d'entraînement contenaient le mot «*classification*» et que l'analyse de cooccurrence dans les documents non libellés révélait que les mots «*catégorisation*» et «*classification*» apparaissent souvent ensemble, alors la technique que nous proposons simulerait la présence du mot «*classification*» dans le texte à classer, donnant la représentation suivante :

- Nouvelle représentation vectorielle : < 0, 0, 1, **1**, 0, 1, 1, 1, 0 >

Par conséquent, un classificateur kNN, par exemple, pourrait plus facilement détecter sa similitude avec des documents d'entraînement contenant le mot «*classification*» et le placerait dans la même catégorie.

Certains détails restent à préciser à propos de cette approche. Comme il a déjà été signalé, l'étude de cooccurrence ne se fait pas avec l'ensemble des mots du vocabulaire du classificateur. D'abord, à cause de la grande taille de ce

vocabulaire, des temps de calcul considérables seraient engendrés. Mais, de façon plus importante, il faut remarquer que l'objectif de la méthode est d'améliorer la capacité du classificateur à bien accomplir sa tâche. Donc, il ne servirait à rien de remplacer dans un texte un mot inconnu par un mot qui s'avère de toute façon inutile pour discerner l'appartenance à une catégorie. Pour cette raison, il a été décidé de ne considérer qu'un sous-ensemble du vocabulaire pour procéder à l'analyse de cooccurrence. Une façon possible de créer ce sous-ensemble aurait été de sélectionner les meilleurs mots en termes de gain d'information par rapport à l'ensemble des classes, cette mesure étant déjà utilisée pour la sélection d'attributs. Cependant, quelques essais nous ont porté à croire qu'une autre approche serait préférable. Une justification possible est qu'en procédant de la sorte, des mots utiles pour différencier certaines catégories auraient été choisis, mais il est probable que pour d'autres catégories, aucun mot pertinent n'aurait été conservé. L'effet bénéfique escompté de la méthode ne se serait alors pas répercuté sur l'ensemble de la tâche de catégorisation. La solution retenue pour la construction du sous-ensemble de mots est plutôt la suivante : pour chaque catégorie, on retient dans le sous-ensemble un seul mot, celui qui semble lui être le plus lié. À cette fin, le critère de décision est le score obtenu au test du χ^2 entre le mot et la catégorie[19] : pour chaque catégorie, on retient le mot qui obtient le plus haut score, indiquant qu'on peut avec plus de certitude rejeter l'hypothèse d'indépendance et par le fait même conclure que le mot est dépendant de la catégorie. Encore une fois, une question se pose : pourquoi conserver un seul mot par catégorie et pas deux, trois, ou plus ? Certains tests avec un plus grand nombre de mots par catégorie ont réellement été effectués, mais ont donné de moins bons résultats. Intuitivement, il semble que pour qu'un mot inconnu soit avantageusement remplacé par un mot

[19] Tel que présenté au chapitre 4, mais le calcul se fait entre un mot et une catégorie plutôt qu'entre deux mots.

représentatif d'une catégorie et qu'il permette l'obtention de meilleurs résultats de classification, il faut que ce dernier mot soit très lié à la catégorie. Évidemment, c'est de moins en moins le cas lorsqu'on ajoute plusieurs mots par catégorie, ceux-ci étant classés en ordre décroissant de score du χ^2. En somme, l'analyse de cooccurrence se fait par rapport à un ensemble de mots représentatifs des différentes catégories, incluant pour chacune d'elles le mot qui lui est le plus relié selon le test du χ^2. L'annexe A rassemble quelques résultats concernant cette sélection de mots.

Une décision semblable a été prise en ce qui a trait aux mots inconnus. Pour des raisons d'efficacité, plutôt que de considérer chaque mot absent du vocabulaire apparaissant dans un texte à classer, il a été choisi de ne traiter seulement ceux qui apparaissent dans un nombre minimal de documents. Ce faisant, on élimine la plupart des mots bruités, c'est-à-dire les mots mal orthographiés, puisqu'une même erreur n'est pas susceptible de se produire plusieurs fois. De plus, le fait d'ignorer ces mots n'influence pas significativement les résultats globaux du classificateur, car le traitement de ceux-ci touche peu de documents. Mais, la raison principale est qu'en procédant ainsi, on fait chuter considérablement le nombre de paires de mots dont il faut calculer la cooccurrence, ce qui réduit avantageusement le temps requis pour compléter le processus. Le nombre de documents qui a été choisi comme seuil minimal est de 4, valeur fixée empiriquement[20]. Le tableau 5.3 expose l'effet radical de l'application de cette stratégie, en ce qui a trait au nombre de mots à traiter. Il est à noter qu'une élimination semblable est effectuée lors de la phase de sélection d'attributs.

[20] Voir l'annexe B pour plus de détails.

Reuters		
Documents d'entraînement	Mots inconnus[21]	Mots inconnus sélectionnés[22]
100	48286	17527
200	47525	16778
300	47063	16320
400	46493	15757
500	46246	15515
1000	44704	14027
5000	38037	8201
10000	33032	5074

WebKB		
Documents d'entraînement	Mots inconnus	Mots inconnus sélectionnés
20	35916	8754
40	35756	8596
60	35556	8397

Ohsumed		
Documents d'entraînement	Mots inconnus	Mots inconnus sélectionnés
100	15486	5802
500	14051	4375
1000	12887	3258
2000	11528	2068
12445	6381	121

Tableau 5.3 - Réduction du nombre de mots inconnus à traiter en utilisant un seuil minimal d'occurrence

En résumé, l'étude de la cooccurrence ne se fait pas entre tous les mots du vocabulaire et tous les mots inconnus du classificateur. Une sélection est effectuée parmi ces ensembles de termes, comme le rappelle la figure 5.5.

[21] Nombre de mots différents apparaissant dans les documents de test, mais étant inconnus du classificateur.
[22] Nombre de mots inconnus apparaissant dans un nombre minimal de documents de test et étant considérés lors de l'analyse de cooccurrence.

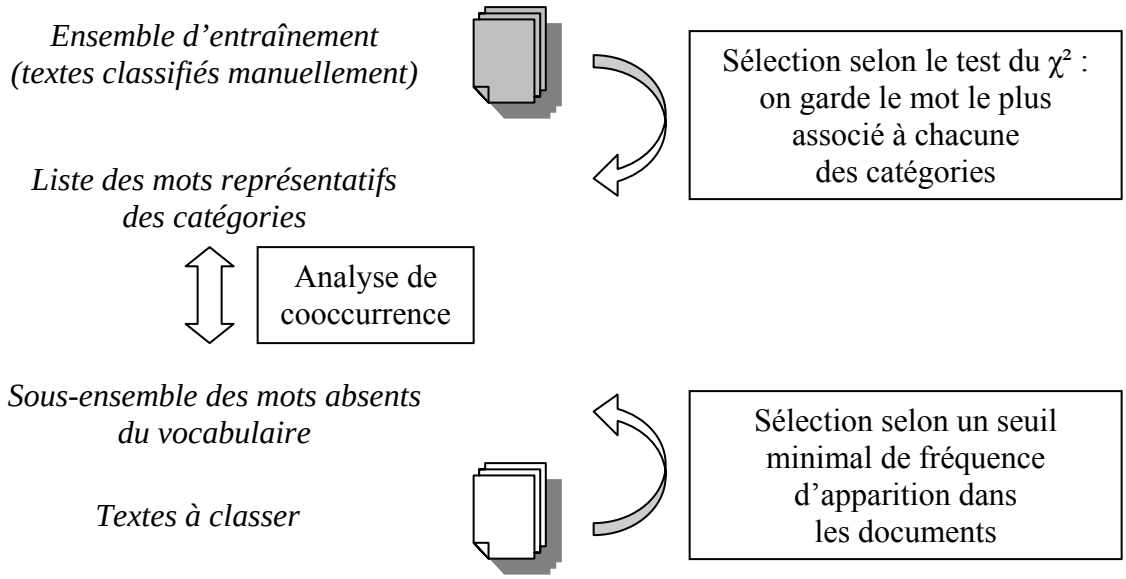

Figure 5.5 - Détermination des mots impliqués dans l'analyse de cooccurrence

Jusqu'à présent, on a vu à quel moment du processus de classification venait s'intégrer la phase d'étude de cooccurrence et quels ensembles de mots étaient impliqués. Il reste à discuter du module de calcul de cooccurrence à annexer à l'application de catégorisation de textes. Mais, essentiellement, son fonctionnement est très simple. Au chapitre précédent, on a pu voir que la plupart des mesures de cooccurrence, dont les trois présentées de façon détaillée, se résument à des équations impliquant quatre valeurs, calculées à partir d'un ensemble de textes. Pour obtenir ces valeurs, tout ce qu'il y a à faire est de s'interroger sur la présence ou l'absence de mots dans des textes. Conséquemment, les traitements impliqués se résument à une phase de séparation des mots des textes[23] de même qu'à une phase de recherche parmi des listes de mots. Au premier abord, la mise en place de ces mécanismes semble très facile. Par contre, pour traiter efficacement et assez rapidement le grand nombre de textes et de mots impliqués, des considérations

