THÈSE DE DOCTORAT DE L'UNIVERSITÉ PARIS 6

Spécialité: **Informatique**

Présentée pour obtenir le titre de
DOCTEUR DE L'UNIVERSITÉ PARIS 6

par

Isabelle MOULINIER

Sujet de la thèse :

Une approche de la catégorisation de textes par l'apprentissage symbolique

Soutenue le 29 novembre 1996

devant le jury composé de :

M. Yves CHIARAMELLA	Rapporteur
M. Patrick GALLINARI	Président
M. Jean-Gabriel GANASCIA	Directeur
M. Michel GONDRAN	Rapporteur
M. Benoît HABERT	Examinateur
M. David HULL	Examinateur
M. Jean-Sylvain LIÉNARD	Examinateur
M. Hamou MESSATFA	Examinateur

Remerciements

Une fois le mémoire rédigé, il reste encore une tâche délicate : les remerciements. Je prie tous ceux qui se sentent oubliés de me pardonner. L'oubli est involontaire.

En premier lieu, je souhaite témoigner au Professeur Jean-Gabriel Ganascia ma reconnaissance pour avoir accepté de diriger mes recherches durant ces quelques années. Il m'a toujours laissé une grande liberté pour poursuivre mes recherches, tout en dispensant des conseils intéressants aux moments importants.

Le Professeur Yves Chiaramella a accepté d'assurer la lourde tâche de rapporteur. Ses connaissances approfondies de la recherche documentaire m'ont permis de combler certaines lacunes. Je l'en remercie vivement.

Je sais également gré à Monsieur Michel Gondran d'être rapporteur de cette thèse et d'avoir trouvé le temps de discuter les différents aspects de mon mémoire.

Mes sincères remerciements vont au Professeur Gallinari, pour avoir assuré la présidence du jury. Sa gentillesse et son intérêt pour la catégorisation de textes en ont fait un interlocuteur attentif.

Je remercie tout particulièrement Monsieur Jean-Sylvain Liénard pour l'honneur qu'il m'a fait en acceptant d'être membre du jury de cette thèse, dont le sujet est assez éloigné de ses préoccupations habituelles.

Merci à Monsieur David Hull d'avoir bien voulu de faire parti de mon jury. Sa disponibilité et ses connaissances approfondies sur le filtrage et les méthodes de classification statistique ont été enrichissantes.

Je voudrais exprimer toute ma gratitude à Monsieur Benoît Habert, à la fois pour sa participation comme membre du jury et pour ses relectures critiques. Ses commentaires successifs ont grandement contribué à la lisibilité de ce manuscrit.

Si j'ai pu mener à bien cette thèse, c'est grâce à Monsieur Hamou Messatfa, qui m'a orientée vers la classification de données textuelles, lors de ma dernière année au Centre Scientifique de Paris d'IBM France. Il me fait également le plaisir de participer à mon jury. Qu'il en soit vivement remercié.

C'est en voyant une présentation de David Lewis que j'ai choisi de travailler sur la catégorisation de textes comme application de l'apprentissage. J'aurais souhaité qu'il participe à mon jury, mais la langue s'est avérée un obstacle insurmontable.
Dave, many thanks for your ever-lasting patience and your useful advice and comments all along this work.

Le Centre Scientifique de Paris d'IBM France m'a accueillie pendant trois ans. Je remercie les différents membres du centre que j'ai cotoyés pour leur disponibilité : l'équipe Prolog au grand complet, le centre de mathématiques appliquées et les «linguistes».

En parlant de linguistes, merci à l'Équipe de Linguistique Informatique de l'École

Normale Supérieure de Fontenay – Saint-Cloud ainsi qu'aux informaticiens de l'École pour leur accueil chaleureux. Les postes d'ATER que j'ai eus dans cet établissement et au Laboratoire de Recherches en Informatique d'Orsay m'ont permis de terminer cette thèse dans de bonnes conditions.

Je tiens également à remercier les différents membres du LAFORIA, qui ont sû créer un cadre et des conditions de travail propices à l'élaboration de cette thèse. En dehors des enseignants et chercheurs, je pense plus particulièrement à Jacqueline Le Baquer, Anne Bancel, Ghislaine Mary, Andrée Musial et Valérie Mangin.