[23] Ce qui correspond à la tâche connue en anglais sous le nom de *«tokenization»*.

d'indexation et d'optimisation sont souhaitables. On voit rapidement que ce qui précède s'apparente au travail que fait un moteur de recherche. Dans le cadre de nos travaux, nous avions accès au moteur de recherche *Copernic Enterprise Search*[24], qui permet l'indexation et la recherche de documents dispersés dans les réseaux d'entreprises. Celui-ci nous a donc facilité l'acquisition rapide de statistiques sur nos corpus de textes non libellés, par le biais de la recherche à partir des mots à analyser. L'indexation de nos collections complétée à l'aide de cet outil, il ne restait plus qu'à lui soumettre des requêtes et à consigner le nombre de documents retournés par chacune d'elles. Partant des résultats obtenus, le calcul des différentes mesures de cooccurrence se résumait à combiner les quatre valeurs de la table de contingence selon l'équation appropriée de façon à obtenir un score final de cooccurrence.

La dernière étape du processus consiste à choisir parmi l'ensemble des paires de mots analysées lesquelles vont commander un ajout dans les textes à classer. Il s'agit de juger si le degré de cooccurrence entre deux mots est assez grand pour justifier la simulation de la présence d'un des mots dans un texte, sachant la présence de l'autre mot. Comme il a été relevé au chapitre 4, les différentes mesures de cooccurrence s'avèrent difficiles à interpréter et sont plutôt utilisées pour ordonnancer des paires de mots, ce qui rend ce jugement assez difficile à faire. Plusieurs stratégies différentes peuvent être mises en place. L'idéal serait que des bases théoriques permettent de fixer une valeur numérique au-delà de laquelle le degré de cooccurrence serait considéré suffisant. Cependant, selon les mesures, des difficultés surviennent. Dans le cas de l'information mutuelle, la théorie sous-jacente ne permet tout simplement pas d'y arriver. Dans le cas du test du χ^2 ou du ratio de vraisemblance, bien que des intervalles de confiance soient associés aux

[24] Plus d'informations et téléchargement possible à www.copernic.com.

valeurs numériques obtenues, ceux-ci font référence à la dépendance entre les mots, mais pas directement à l'interprétation qui en est faite dans notre contexte. Par exemple, supposons qu'on obtienne un score supérieur à 3.841 au test du χ^2 entre deux mots, on peut en effet accepter la dépendance de ces mots avec une confiance de 95%. Mais rien n'indique clairement si les deux mots sont assez dépendants pour commander un ajout dans les textes à classer. Il faut se rabattre sur la stratégie qui consiste à fixer un seuil empiriquement, à partir d'expériences passées. Une discussion à ce sujet sera faite au chapitre suivant, à la suite de la présentation des résultats de nos expérimentations en ce sens.

Résumé

Donc, rappelons ce que notre approche propose. Devant un nouveau texte à classer, on considère les mots inconnus du classificateur, c'est-à-dire ceux ne faisant pas partie de son vocabulaire. On mesure leur cooccurrence par rapport aux mots faisant partie du vocabulaire, et ce à l'intérieur d'une collection de textes non étiquetés. Une transformation est finalement appliquée au texte à classer : pour un mot inconnu donné, on simule la présence d'un mot connu qui lui est habituellement très cooccurrent. Le résultat est que les mots connus ajoutés vont jouer un rôle dans le processus de catégorisation, contrairement aux mots inconnus qui sont ignorés. Profitant d'une quantité supérieure d'information, le classificateur est en mesure de mieux performer. Le chapitre suivant présente les améliorations que cette approche permet de réaliser quant à la capacité d'un classificateur à classer correctement.

Chapitre 6

Évaluation de l'approche proposée

Au chapitre précédent, une approche tirant profit de textes non libellés pour assister une tâche de catégorisation automatique de textes a été exposée. De l'information au sujet de la cooccurrence entre les mots du vocabulaire est extraite des textes non libellés en vue de modifier la représentation des documents à soumettre au classificateur. Ce faisant, on souhaite que ce dernier soit en mesure de mieux répondre à sa tâche. Plus précisément, l'effet escompté de l'utilisation de cette nouvelle approche est une amélioration du score de micro-F1 obtenu par le classificateur, en comparaison avec sa performance sans utiliser de documents non étiquetés. Pour valider cette hypothèse, des tests ont été effectués sur différentes collections de textes, à l'aide de différents types de classificateurs et en faisant varier la taille de l'ensemble d'entraînement et d'autres paramètres. Ce chapitre a pour but de spécifier d'abord et avant tout la méthodologie d'évaluation mise en place, puis de présenter les résultats des différentes expérimentations menées et de discuter des interprétations qu'on peut en faire.

6.1 Méthodologie

6.1.1 Corpus de textes

Dans cette section seront présentés les différents corpus de textes qui ont été utilisés au cours de nos expérimentations. Une attention particulière a été portée quant au choix de ces collections. Une de nos préoccupations était de choisir des ensembles de textes facilement accessibles et disponibles, ayant déjà utilisés dans le passé et étant susceptibles de l'être encore dans le futur. Aussi, nous voulions introduire une certaine variété en ce qui a trait aux types de textes considérés et aux types de tâches de classification à traiter. Il a déjà été montré que le type de textes pouvait influencer la performance des algorithmes. De la même façon, le type de tâche de catégorisation (nombre de catégories possibles, nombre de catégories à assigner à chaque document, etc.) peut avoir une certaine influence sur les résultats. Il était alors intéressant de s'assurer que la méthode proposée pouvait être utile dans différentes situations, dans des contextes variés. Un autre aspect assez important était la taille de ces corpus. Pour que nos résultats soient statistiquement significatifs, nos ensembles de test devaient contenir un nombre suffisant de documents.

Une remarque est à faire concernant la division de nos corpus en ensembles d'entraînement et de test. Les publications faisant état de travaux en catégorisation automatique de textes tentent lorsque c'est possible d'utiliser des divisions entraînement/test standards pour faciliter la comparaison de leurs résultats. En effet, divers facteurs peuvent avoir une influence à ce sujet, qu'on pense évidemment au nombre de documents appartenant à chaque ensemble, mais aussi à des phénomènes temporels [NMTM00]. En s'y arrêtant, on peut comprendre qu'un classificateur, entraîné sur un ensemble de textes datant tous d'une période

antérieure à celle d'où proviennent les textes de l'ensemble de test, puisse performer différemment par rapport à un classificateur entraîné et testé sur des textes plus homogènes quant à leur date. L'une des raisons est que pendant une certaine période de temps, un mot peut prendre une signification différente de celle qu'on lui associe à un autre moment. Pour mieux saisir, voici un exemple simple dans le contexte de nouvelles d'actualité : pendant un certain temps, le mot «*Dolly*» a fait les manchettes en faisant référence à une brebis clonée, alors que pendant une autre période, ce mot faisait allusion à un ouragan dans les bulletins de nouvelles. Un classificateur entraîné sur des textes où Dolly représente une brebis va commettre des erreurs s'il est testé sur des textes où ce même mot est interprété comme étant un ouragan. Cette parenthèse avait pour but de souligner l'importance d'uniformiser entre les expérimentations les ensembles d'entraînement et de test utilisés. Par contre, dans le cadre de nos travaux, il n'a pas été possible de respecter ce principe la plupart du temps et d'utiliser les mêmes ensembles d'entraînement que d'autres travaux du domaine. Ces derniers s'intéressent en grande partie à améliorer la performance des classificateurs dans des conditions idéales, c'est-à-dire en présence de plusieurs documents d'entraînement. La méthode que nous proposons se veut plutôt être bénéfique dans le cas où le nombre de documents d'entraînement est limité. Conséquemment, il a été nécessaire de réduire la taille des ensembles d'entraînement proposés pour chaque corpus.

Le premier ensemble de textes utilisé provient de l'agence de presse très connue Reuters. Plusieurs versions de corpus proviennent de cette organisation et ont été largement utilisées en recherche. Pour nos travaux, nous avons choisi d'exploiter la dernière version, rendue disponible[25] assez récemment : Reuters Corpus Volume 1

[25] Pour plus d'informations : http://about.reuters.com/researchandstandards/corpus/.

(RCV1). Celle-ci rassemble un peu plus de 800 000 courtes nouvelles classées manuellement parmi une centaine de catégories, organisées hiérarchiquement. Un grand nombre de catégories font référence à l'économie (actions, intérêts, prêts, liquidités, etc.), mais des catégories plus générales sont aussi incluses (science et technologie, art et culture, etc.) Cette collection rassemble toutes les nouvelles en anglais produites par les journalistes de Reuters entre le 20 août 1996 et le 19 août 1997 [RSW02]. Dans ce contexte, il faut savoir que plus d'une catégorie peut être associée à un même document : en moyenne, chacun se voit assigner 3.5 étiquettes de catégorie. Le corpus RCV1 se prête très bien à la recherche en catégorisation automatique de textes, mais quelques ajustements et corrections sont proposés dans [LYRL04] afin de le rendre encore plus pertinent dans ce contexte.

Pour nos expérimentations, des ensembles d'entraînement de taille variable allant de 100 à 1000 documents ont été constitués en choisissant les premiers documents. L'ensemble de test, quant à lui, demeure toujours le même. Il s'agit des 10 000 documents consécutifs se trouvant au centre du corpus, à partir du 400 000e document. Comme on se souvient, la technique proposée requiert un troisième ensemble de textes, non libellés cette fois, dans lequel sera faite l'analyse de cooccurrence. Nous avons opté pour un ensemble comptant un peu plus de 100 000 documents formant les deux derniers mois du corpus, juillet et août 1997.

Une deuxième collection a également été mise à l'épreuve, WebKB[26]. De nature différente, ce corpus comporte des documents provenant de diverses universités américaines, soit les pages Web reliées aux départements d'informatique de celles-ci. Contrairement au corpus précédent, chaque document est dans ce cas-ci associé à une seule catégorie. Originalement, sept classes étaient proposées, mais pour se

[26] Disponible à l'adresse : http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/.

rapprocher des expérimentations présentées par [NMTM00] et [ZH01], seulement les quatre plus populeuses en dehors de la catégorie «*Autres*» ont été utilisées : «*Cours*», «*Étudiant*», «*Faculté*» et «*Projet*».

Pour la division de ce corpus, nous avons choisi d'utiliser la même méthodologie que [NMTM00]. Quatre ensembles de test sont formés, chacun contenant les pages de l'un des quatre départements d'informatique dont l'ensemble des pages apparaissent dans le corpus (Cornell, Texas, Washington, Wisconsin). Pour chacun de ces ensembles de test, 10 ensembles d'entraînement disjoints sont formés en choisissant au hasard des textes parmi ceux qui restent, un nombre égal pour chaque catégorie. Dans notre cas, nous avons répété l'expérience avec trois tailles différentes pour l'ensemble d'entraînement (20, 40 et 80 documents). Les résultats présentés plus loin correspondent à la moyenne des 40 tests (4 ensembles de test x 10 ensembles d'entraînement) effectués pour une taille donnée de l'ensemble d'entraînement. Pour étudier la cooccurrence, nous avons pris la décision d'utiliser l'ensemble des textes du corpus, soit 8007 documents, étant donné sa taille déjà réduite.

Finalement, une troisième collection de textes a été exploitée lors des expérimentations : il s'agit du corpus Ohsumed. Encore une fois, il présente des caractéristiques différentes des corpus précédents. À l'origine, il est constitué d'environ 300 000 entrées d'une banque de données médicales, contenant chacune un titre et un résumé d'article. En fait, ce sont des références concernant la littérature médicale gérée par la National Library of Medicine. Les étiquettes de catégories correspondent à des termes MeSH[27] valides. Comme c'est généralement le cas lorsqu'il est utilisé en catégorisation automatique de textes [LSCP96], nous

[27] Acronyme pour «*Medical Subject Headings*».

n'avons utilisé ici que les sous-catégories correspondant aux maladies du cœur («*Heart Diseases*») et contenant au moins 75 documents. Il en résulte un ensemble de 49 catégories, nombre qui se situe entre les 4 catégories de WebKB et les 103 catégories de RCV1. Quant au vocabulaire, il présente plusieurs termes médicaux et techniques spécialisés [HBLH94].

La division du corpus a été faite de façon similaire à celle retrouvée dans [LSCP96]. L'ensemble d'entraînement comprend les textes des années 1988 à 1990 Parmi ceux-ci, les 2000 premiers documents ont réellement servi à l'entraînement de nos classificateurs, alors que les 10 445 documents restants ont formé l'ensemble utilisé pour l'analyse de cooccurrence. Dans le cas des documents d'entraînement, une procédure similaire à celle de WebKB a été suivie : pour une taille donnée de l'ensemble d'entraînement, plusieurs groupes disjoints ont été formés parmi les 2000 premiers documents et les résultats présentés plus loin correspondent à la moyenne des tests effectués sur ces groupes. Quant à l'ensemble test, il est constitué des documents de l'année 1991, au nombre de 3632.

Le tableau 6.1 résume les principales caractéristiques des trois corpus exploités.

	Reuters[28]	WebKB	Ohsumed
Taille de l'ensemble d'entraînement[29]	> 800 000	1006	12 445
Taille de l'ensemble de test		3094	3632
Nombre de catégories	103	4	49
Plusieurs catégories possibles par texte ?	Oui	Non	Oui
Type de textes	Courtes nouvelles	Pages Web	Articles médicaux

Tableau 6.1 - Résumé des renseignements sur les corpus de textes utilisés

[28] Il n'y pas encore de séparation *entraînement / test* standard pour cette collection.
[29] Il s'agit de la taille standard utilisée par les chercheurs pour ces corpus de textes. Dans nos expérimentations, nous avons plutôt utilisé des sous-ensembles de ceux-ci.

6.1.2 Caractéristiques des classificateurs utilisés

Un deuxième point important à clarifier à propos de notre méthodologie d'évaluation est le choix des types de classificateurs qui entrent en jeu. Deux d'entre eux ont été mis à l'œuvre dans le cadre de nos travaux, à savoir un classificateur k-voisins les plus proches (kNN) et un classificateur basé sur les machines à support vectoriel (SVM). Ceux-ci ont été choisis parce qu'il s'agit des deux algorithmes d'apprentissage automatique qui performent le mieux en catégorisation automatique de textes, selon de récentes études comparatives [YL99] [Seb99] [Joa98a]. Étant donné qu'il en existe de nombreuses variantes et que plusieurs paramètres peuvent fluctuer, il importe de présenter de façon assez détaillée les caractéristiques des algorithmes que nous avons utilisés.

D'abord, mentionnons que dans tous les cas, les documents sont représentés selon l'approche vectorielle, chaque attribut représentant un mot du vocabulaire. Aucun processus de recherche de radical («*stemming*») n'est appliqué, de même qu'aucune liste de mots vides de sens («*stop words*») n'est utilisée. Quant à la technique de pondération des attributs, c'est l'approche TFIDF qui a été retenue. Pour réduire la taille des vecteurs tout en éliminant les mots inutiles, une sélection d'attributs est appliquée. Une première phase de sélection consiste tout simplement à mettre de côté les mots dont la fréquence d'apparition parmi les documents ne dépasse pas un certain seuil minimal, fixé empiriquement à 4 d'après des expérimentations antérieures. Autrement dit, si un terme n'apparaît pas dans au moins 4 documents différents, il est éliminé du vocabulaire. Par la suite, une deuxième phase de sélection se base sur une mesure d'évaluation particulière, le gain d'information. Encore une fois, un seuil empirique a été fixé, et ce à 0.01. Donc, les mots dont la valeur du gain d'information par rapport aux catégories se tient sous cette valeur de seuil sont éliminés du vocabulaire.