Le mémoire ne serait pas ce qu'il est sans les relectures successives d'Adeline Nazarenko et de Florence d'Alche-Buc. Leurs encouragements et suggestions m'ont été très utiles. Merci à elles deux et également à Christophe Marsala et Jean-Daniel Zucker, qui ont assuré les lectures d'urgence.

L'équipe ACASA m'a offert un cadre de travail agréable. J'en remercie tous ses membres présents ou passés, en particulier Jérôme, dont j'ai exploité à outrance le système ENIGME, Gailius sans qui SCAR n'aurait jamais vu le jour, Mourad l'angliciste cinéphile et Frédéric.

Les thésards du LAFORIA, bien qu'éparpillés entre les tours 46-00, 45-46, 45-55 et 56-66, ont su créer une ambiance à la fois détendue et motivante. Je les en remercie. Je n'oublie pas notamment les tracasseries LaTeXiennes incessantes de Maria, qui a acheté son pardon grâce à de fabuleux tiramisu. Heureusement, un gourou TeX résidait non loin de là : merci Speedy, pour TeX et le reste ! Merci également aux exilés de la tour 45-46, Myni, l'affreux PH et Armelle.

Finalement, merci aux amis qui m'ont entouré de leur soutien et leur bonne humeur, tout au long de mes recherches. Je pense surtout à Sylvie Cauvin et Christophe Matiachoff, que je remercie de parcourir le monde avec moi, à Isabelle Viaud-Delmon et toute la bande pour les moments de détente et leur bonne humeur, à Titou et Gabriel. Enfin, merci à Günet, dont le soutien a vaincu les continents :
Günet'e uzaklardan mesaj ve mektupla verdiği moral desteği için çok teşekkür ederim.

À mes parents

Résumé

L'objectif principal des travaux présentés dans cette thèse est de déterminer si la classification automatique, en particulier au moyen de l'apprentissage symbolique supervisé, peut être appliquée à la catégorisation de documents, tâche qui permet d'affecter des catégories sémantiques à des documents en fonction de leur contenu.

Le mémoire s'articule autour de deux axes complémentaires. L'axe principal consiste à déterminer si les techniques d'apprentissage apportent des solutions aux préoccupations de la recherche documentaire, en particulier le filtrage de documents. Dans cette optique, il est nécessaire de prendre en compte les caractéristiques des collections textuelles, en particulier celles liées aux dimensions des données textuelles (plusieurs milliers d'exemples et des dizaines de milliers d'attributs), qui sortent du cadre classique des applications de l'apprentissage. Ce problème constitue le second axe de notre travail.

Pour des raisons de complexité, notre démarche est d'introduire une étape préalable de sélection d'attributs avant tout apprentissage. Ceci nous conduit à proposer une méthode originale de réduction, appelée SCAR, qui tire parti des caractéristiques des données textuelles. Nous comparons SCAR à deux méthodes classiques pour la réduction de dimension. Nous évaluons ces méthodes sur la collection Reuters-22 173, dont la taille est conséquente.

Nous nous intéressons ensuite aux liens entre le présupposé d'apprentissage (nature de l'apprentissage) et les données textuelles, dans l'optique d'une application au filtrage. Nos résultats montrent que, malgré la relative équivalence observée en moyenne en comparant les différents algorithmes proposés, on peut distinguer une réelle influence du présupposé d'apprentissage sur la performance de l'algorithme, lorsque celui est appliqué à certaines classes de problèmes que nous avons identifiées.

Mots clés : Apprentissage symbolique automatique, catégorisation de textes, réduction de dimensions, recherche documentaire

Résumé anglais

Text categorization: a symbolic approach

Abstract

Our aim in this dissertation is to assess whether classification, especially symbolic machine learning, may be applied to the text categorization task, *i.e.* the content-based assignement of categories to documents.

We focus on two complementary aspects. First, we investigate the extent to which learning techniques provide solutions for Information Retrieval problems, with an emphasis on document filtering. Then, we stress that textual data possess specificities outside the usual scope of machine learning applications. Indeed, such data involve thousands of exemples and tens of thousands of features.

For the sake of computational efficiency, we introduce a feature selection stage prior to the learning process. We thus propose the SCAR reduction method, which takes into account the specificities of textual data. We compare the SCAR method with two state-of-the-art approaches. Evalution is carried out on a large-sized collection: the Reuters-22,173 corpus.

Finally, we study the relationships between learning bias and textual data, in the context of filtering applications. We observe an overall equivalence between all learners. However, learning bias turns out to have a real impact on effectiveness, depending on the class of problems we identified.

Keywords: Machine learning, text categorization, feature reduction, information retrieval

Table des matières

Introduction

Chaque jour, la quantité d'information disponible croît inexorablement, en particulier en raison de l'essor des communications électroniques. Le public auquel de telles masses d'information sont destinées est aussi en constante évolution : le développement rapide de l'Internet en est l'exemple le plus marquant.

Les formes sous lesquelles cette information est disponible dépendent de son support de communication : les contributions scientifiques peuvent être publiées de manière traditionnelle, c'est-à-dire dans des journaux, ou de manière électronique ; les nouvelles financières ou politiques sont diffusées par des agences de presse et acheminées vers leurs clients grâce à un support électronique, tandis que ces derniers utilisent presse écrite, radio ou télévision pour les communiquer au grand public. Un *document* représente une source d'information et correspond généralement à un texte, un son, une image, une séquence animée.

Nous sommes entrés dans l'ère de l'information où l'utilisateur est submergé par une masse d'information. Il ne se pose plus la question d'accéder à l'information. En fait, son problème est à présent de trouver l'information dont il a besoin parmi toutes celles dont il dispose. Il apparaît donc essentiel de disposer d'outils permettant de classer, d'archiver, d'analyser et de diffuser ces éléments d'information. C'est l'objectif principal que se fixe la *recherche documentaire* [SM83]. Lorsqu'un utilisateur formule un besoin d'informations sous la forme d'une *requête* exprimée le plus souvent en langage naturel, le système de recherche documentaire s'attache à fournir une liste de documents pertinents dans une *collection* documentaire. Un document est jugé pertinent s'il contient un élément d'information recherché par l'utilisateur.

Idéalement, un système de recherche documentaire doit être capable de retrouver une information, qu'elle soit disponible sous la forme d'un texte, d'une image ou d'un son. Construire un tel système est extrêmement difficile. C'est pourquoi un grand nombre de travaux ne considère que la recherche à partir de textes, qui constitue déjà un problème difficile. Nous nous plaçons dans ce contexte textuel.

La recherche documentaire : un problème de classification

Si l'on conçoit la recherche documentaire comme une mise en correspondance entre requête et document, il est quasiment impossible d'envisager une mise en correspondance exacte, en raison de la nature par essence ambiguë et redondante du langage naturel. En fait, la recherche documentaire regroupe différentes tâches qui s'appuient sur des requêtes ou des profils, en particulier la recherche *ad-hoc* et le filtrage.

La recherche *ad-hoc* suppose qu'il existe une collection figée de documents. Lorsqu'un utilisateur exprime un besoin d'information au moyen d'une requête sous forme d'un court texte en langue naturelle, le système utilise uniquement cette dernière pour trouver les documents pertinents dans la collection, c'est-à-dire ceux qui apportent un élément de réponse au besoin d'information de l'utilisateur. Les documents trouvés sont ensuite classés et présentés à l'utilisateur en fonction de leur degré de pertinence. Un exemple de recherche *ad-hoc* est une recherche dans le catalogue en ligne d'une bibliothèque.

Le filtrage s'appuie sur la notion de profil pour diffuser des documents à un utilisateur donné : un profil représente les centres d'intérêt de cet utilisateur. Le filtrage s'appuie non seulement sur un profil, qui correspond à une requête pour la recherche *ad-hoc*, mais aussi sur un ensemble de documents déjà jugés par l'utilisateur qui a défini le profil. Cet ensemble est une source de connaissances qui aide à déterminer quels documents sont pertinents, c'est-à-dire les documents qui correspondent au profil donné par l'utilisateur. De plus, le filtrage est essentiellement un processus dynamique, dans la mesure où l'on traite des flots de documents échelonnés dans le temps. Un exemple de filtrage est la diffusion de dépêches par des agences de presse ou le filtrage de nouvelles électroniques.

Quelle que soit la tâche envisagée, trouver les documents pertinents peut être considéré comme un problème de classification. En somme, il s'agit de regrouper, pour chaque requête, les documents qui sont pertinents d'une part et ceux qui ne le sont pas d'autre part [Lew92, Hul94]. Le rôle de la classification est en effet de regrouper des objets ayant des caractéristiques semblables et de prédire l'appartenance d'un objet à une classe donnée en fonction de ses caractéristiques.