Comme il a été dit précédemment, un des classificateurs exploité lors de nos expérimentations est l'algorithme des k-voisins les plus proches. Les grandes lignes de cet algorithme ont déjà été présentées au chapitre 2, mais certains détails d'implantation sont encore à préciser. En ce qui a trait à la mesure de similarité mise en place, le choix s'est arrêté sur la similarité cosinusoïdale, pour son aptitude à mieux comparer les documents textuels. Un second paramètre à fixer est le nombre de voisins à considérer : la valeur retenue est de 30. Le résultat du calcul de la similarité d'un document à classer avec ses voisins sert à obtenir, pour chaque catégorie, une valeur représentant le degré de confiance de l'appartenance du document à cette catégorie. Cette valeur correspond à la somme des similarités avec les voisins appartenant à cette catégorie. Au chapitre 2, il a été spécifié que l'algorithme des k-voisins les plus proches ne comprenait pas de phase d'entraînement en tant que telle. Dans la version que nous avons implanté, au contraire, le classificateur s'entraîne sur les documents d'entraînement pour apprendre un seuil pour chaque catégorie, un seuil à partir duquel classer un nouveau document lorsque la valeur calculée d'après les similarités le dépasse. Ces seuils sont choisis de manière à réduire au minimum le nombre d'erreurs de classification obtenues sur l'ensemble d'entraînement. Cette configuration de l'algorithme kNN s'apparente à celle de [YL99].

Le deuxième algorithme d'apprentissage ayant servi lors de nos expérimentations est celui des machines à support vectoriel. La théorie soutenant les SVM a été introduite au chapitre 2. L'implantation choisie est SVM^{light} qui est accessible gratuitement sur le Web[30] et qui a été exploitée à plusieurs reprises dans le domaine de la catégorisation de textes. Des précisions sur les algorithmes d'optimisation mis en place sont présentées dans [Joa98b].

[30] Disponible à http://svmlight.joachims.org/.

6.2 Résultats

Les trois mesures de cooccurrence présentées au chapitre 4 ont été testées dans le cadre de nos travaux, à savoir le test du χ^2, le ratio de vraisemblance et l'information mutuelle. Il semblait intéressant d'observer l'effet de différentes mesures d'association sur les performances de notre approche et de déterminer laquelle serait la plus appropriée. Le tableau 6.2 présente un aperçu des paires de mots jugés cooccurrents par les mesures du χ^2 et de l'information mutuelle lors d'une expérimentation sur le corpus RCV1. On y retrouve, pour cette expérimentation seulement, les 25 paires de mots ayant obtenu le plus haut score au test de cooccurrence, ces paires étant formées d'un mot inconnu du classificateur et d'un mot faisant partie de son vocabulaire. La simple observation de ce tableau permet déjà de constater que les paires de mots cooccurrents retournées par le test du χ^2 semblent très pertinentes, c'est-à-dire qu'elles représentent réellement des mots qu'on retrouve dans des contextes similaires. Quant aux paires de mots retournées par la mesure de l'information mutuelle, on remarque que plusieurs mots rares en font partie. Cela correspond bien à la remarque faite au chapitre 4 concernant cette mesure d'association et sa tendance à favoriser les termes rares. Les paires de mots retournées par la mesure du ratio de vraisemblance ne sont pas présentées ici, étant donné la ressemblance très forte avec les résultats du test du χ^2.

Test du χ^2			Mesure de l'information mutuelle		
Mot inconnu	Mot connu	Score	Mot inconnu	Mot connu	Score
brazilian	brazil	38439	golub	mn	9.6
salomon	brothers	35497.4	ehlers	mn	9.3
sao	brazil	34469	az	mn	9.1
paulo	brazil	34103.5	whitney	mn	9.1
crude	barrel	31092.6	beloit	mn	8.6
lehman	brothers	29324.7	harradine	canberra	8.3
brent	barrel	25446.9	lanterns	canberra	8.3
digest	newspapers	20861.6	dianne	mn	8.3
barrels	barrel	20754.8	pramod	students	8.1
stories	newspapers	20054.6	nd	mn	8.0
reais	brazil	19698.5	tic	mn	8.0
accuracy	newspapers	17814.4	ult	mn	7.9
crudes	barrel	16201.6	whining	considerable	7.8
verified	newspapers	15925.2	cer	canberra	7.8
vouch	newspapers	15847.5	pillows	canberra	7.8
press	newspapers	14466.1	jull	canberra	7.8
janeiro	brazil	14200.5	colston	canberra	7.8
export	exports	14072.7	nh	mn	7.7
gasoline	barrel	13684.5	minn	mn	7.7
dec	oct	13221.2	bbi	mn	7.6
nymex	barrel	12864.4	trhy	students	7.6
sep	oct	12593.8	dmitriev	card	7.5
rio	brazil	12430.6	minneapolis	mn	7.5
ipe	barrel	12243.7	presale	mn	7.5
iraqi	barrel	12161.1	qual	mn	7.5

Tableau 6.2 - Exemples de paires de mots jugés cooccurrents

Des expérimentations préliminaires ont été menées avec les trois mesures de cooccurrence, exploitées dans un contexte de catégorisation de textes selon la méthode proposée précédemment. Il a été constaté que le χ^2 donnait de meilleurs résultats que l'information mutuelle (celle-ci entraînant même parfois une détérioration de la performance du classificateur). Le ratio de vraisemblance, quant à lui, offrait la possibilité au classificateur de performer de façon semblable au test du χ^2. Comme à première vue le χ^2 constitue un test d'hypothèse très connu et très

simple à calculer, c'est à partir de cette mesure de cooccurrence que nous avons décidé de mener une évaluation plus exhaustive de notre approche.

Les tableaux 6.3, 6.4 et 6.5 présentent respectivement les résultats obtenus sur les corpus RCV1, WebKB et Ohsumed. Il y est possible de comparer le score de micro-F1 atteint par un classificateur sans et avec analyse de cooccurrence (selon la démarche discutée au chapitre précédent) sur des ensembles d'entraînement de différentes tailles. Comme il a été mentionné auparavant, un classificateur kNN et un classificateur SVM ont été mis à l'épreuve. Pour chacun des tests, différentes valeurs pour le seuil de cooccurrence ont été essayées et les scores présentés dans ces tableaux représentent la meilleure performance atteinte parmi ces essais.

Taille de l'ensemble d'entraînement	Micro-F1 sans analyse de cooccurrence	Micro-F1 avec analyse de cooccurrence	Différence (valeur absolue)	Différence (pourcentage)
Classificateur kNN				
100	0.191	0.2142	+ 0.0232	+ 12.15
200	0.2152	0.2506	+ 0.0354	+ 16.45
300	0.2829	0.3065	+ 0.0236	+ 8.34
400	0.4015	0.4226	+ 0.0211	+ 5.26
500	0.4487	0.473	+ 0.0243	+ 5.42
1000	0.6143	0.6233	+ 0.0090	+ 1.47
Classificateur SVM				
100	0.2029	0.2077	+ 0.0048	+ 2.37
200	0.178	0.1709	- 0.0071	- 3.99
300	0.2633	0.266	+ 0.0027	+ 1.03
400	0.3828	0.3934	+ 0.0106	+ 2.77
500	0.415	0.4288	+ 0.0138	+ 3.33
1000	0.5674	0.5758	+ 0.0084	+ 1.48

Tableau 6.3 - Évaluation de la méthode proposée sur le corpus RCV1

Taille de l'ensemble d'entraînement	Micro-F1 sans analyse de cooccurrence	Micro-F1 avec analyse de cooccurrence	Différence (valeur absolue)	Différence (pourcentage)
Classificateur kNN				
20	0.5445	0.5591	+ 0.0146	+ 2.68
40	0.6033	0.6205	+ 0.0172	+ 2.85
80	0.6147	0.628	+ 0.0133	+ 2.16
Classificateur SVM				
20	0.4303	0.4736	+ 0.0433	+ 10.06
40	0.4775	0.5133	+ 0.0358	+ 7.50
80	0.5403	0.5743	+ 0.0340	+ 6.29

Tableau 6.4 - Évaluation de la méthode proposée sur le corpus WebKB

Taille de l'ensemble d'entraînement	Micro-F1 sans analyse de cooccurrence	Micro-F1 avec analyse de cooccurrence	Différence (valeur absolue)	Différence (pourcentage)
Classificateur kNN				
100	0.2409	0.2467	+ 0.0058	+ 2.41
500	0.4435	0.4529	+ 0.0095	+ 2.13
1000	0.5573	0.5592	+ 0.0019	+ 0.34
2000	0.586	0.5863	+ 0.0003	+ 0.05
Classificateur SVM				
100	0.1814	0.189	+ 0.0076	+ 4.19
500	0.4228	0.4367	+ 0.0139	+ 3.29
1000	0.5850	0.5888	+ 0.0038	+ 0.65
2000	0.635	0.6372	+ 0.0022	+ 0.35

Tableau 6.5 - Évaluation de la méthode proposée sur le corpus Ohsumed

6.3 Discussion

Les résultats de cette évaluation ouvrent la porte à des réflexions intéressantes. À la lumière de cette compilation, une des premières remarques à faire est que dans tous les cas présentés (sauf un), la mise en place de l'approche d'analyse de cooccurrence proposée a donné lieu à une amélioration des résultats du classificateur. Autrement dit, étudier la cooccurrence des mots absents de son vocabulaire par rapport à ceux en faisant partie dans un ensemble de textes non

étiquetés pour ensuite modifier la représentation des documents à classer a permis d'atteindre un meilleur score de micro-F1.