La catégorisation de textes représente également un problème de classification à partir de textes : il s'agit d'affecter une ou plusieurs catégories à des documents en fonction des informations qu'ils contiennent. Les catégories sont choisies parmi un ensemble prédéfini, par exemple les mots clés autorisés par un journal scientifique pour indiquer le contenu des articles publiés. La catégorisation de textes s'inscrit dans le cadre de la recherche documentaire dans la mesure où elle peut servir de support pour l'indexation, c'est-à-dire l'étape qui consiste à représenter le contenu des textes. Elle peut également aider à construire des systèmes de filtrage si l'on représente les profils définis par les utilisateurs comme un ensemble de catégories prédéfinies. Parmi les différentes approches de classification proposées pour la catégorisation de textes, on distingue principalement celles qui font appel à des techniques d'intelligence artificielle et celles dont l'origine est plutôt statistique.

Classification et intelligence artificielle

La classification joue, depuis l'Antiquité, un rôle majeur dans le raisonnement en permettant d'organiser des connaissances. Elle intervient dans de nombreux domaines (biologie, mathématiques appliquées, ...), mais correspond souvent à des notions distinctes. En particulier, on peut classer pour regrouper des objets proches et chercher leurs caractéristiques communes ou bien on peut construire une classification dans le but d'identifier des objets, et de prédire leur appartenance à une classe donnée. La classification automatique (en anglais *cluster analysis*) en analyse de données et le regroupement conceptuel en intelligence artificielle (IA) s'intéressent à regrouper des objets.

Nous nous concentrons sur la construction de classifications dans un but prédictif. La recherche documentaire correspond à un processus d'identification des documents qui sont pertinents pour une requête. Le terme classification correspond à la fois à un processus et au résultat de ce processus ; afin de différencier ces deux sens, nous employons les termes règle de classification ou *classifieur* pour le résultat.

On a proposé différentes approches pour la classification en IA. En particulier, un classifieur peut être construit manuellement, comme un système à base de connaissances. C'est l'approche adoptée par le système MYCIN pour le diagnostic [Sho76]. Cependant, de tels systèmes ont fait l'objet de critiques [Cla85], relatives principalement au coût associé à la mise au point de classifieurs complexes et à la difficulté de les faire évoluer dans le temps.

L'apprentissage symbolique automatique répond en partie à ces critiques, en automatisant l'acquisition des connaissances nécessaires pour construire un système à base de connaissances. Différentes approches sont regroupées sous le terme apprentissage symbolique [MCM83]. Nos travaux s'intéressent plus particulièrement à un type d'apprentissage particulier, l'apprentissage inductif ou apprentissage à partir d'exemples.

Cette notion d'apprentissage inductif n'est pas propre à l'IA. Elle est présente dans bien des approches d'analyse de données ou statistiques [DH73, Jam85]. Les idées sous-jacentes sont semblables dans les deux communautés. On dispose d'un ensemble d'objets divisé en plusieurs classes ; chaque objet est décrit par un certain nombre d'attributs. On connaît, de plus, pour certains objets la classe à laquelle ils appartiennent. On généralise ces exemples sous la forme d'une fonction, le classifieur. Cette fonction f sert ensuite à prédire la classe d'appartenance de tout objet non classé.

La distinction principale entre induction numérique et apprentissage symbolique inductif réside majoritairement dans l'expression de la fonction f : l'apprentissage symbolique favorise les expressions compréhensibles, telles que des règles de production ou des arbres de décision, tandis que la fonction f est une fonction numérique, qu'il est souvent difficile d'interpréter. L'apprentissage inductif, qu'il soit numérique ou symbolique, a été appliqué à différents types de problèmes de classification. La recherche documentaire et la catégorisation de textes représentent cependant des domaines d'application atypiques.

Apprentissage et catégorisation de textes

La catégorisation de textes a connu un renouveau au début des années 90. Les bonnes performances d'un système à base de connaissances [HW90] et la disponibilité d'une collection textuelle ont donné une impulsion aux techniques d'apprentissage pour la catégorisation [Lew92, ADW94a, Yan96b, CS96, MRG96].