Bien que l'amélioration des performances des classificateurs découlant de notre approche soit modeste, un argument important en sa faveur est que son intégration dans une application de catégorisation de textes s'avère très peu coûteuse en temps humain. En fait, la seule tâche à accomplir est la constitution d'un corpus de textes non libellés. Vraisemblablement, il s'agit d'une activité très rapide puisqu'il n'est aucunement question d'associer une catégorie à ces textes, mais simplement de les rassembler. Évidemment, un certain coût de calcul est ajouté au processus de la machine, qui doit évaluer la cooccurrence de chaque paire de mots considérée. Ce coût peut être très variable selon le mécanisme choisi pour mesurer la cooccurrence. Dans le cadre de nos travaux, c'est un moteur de recherche qui a été utilisé. Trois requêtes sont requises pour acquérir les quatre valeurs impliquées dans le calcul du score de cooccurrence entre deux mots. Le coût de cette opération est donc fonction de l'efficacité du moteur de recherche.

À propos de la compatibilité de la nouvelle approche envers différents types de classificateurs, on peut à tout le moins remarquer qu'un classificateur kNN et un classificateur SVM ont tous deux su profiter de la mise en place du mécanisme d'analyse de cooccurrence. Pour chacun de ces classificateurs, une hausse de micro-F1 a été observée suite à la mise en œuvre de l'approche d'analyse de cooccurrence proposée. Donc, celle-ci semble être utile de façon générale et n'est pas seulement compatible avec un seul algorithme d'apprentissage. Selon les corpus, les gains ont été parfois supérieurs avec le classificateur kNN et d'autres fois supérieurs avec le classificateur SVM. Il est par contre difficile d'expliquer ces variations, d'autant plus que sans aucune analyse de cooccurrence, les deux classificateurs performent aussi plus ou moins bien selon les corpus de textes.

Il est à noter que les résultats présentés précédemment sont basés sur des ensembles d'entraînement de faible taille, pour la simple et bonne raison que l'intérêt de la méthode mise au point est justement de pallier un manque de documents d'entraînement. Dans des situations où ces documents sont plus nombreux, l'effet de l'analyse de cooccurrence se fait de moins en moins sentir et devient rapidement nul. En effet, plus il y a de documents d'entraînement, moins le classificateur rencontre des termes inconnus dans les textes qu'il classe. Or, c'est précisément sur ces termes qu'agit l'analyse de cooccurrence. Cette observation concorde avec les résultats obtenus par [NMTM00] et [ZH01]. Leurs travaux ont mis en évidence le fait que les bénéfices résultant de l'exploitation de documents non libellés sont moins évidents lorsque l'ensemble d'entraînement est plus important.

Par ailleurs, une constatation qui ressort de l'analyse des résultats est qu'avec un nombre vraiment trop faible de documents d'entraînement, la méthode ne s'avère pas aussi efficace qu'avec un nombre un peu plus élevé de ceux-ci. Il semble qu'il faille un minimum de textes d'entraînement pour permettre à l'analyse de cooccurrence de cerner les mots représentatifs des catégories avec justesse. Par contre, si la taille de l'ensemble d'entraînement devient trop grande, le vocabulaire du classificateur va être assez riche pour que la majorité des mots soient reconnus et, par conséquent, l'analyse de cooccurrence ne sera plus en mesure d'apporter une aide significative. Le comportement général du gain apporté par l'analyse de cooccurrence en fonction du nombre de documents d'entraînement est donc le suivant : il commence petit, croît pendant un certain temps puis à partir d'un certain niveau commence à descendre. On peut toutefois sentir qu'il est difficile de déterminer avec exactitude les conditions dans lesquelles l'analyse de cooccurrence va permettre les meilleurs gains.

Une interrogation survient lors de l'observation des résultats précédents. Les bénéfices de l'approche d'analyse de cooccurrence apparaissent dans des contextes où le score de micro-F1 atteint par les classificateurs est à la base faible. D'autre part, il est facile de concevoir que l'adoption d'un mécanisme de catégorisation automatique de textes implique l'attente d'un niveau de micro-F1 minimal, acceptable. C'est-à-dire que quelqu'un qui choisit d'utiliser cette technologie au sein d'une application est en droit d'espérer un certain niveau de micro-F1. Dans le cas où un classificateur présente un degré de performance jugé insuffisant au départ, est-ce que la mise en place de l'analyse de cooccurrence est en mesure de l'élever à un degré plus acceptable ? La réponse réaliste à cette question est probablement non, vu les gains observés plus haut. En revanche, les points de micro-F1 gagnés le sont sans besoin de ressources humaines ou presque, et sont donc toujours les bienvenus. On peut même supposer que cette mesure combinée à d'autres outils pourrait alors amener un classificateur à un niveau acceptable de performance.

Une limite de nos travaux réside dans le choix du seuil de cooccurrence minimal à partir duquel deux mots sont jugés cooccurrents. Chacune de nos expériences a mis en jeu plusieurs valeurs de seuil et les résultats présentés dans nos tableaux représentent le seuil ayant permis l'obtention de la meilleure valeur de micro-F1. En chiffres, on remarque que ce seuil optimal (pour le test du χ^2) oscille entre des valeurs de 100 et de 1000. En réalité, il faudrait définir un seuil de cooccurrence fixe ou une façon précise de calculer un tel seuil. Malheureusement, l'exploitation d'un ensemble de validation sur lequel on pourrait optimiser le seuil est pratiquement impossible, étant donné la rareté des documents libellés dans les domaines d'application envisagés de la méthode. Il faut tout de même préciser que le choix d'un seuil plutôt qu'un autre dans l'intervalle [100, 1000] n'entraîne que de

légères variations. On peut constater que de s'arrêter sur un seuil de 500 aurait constitué un bon compromis dans toutes les expériences menées ici.

Une analyse plus en profondeur des résultats a permis de mettre en relief un autre phénomène. On se souvient que le score de micro-F1 est une combinaison de la précision et du rappel[31]. Il apparaît que de façon générale, lorsque le nombre de documents est restreint et que le score de micro-F1 est faible, le principal responsable est le rappel. Un coup d'œil sur les résultats après l'application de l'analyse de cooccurrence permet de constater que c'est justement cet aspect qui est favorisé par notre approche, au détriment parfois de la précision, mais de façon très légère. Le tableau 6.6 illustre le faible taux de rappel d'un classificateur lorsque le nombre de documents d'entraînement est restreint ainsi que l'amélioration du rappel par l'application de l'analyse de cooccurrence. Les données font référence au corpus RCV1, mais un comportement similaire s'est produit avec les autres collections de textes.

Taille de l'ensemble d'entraînement	Précision sans analyse de cooccurrence	Rappel sans analyse de cooccurrence	Variation de la précision avec analyse de cooccurrence	Variation du rappel avec analyse de cooccurrence
100	0.8287	0.1079	+ 0.0086	+ 0.0149
200	0.838	0.1234	- 0.0234	+ 0.0246
300	0.9021	0.1678	- 0.0215	+ 0.0177
400	0.9039	0.258	- 0.0098	+ 0.0187
500	0.8917	0.2998	- 0.0044	+ 0.0226
1000	0.8444	0.4828	- 0.0107	+ 0.0149

Tableau 6.6 - Comportement de la précision et du rappel sur le corpus RCV1 et sur un classificateur kNN

Les travaux de [NMTM00] et [ZH01], présentés au chapitre 3 et utilisant respectivement l'algorithme EM et l'indexation sémantique latente (LSI) pour tirer

[31] Voir la section 2.2.

profit de documents non libellés, ont testé leur approche respective sur le corpus WebKB, entre autres. Il est donc possible de comparer nos résultats avec les leurs. Nous avons appliqué un classificateur bayésien naïf (puisque l'algorithme EM est appliqué sur ce type de classificateur) sans et avec notre approche d'analyse de cooccurrence sur le corpus WebKB. Le tableau 6.7 rapporte les résultats publiés à propos de l'algorithme EM et de la LSI, puis les résultats obtenus lors de nos expérimentations avec l'analyse de cooccurrence. Les ensembles d'entraînement et de test ont été formés d'une façon semblable à celle présentée dans [NMTM00], mais ne sont pas parfaitement identiques, à cause d'un facteur aléatoire dans la méthodologie. C'est ce qui explique les différences observées entre les résultats obtenus sur le classificateur bayésien naïf de base.