Déterminer le contenu d'un texte, comme le requiert la catégorisation, n'est pas difficile pour un humain. Cette tâche l'est beaucoup plus pour une machine (cf. [Pit86, Sab89]). Les approches actuelles, développées en IA pour le traitement automatique du langage, essaient de représenter la structure profonde des textes, par exemple les relations syntaxiques ou sémantiques entre les mots. Elles font en général appel à des formalismes de représentation des connaissances tels que les réseaux sémantiques et des logiques classiques ou non, et s'appuient sur une analyse complète des documents. De telles méthodes sont souvent trop complexes pour être mises en œuvre dans le cadre d'un système de recherche documentaire [JR93] et cèdent le pas devant un traitement plus superficiel, qui s'appuie par exemple sur l'analyse de portions de textes. En fait, dans de nombreux travaux, ce traitement s'arrête au stade lexical : on reconnaît les mots qui composent les textes et on les utilise pour représenter les textes, sous forme vectorielle. Ces vecteurs servent ensuite d'appui à une modélisation mathématique de la recherche documentaire [1].

La catégorisation de textes présente un défi comme application de la classification et de l'apprentissage, en raison des dimensions qui la caractérisent. Intéressons nous tout d'abord à la représentation des textes. Une représentation vectorielle à partir des mots, si elle semble naturelle, possède divers inconvénients. En effet, le vocabulaire composé de termes apparaissant dans une collection documentaire, de quelques milliers à plusieurs centaines de milliers pour une collection de taille classique, est gigantesque comparativement au nombre d'attributs que l'on observe habituellement dans les applications de l'apprentissage. Dans un tel contexte, construire un classifieur pertinent et précis est une tâche délicate. Par ailleurs, cette difficulté à apprendre est amplifiée par la représentation à partir des mots, qui est en général redondante et ambiguë : redondante car un même concept peut être exprimé par différents mots, ambiguë car un mot peut posséder plusieurs sens, selon le contexte dans lequel il est employé. Le problème des dimensions est en général abordé par la réduction a priori du vocabulaire, avec l'idée de réduire la complexité de la représentation et par conséquent de faciliter l'apprentissage. Certaines techniques de réduction permettent de plus de diminuer la redondance de la représentation [SHP95]. L'ambiguïté reste un problème ouvert.

Les dimensions d'une collection textuelle se définissent aussi par le nombre de documents compris dans celle-ci. En général, ce nombre est plus élevé que les applications traditionnelles de l'apprentissage, mais la différence est moins frappante que précédemment. La taille globale (de quelques milliers à des centaines de milliers) représente à elle seule un attrait, en particulier pour les approches d'apprentissage symbolique. Mais le réel problème qui se pose est la qualité d'une telle collection pour l'apprentissage. Traditionnellement, en effet, on dispose d'exemples équitablement répartis entre les différentes catégories que l'on souhaite apprendre. Une collection textuelle n'a pas une telle régularité : la proportion de documents étiquetés par une catégorie donnée est en général insignifiante face à la taille totale de la collection. Il est alors délicat de bien décrire les textes possédant cette catégorie. Il est encore plus difficile de caractériser les textes auxquels cette catégorie n'est pas affectée, puisque leur seul point commun est justement l'absence de celle-ci.

1. On retrouve dans cet abandon des aspects sémantiques du traitement des textes la distinction que fait C. Fuchs entre linguistique informatique et informatique linguistique [Fuc93] : la recherche documentaire, en général, manipule des données textuelles sans chercher à construire ou évaluer des théories linguistiques et vérifie en cela les propriétés des applications en informatique linguistique.

Notre contribution

La majorité des approches de classification appliquées à la catégorisation de textes [Lew92, SHP95, Yan96a, ADW94a, CS96] sont soit numériques, soit symboliques mais s'appuyant sur des critères numériques. Nous nous démarquons de ces travaux en choisissant un point de vue différent. Nous proposons une approche symbolique pour la catégorisation de textes, aussi bien lors de la réduction du nombre de termes décrivant les documents que lors de la construction de classifieurs par apprentissage.