Taille de l'ensemble d'entraînement	Bayes	Bayes + EM	LSI	Bayes	Bayes + analyse de cooccurrence
20	0.6031	0.6722	0.6477	0.6113	0.6176
40	0.6912	0.7458	0.6960	0.7116	0.7206
80	0.7599	0.7639	0.7255	0.7337	0.7374

Tableau 6.7 - Comparaison de notre approche avec deux autres travaux tirant profit de documents non étiquetés, sur le corpus WebKB

On peut voir que les gains obtenus par l'algorithme EM sont supérieurs à ceux de notre approche pour des ensembles d'entraînement très petits, mais à partir de 80 documents d'entraînement, le bénéfice est comparable. Pour des ensembles encore plus grands, l'algorithme EM entraîne même une baisse de performance. L'analyse de cooccurrence, bien que moins efficace pour des ensembles d'entraînement très petits, n'occasionne pas de détérioration lors d'une hausse du nombre de documents d'entraînement. Les gains diminuent graduellement, jusqu'à devenir négligeables. Maintenant, par rapport à l'approche LSI, déjà à partir de 40 documents d'entraînement, l'analyse de cooccurrence permet d'atteindre un score

supérieur. En somme, on ne peut affirmer que notre approche surpasse les deux autres, puisqu'elles semblent toutes performer différemment selon les conditions. Par contre, on peut remarquer qu'elle s'y compare très bien, à tout le moins sur le corpus WebKB. Il serait intéressant de mener une comparaison plus exhaustive.

Un dernier détail à mentionner est que dans notre contexte de recherche, la catégorisation a toujours été effectuée par lot («*batch*»), c'est-à-dire qu'un ensemble complet de textes à classer était soumis au classificateur d'un seul coup. Dans une situation réelle, il se peut fort bien que les documents à classer soient soumis à l'application un à la fois («*on-line*»). Le seul aspect de notre technique qui ne peut se transposer dans ce contexte est le seuil minimal d'apparition que les termes absents du vocabulaire doivent franchir pour être considérés par l'analyse de cooccurrence. Dans un contexte de catégorisation «*on-line*», il n'y aurait évidemment aucun moyen de vérifier cette condition et, par conséquent, tous les mots inconnus du classificateur seraient l'objet d'une étude de cooccurrence.

Résumé

Suite à l'examen des résultats obtenus au cours de nos expérimentations, on peut juger de la pertinence de l'approche proposée. Dans des situations où la taille de l'ensemble d'entraînement est petite, son application entraîne une hausse de micro-F1. En général, cette hausse, qui varie selon les tâches de classification et les types de classificateurs, n'est pas immense, mais s'avère tout de même significative de par son faible coût. Nos travaux constituent donc un pas dans la bonne direction.

Conclusion

Retour sur le mémoire

Au terme de ce mémoire, il convient de revenir sommairement sur ce qu'il a permis de mettre en évidence. Avant toute chose, le problème de la catégorisation automatique de textes a été introduit dans les deux premiers chapitres et on a pu constater son potentiel dans un contexte où une gestion efficace de l'information devient essentielle. Surtout, le fonctionnement, les particularités et les difficultés des classificateurs automatiques de textes ont été abordés. Des considérations relatives à la représentation des documents, à la sélection d'attributs et aux algorithmes d'apprentissage ont été exprimées.

Entre autres, un des aspects problématiques à avoir été mis en évidence est la constitution d'un corpus d'entraînement. La classification automatique de textes, mettant en place un mécanisme d'apprentissage supervisé, se base sur l'observation d'un ensemble de textes déjà classés par des humains. Le problème est que l'étiquetage de ces textes consomme du temps et des ressources considérables. Des solutions pour atténuer cette difficulté sont donc souhaitables. La discussion s'est tournée au chapitre 3 vers des moyens mis en place par certains chercheurs pour tirer profit de connaissances extérieures à cet ensemble d'entraînement. L'objectif poursuivi est d'éventuellement réduire le besoin d'un grand nombre de documents étiquetés requis pour obtenir le degré de performance souhaité d'un classificateur, ou, dans les situations où la cueillette de ceux-ci ne pose pas problème, d'améliorer tout simplement la qualité de la catégorisation. Dans cette optique, des travaux ont porté sur l'utilisation d'un lexique ou sur l'utilisation de textes non étiquetés. Cette dernière proposition ayant

démontré du potentiel, la question se posait à savoir quelle information éventuellement utile au processus de catégorisation on pouvait tirer de ces textes.

On a alors vu que la connaissance du degré de cooccurrence entre les mots, leur tendance à apparaître dans un même contexte, pouvait être extraite d'un ensemble de textes non libellés et pouvait être utile à un classificateur. En ce sens, la notion de cooccurrence a été précisée au chapitre 4 et quelques façons de la mesurer numériquement ont été abordées.

Résumé de la méthode proposée et de son utilité

Suite à ce survol de l'état de l'art dans le domaine de la catégorisation automatique de textes et à cette introduction à l'utilisation de documents non libellés et à la cooccurrence de mots, le sujet principal du mémoire a été mis à l'avant-plan au chapitre 5. Il s'agit de la proposition d'une approche étudiant la cooccurrence de mots au sein de documents non classés et modifiant en conséquence la représentation des textes à traiter. Sans revenir sur tous les détails, rappelons qu'un classificateur se représente un texte à classer à partir des mots qu'il a déjà rencontrés au cours de son entraînement. Il fait abstraction de tous les mots inconnus qui surviennent. La solution proposée est donc d'étudier, à l'intérieur de textes non libellés, la cooccurrence de ces mots inconnus par rapport aux mots présents dans le vocabulaire du classificateur. Ensuite, lorsque le classificateur rencontre un mot nouveau dans un texte à classer, et que ce mot s'est avéré très cooccurrent avec un mot qu'il connaît, alors il simule la présence de ce mot connu. Comme il a été jugé que ces deux mots avaient tendance à apparaître dans les mêmes documents, cette action peut aider le classificateur à mieux cerner la catégorie d'appartenance du texte.

Une évaluation exécutée sur différents corpus de textes et sur différents types de classificateurs a permis de démontrer que la mise en place d'un tel mécanisme pouvait donner lieu à une amélioration des performances du classificateur. La présentation des résultats de cette évaluation ainsi qu'une discussion à leur sujet ont occupé le chapitre 6. Certes, les hausses de micro-F1 observées au cours de nos expérimentations ne sont pas vertigineuses, au contraire. Dans la majorité des cas, l'analyse de cooccurrence a permis de gagner entre 1 et 4 points de micro-F1. Il faut savoir que, dans le domaine de la catégorisation automatique de textes, de telles améliorations sont en général très bien reçues, dans un contexte où l'on travaille à parfaire un classificateur dans des conditions «*idéales*», c'est-à-dire avec un ensemble d'entraînement de taille raisonnable. Comme il a déjà été spécifié, nos travaux se situent plutôt dans un contexte où cet ensemble est restreint. Évidemment, les résultats obtenus sans analyse de cooccurrence sont au départ beaucoup moins bons. L'impact paraît donc moins attrayant. Toutefois, comme l'analyse de cooccurrence est exécutée sur des textes non étiquetés, aucune ressource humaine ou presque n'est requise pour mener à bien cette tâche et donc les gains obtenus n'auront coûté que du temps de calcul machine.

Une autre conclusion à tirer : les expérimentations menées ont permis de déterminer que le test du χ^2 est préférable dans notre contexte pour mesurer la cooccurrence de mots que les autres mesures étudiées. En somme, l'approche nouvelle exposée dans ce mémoire a le potentiel d'assister un classificateur dans l'exécution de sa tâche lorsque les données d'entraînement se font plus rares. À défaut d'avoir observé des gains très importants au cours de nos expérimentations, nous avons pu mettre en évidence certaines problématiques et nous avons lancé une piste intéressante de solution qui mérite sûrement d'être raffinée.

Limites de l'approche et avenues de recherche pertinentes

Nos travaux comportent évidemment certaines limites ouvrant la voie à d'autres avenues de recherche. Les multiples variables intervenant dans le processus de catégorisation créent un nombre quasi infini de configurations de paramètres à tester. Malheureusement, le temps, lui, n'est pas infini et il a été nécessaire de fixer certains paramètres pour en étudier d'autres plus en profondeur. Évidemment, il aurait été intéressant d'observer le comportement de notre approche sur plus de trois corpus de textes et sur plus de deux classificateurs. En plus, faire varier les divers paramètres des classificateurs et la méthode de sélection d'attributs aurait peut-être permis de faire ressortir des phénomènes de compatibilité entre le processus d'analyse de cooccurrence et les caractéristiques des classificateurs.

Dans un même ordre d'idées, nos expérimentations n'ont pas pris en considération la nature du corpus de documents non libellés dans lequel se fait l'analyse de cooccurrence. Est-il préférable qu'il contienne des textes de nature semblable à ceux traités par le classificateur ou peut-il être constitué de documents cueillis sur le Web aléatoirement ? Il serait très possible que les bénéfices de la méthode soient influencés à un certain niveau par ce facteur. De la même façon, nos travaux n'ont pas permis d'évaluer l'influence de la taille de ce corpus.

Une étude plus poussée s'impose aussi concernant la détermination du seuil de cooccurrence idéal au-delà duquel deux mots sont jugés assez cooccurrents pour être substitués l'un à l'autre. L'idéal serait de mettre au point un moyen de calculer ce seuil idéal à partir de certaines caractéristiques de la tâche de catégorisation, mais les observations actuelles n'ont pas permis d'y arriver.

Un autre aspect qui semble ne pas avoir été étudié en profondeur concernant la méthode mise de l'avant est le choix des mots du vocabulaire étant impliqués dans

l'analyse de cooccurrence. La stratégie retenue, qui consiste à ne considérer que le mot le plus relié à chacune des catégories selon le test du χ^2, peut paraître plutôt arbitraire. Il serait peut-être plus utile de baser ce choix de mots sur un autre critère.

Nos travaux portent aussi à croire qu'il pourrait être profitable d'imaginer d'autres façons d'incorporer dans le processus de catégorisation la connaissance sur la cooccurrence des mots. Également, des moyens supplémentaires de traiter le problème des mots inconnus du classificateur seraient sans contredit les bienvenus.

Autres avenues de recherche connexes

En ce qui a trait à la catégorisation automatique de textes en général, plusieurs portes sont également encore ouvertes pour la recherche. Du point de vue de la représentation des documents, une multitude de travaux ont déjà porté sur la recherche d'alternatives à la traditionnelle représentation «*bag-of-words*». Cette dernière demeure pourtant celle qui semble la meilleure. Quoi qu'il en soit, il est toujours légitime de tenter de développer de nouvelles façons de représenter les documents. Mais, il serait sûrement intéressant d'essayer de comprendre précisément ce qui rend la représentation «*bag-of-words*» si efficace.

En ce qui concerne la sélection d'attributs, c'est un des aspects qui s'avèrent influencer significativement la capacité des classificateurs à bien effectuer leur travail. Alors, le développement de nouvelles techniques de sélection d'attributs constitue un thème qui inspirera encore plusieurs chercheurs. Un champ de recherche supplémentaire est bien sûr l'application et l'adaptation de nouveaux

algorithmes d'apprentissage au problème de la catégorisation automatique de textes.

On peut également noter deux problèmes qui interfèrent certainement dans le processus de catégorisation : la synonymie et la polysémie. Le fait que des mots distincts puissent faire référence à une même sémantique ou qu'un mot unique puisse véhiculer des sens différents rend difficile la tâche d'un classificateur. D'un texte à l'autre, le vocabulaire utilisé pour parler d'un même sujet peut varier et il devient plus difficile pour un algorithme de généraliser, de faire des recoupements entre ces textes. La mise en correspondance risque aussi de se faire difficilement entre les textes d'entraînement et les textes à classer. Des solutions efficaces à ces problèmes sont donc souhaitables.

En un mot, la catégorisation automatique de textes est un domaine loin d'en être à ses débuts, mais qui présente encore plusieurs défis. Il s'agit d'une technologie ayant le potentiel de soutenir des applications très utiles et intéressantes, mais démontrant certaines lacunes qui doivent être résolues. Les travaux de recherche communiqués par ce mémoire s'attaquaient à l'une de ces lacunes : le coût de constitution des ensembles d'entraînement. Nous avons démontré qu'en étudiant la cooccurrence de mots dans une collection de textes non libellés, un classificateur entraîné sur un petit nombre d'exemples pouvait enrichir son vocabulaire et améliorer sa capacité à classer correctement de nouveaux documents. L'approche proposée apparaît donc comme une piste de solution alternative à l'usage d'ensembles d'entraînement volumineux et coûteux à construire. Elle peut également inspirer le développement de moyens similaires de répondre au même problème.

Bibliographie

[Aiz01] A. Aizawa. Linguistic Techniques to Improve the Performance of Automatic Text Categorization. *Proceedings of NLPRS-01, 6th Natural Language Processing Pacific Rim Symposium*, pp. 307-314, 2001.

[BGMM02] J. Brank, M. Grobelnik, N. Milic-Frayling & D. Mladenic. Interaction of Feature Selection Methods and Linear Classification Models. *Proceedings of ICML-02, 19th Conference on Machine Learning, Workshop on Text Learning*, Sydney, Australia, 2002.

[BM98a] L.D. Baker & A.K. McCallum. Distributional Clustering of Words for Text Classification. *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval,* pp. 96-103, ACM Press, New York, US, 1998.

[BM98b] A. Blum & T. Mitchell. Combining Labeled and Unlabeled Data with Co-training. *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pp. 92–100, 1998.

[BMP01] R. Basili, A. Moschitti & M.T. Pazienza. NLP-driven IR : Evaluating Performances over a Text Classification task. *Proceedings of IJCAI-01, 17th International Joint Conference on Artificial Intelligence*, pp. 1286-1291, 2001.

[Cha01] R.P. Chaves. WordNet and Automated Text Summarization. *Proceedings of NLPRS-01, 6th Natural Language Processing Pacific Rim Symposium*, Tokyo, Japan, 2001.

[Cho88] Y. Choueka. Looking for Needles in a Haystack or Locating Interesting Collocational Expressions in Large Textual Databases. *Proceedings of the RIAO '88 Conference on User-Oriented Content-Based Text and Image Handling*, pp. 1-15, Cambridge, US, 1988.

[CH03] L. Cai & T. Hofmann. Text Categorization by Boosting Automatically Extracted Concepts. *Proceedings of SIGIR-03, 26th ACM International Conference on Research and Development in Information Retrieval*, pp. 182-189, ACM Press, New York, US, 2003.

[CH90] K.W. Church & P. Hanks. Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics*, 16(1), pp. 22-29, 1990.

[DLR77] A.P. Dempster, N.M. Laird & D.B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B*, 39, pp. 1-38, 1977.

[Dun93] T. Dunning. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19(1), pp. 61-74, 1993.

[Eve04] S. Evert. Association measures. http://www.collocations.de/AM/.

[FGLS99] N. Fuhr, N. Gövert, M. Lalmas & F. Sebastiani. Categorisation tool : Final prototype. Deliverable 4.3, TELEMATICS Project LE4-8303 "EUROSEARCH", Commission of the European Communities, 1999.

[GBUMG02] J.M. Gomez Hidalgo, M. de Buenaga Rodriguez, L.A. Urena Lopez, M.T. Martin Valdivia & M. Garcia Vega. Integrating Lexical Knowledge in Learning-Based Text Categorization. *Proceedings of JADT-02, 6th International Conference on the Statistical Analysis of Textual Data*, 2002.

[HBLH94] W. Hersh, C. Buckley, T.J. Leone & D. Hickman. Ohsumed : an Interactive Retrieval Evaluation and New Large Text Collection for Research. *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pp. 192-201, 1994.

[Joa98a] T. Joachims. Text Categorization with Support Vector Machines : Learning with Many Relevant Features. *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pp. 137-142, Springer Verlag, Heidelberg, DE, 1998.

[Joa98b] T. Joachims. Making Large-Scale SVM Learning Practical. *LS8-Report*, 24, Universität Dortmund, 1998.

[KPS02] Y. Ko, J. Park & J. Seo. Automatic Text Categorization using the Importance of Sentences. *Proceedings of COLING-02, 19th International Conference on Computational Linguistics*, 2002.

[Lew92] D.D. Lewis. Feature Selection and Feature Extraction for Text Categorization. *Proceedings of Speech and Natural Language Workshop,* pp. 212-217, Morgan Kaufmann, San Francisco, US, 1992.

[LJ98] Y.H. Li & A.K. Jain. Classification of text documents. *The Computer Journal*, 41(8), pp. 537-546, 1998.

[Lov68] J.B. Lovins, Development of a Stemming Algorithm. *Mechanical Translation and Computational Linguistics*, 11, pp. 22-31, 1968.

[LSCP96] D.D. Lewis, R.E. Schapire, J.P. Callan & R. Papka. Training Algorithms for Linear Text Classifiers. *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pp. 298-306, ACM Press, New York, US, 1996.

[LYRL04] D.D. Lewis, Y. Yang, T.G. Rose & F. Li. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, 5, pp. 361-397, 2004.

[Mar61] M. Maron. Automatic Indexing: An Experimental Inquiry. *Journal of the Association for Computing Machinery*, 8, pp. 404-417, 1961.

[Mit97] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[MN98] A.K. McCallum & K. Nigam. A Comparison of Event Models for Naive Bayes Text Classification. *ICML-98, Workshop on Learning for Text Categorization*, pp. 41-48, AAAI Press, 1998.

[MS99] C.D. Manning & H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[NMTM00] K. Nigam, A.K. McCallum, S. Thrun & T. Mitchell. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39(2/3), pp. 103-134, 2000.

[Pla99] J.C. Platt. Fast Training of Support Vector Machines Using Sequential Minimal Optimization. *Advances in Kernel Methods - Support Vector Learning*, MIT Press, pp. 185-208, 1999.

[Por80] M.F. Porter. An Algorithm for Suffix Stripping. *Program*, 14(3), pp. 130-37, 1980.

[RSW02] T.G. Rose, M. Stevenson & M. Whitehead. The Reuters Corpus Volume 1 - from Yesterday's News to Tomorrow's Language Resources. *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pp. 827-832, 2002.

[SB88] G. Salton & C. Buckley. Term-weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5), pp. 513-523, 1988.

[Seb99] F. Sebastiani. A Tutorial on Automated Text Categorisation. *Proceedings of ASAI-99, 1st Argentinian Symposium on Artificial Intelligence*, pp. 7-35, 1999.

[SM99] S. Scott & S. Matwin. Feature Engineering for Text Classification. *Proceedings of ICML-99, 16th International Conference on Machine Learning*, pp. 379-388, Morgan Kaufmann, San Francisco, US, 1999.

[Sou02] P. Soucy. *Adaptation de l'algorithme des k-voisins les plus proches pour la classification automatique de textes*. Mémoire de maîtrise. Université Laval. Janvier 2002.

[SSV00] F. Sebastiani, A. Sperduti & N. Valdambrini. An Improved Boosting Algorithm and its Application to Automated Text Categorization. *Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management*, pp. 78-85, McLean, US, 2000.

[Tur01] P.D. Turney. Mining the Web for Synonym : PMI-IR versus LSA on TOEFL. *Proceedings of ECML-2001, 12th European Conference on Machine Learning*, pp. 491-502, Freiburg, Germany, 2001.

[Vap95] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[Yan99] Y. Yang. An Evaluation of Statistical Approaches to Text Categorization. *Information Retrieval*, 1(1/2), pp. 69-90, 1999.

[Yan01] Y. Yang. A Study on Thresholding Strategies for Text Categorization. *Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval*, pp. 137-145, ACM Press, New York, US, 2001.

[YL99] Y. Yang & X. Liu. A Re-examination of Text Categorization Methods. *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pp. 42-49, ACM Press, New York, US, 1999.

[YP97] Y. Yang & J.O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pp. 412-420, Morgan Kaufmann, San Francisco, US, 1997.

[ZH01] S. Zelikovitz & H. Hirsh. Using LSI for Text Classification in the Presence of Background Text. *Proceedings of CIKM-01, 10th ACM International Conference on Information and Knowledge Management*, pp. 113-118, ACM Press, New York, US, 2001.

[ZH02] S. Zelikovitz & H. Hirsh. Integrating Background Knowledge into Nearest-Neighbor Text Classification. *Proceedings of the 6th European Conference on Case Based Reasoning*, 2002.

Annexe A

À la section 5.2, lors de la description de l'approche proposée par nos travaux, il a été mentionné que seulement un sous-ensemble des mots du vocabulaire du classificateur était considéré lors de l'analyse de cooccurrence. Ce sous-ensemble contient, pour chaque catégorie, le mot qui lui est le plus relié selon le test du χ^2.

La présente annexe illustre à l'aide de quelques exemples les mots qui ont été jugés représentatifs de certaines catégories selon ce critère, lors de nos expérimentations.

Le tableau suivant recense les résultats relatifs aux expériences sur le corpus WebKB utilisant 20, 40 et 60 documents. On peut constater que pour certaines catégories, le mot choisi change selon le nombre de documents d'entraînement. Il semble que moins il y a d'exemples, moins le test du χ^2 permet de déterminer le mot le plus représentatif.

Nom de catégorie	Mots jugés représentatifs		
	20 documents d'entraînement	40 documents d'entraînement	60 documents d'entraînement
Course	how	homework	course
Faculty	professor	professor	professor
Project	in	related	nnn (trois chiffres)
Student	am	student	student

Le deuxième tableau qui suit expose les résultats relatifs à une expérience sur le corpus RCV1 utilisant 500 documents d'entraînement. En comparant les noms de catégories et les mots qui ont été associés à ces catégories, on peut remarquer qu'à

première vue, le choix semble pertinent dans la plupart des cas. Notons que le tableau n'inclut qu'un échantillon constitué au hasard parmi les 103 catégories de la tâche, pour des raisons de simplicité.

Code de catégorie	Nom de catégorie	Mot jugé représentatif
C11	strategy / plans	plans
C15	performance	net
C17	funding / capital	shareholders
C18	ownership changes	stake
C22	new products / services	providers
C312	external markets	exports
C34	monopolies / competition	clears
E11	economic performance	grow
E121	money supply	circulation
E21	government finance	aboriginal
E41	employment / labour	wage
E51	trade / reserves	civilian
ECAT	economics	inflation
G154	monetary / economic	rand
G158	external relations	canadian
GDIS	disasters and accidents	strengthening
GJOB	labour issues	wage
GPOL	domestic politics	house
GVIO	war, civil war	police
M12	bond markets	yield
M141	soft commodities	tonne

Annexe B

Lorsqu'il a été question à la section 5.2 des mots inconnus du classificateur traités par l'analyse de cooccurrence, on a pu constater que nos travaux les ont limités à ceux qui apparaissaient dans au moins quatre documents de test. En fait, la valeur de ce seuil a été fixée de façon empirique.

On a pris en considération la fréquence des mots inconnus survenant dans l'ensemble de test. L'observation des résultats de quelques expériences a permis de s'apercevoir que dès que l'on exige une fréquence minimale d'apparition dans deux documents, le nombre de mots inconnus à traiter chute de façon importante. Cette baisse du nombre de mots continue à se faire lorsque l'on passe à des fréquences minimales de trois ou quatre documents. Pour des valeurs supérieures, le nombre de documents continue évidemment de baisser, mais de façon moins marquée.

Comme il semblait que le seuil de quatre documents permettait de sélectionner un nombre raisonnable de mots à inclure dans l'analyse de cooccurrence, notre choix s'y est arrêté.

Le tableau suivant illustre le phénomène qui vient d'être décrit à l'aide de données provenant de tests effectués sur les corpus RCV1, WebKB et Ohsumed.

RCV1 (100 doc. d'entraînement) (10 000 documents de test)		WebKB (20 doc. d'entraînement) (3094 documents de test)		Ohsumed (500 doc. d'entraînement) (3632 documents de test)	
Fréquence d'apparition	Nb de mots inconnus	Fréquence d'apparition	Nb de mots inconnus	Fréquence d'apparition	Nb de mots inconnus
1	48286	1	35916	1	14051
2	28790	2	16021	2	7582
3	21689	3	11166	3	5515
4	17527	4	8754	4	4375
5	14899	5	7364	5	3624
6	13099	6	6431	6	3108
7	11757	7	5740	7	2680
8	10739	8	5210	8	2374
9	9910	9	4777	9	2098
10	9141	10	4446	10	1853