Nous nous intéressons d'abord à la réduction de vocabulaire, sans prendre en considération les problèmes de redondance et d'ambiguïté sémantique issus de la représentation à partir de mots. Nous montrons tout d'abord que la réduction du vocabulaire s'inscrit dans le cadre des travaux sur la sélection d'attributs, menés en apprentissage depuis quelques années. Nous proposons une méthode de réduction symbolique qui s'appuie sur deux hypothèses, la première relative à la représentation des exemples et de la fonction à apprendre, et la seconde relative à l'algorithme d'apprentissage utilisé en aval. Cette dernière hypothèse tend à favoriser la sélection des attributs qui seront utiles à l'algorithme en exploitant les propriétés de celui-ci lors de la réduction. La première hypothèse vient du constat qu'il est difficile de caractériser les documents qui ne sont pas étiquetés par une catégorie c donnée. Nous choisissons d'apprendre à reconnaître les documents possédant la catégorie c, en utilisant uniquement les mots qui apparaissent dans ces textes. Ceci revient à faire l'hypothèse que l'absence est un défaut dans les règles de classification comme dans la représentation des documents. Nous comparons ensuite notre approche à deux autres approches numériques : une approche classique en catégorisation de textes, et une approche proposée en apprentissage, que nous avons modifiée pour qu'elle respecte l'hypothèse de l'absence par défaut.

Nous cherchons ensuite à défricher un problème ouvert : étudier les liens entre la nature des données textuelles et les caractéristiques des algorithmes d'apprentissage. En effet, s'il existe un certain nombre d'approches de classification appliquées à la recherche documentaire, rares sont celles qui s'attachent à déterminer les propriétés des algorithmes proposés pour les données textuelles. De plus, l'intérêt apporté aux résultats cumulés sur toutes les catégories ne permettent pas de mettre en évidence les difficultés locales pour une méthode particulière. Or, du point de vue de l'apprentissage, chaque catégorie pose un problème distinct, puisque le nombre de documents étiquetés, le vocabulaire employé varient. Nous défendons donc l'idée qu'une comparaison entre différentes méthodes doit s'appuyer sur une étude catégorie par catégorie des comportements de chacune pour compléter les résultats cumulés. Nous choisissons une représentation commune, en employant l'approche de réduction classique en catégorisation. Nous décomposons notre comparaison en deux étapes. Nous faisons tout d'abord évoluer la représentation des règles de classification en employant le même algorithme d'apprentissage. Ceci nous permet d'évaluer le coût de cette représentation. Puis nous comparons quatre algorithmes d'apprentissage en étudiant leur comportement sur un sous-ensemble représentatif de catégories. Ceci nous permet d'esquisser une première caractérisation des liens entre nature des données et algorithmes.

Plan de la thèse

La thèse commence par introduire la catégorisation de textes. Nous proposons un cadre qui permet de modéliser la majorité des approches de la littérature. Ce cadre met en évidence deux étapes importantes, la représentation des textes et la construction de classifieurs. Un certain nombre d'auteurs se sont servis de la même collection documentaire pour évaluer leurs travaux, la collection Reuters-22173. Nous concluons le chapitre 1 par une présentation de cette collection, que nous emploierons, nous aussi, par la suite.

Les deux chapitres suivants présentent le contexte des deux étapes de la catégorisation. Le chapitre 2 introduit l'indexation documentaire, qui intervient lors de la représentation des textes. Ce chapitre se conclut par un rapide aperçu des méthodes de mise en correspondance entre requête et documents en recherche documentaire. Le chapitre 3 introduit l'apprentissage symbolique et son application à la classification. Nous montrons en particulier comme l'apprentissage peut être vu comme un problème de recherche et comment on peut contraindre cette recherche.

Nous présentons ensuite, dans le chapitre 4, la catégorisation comme un problème d'apprentissage en grandes dimensions, en établissant un parallèle avec l'extraction de connaissances dans les bases de données. Nous reprenons les travaux pour la sélection d'attributs en apprentissage et analysons leurs intérêts et limites dans le contexte de la catégorisation.

Le chapitre 5 s'intéresse à l'évaluation des classifieurs construits pour la catégorisation. Nous montrons que les techniques d'évaluation en apprentissage ne permettent pas de prendre en compte l'objectif d'un système de catégorisation. Nous concluons par une discussion sur les tests statistiques employés pour la comparaison entre différents systèmes.

Le chapitre 6 introduit les quatre méthodes de classification que nous évaluerons dans la suite de nos travaux. En particulier, nous présentons l'algorithme symbolique CHARADE, qui sert de support aux deux chapitres suivants.

En tenant compte des limites des approches pour la sélection d'attributs données précédemment, nous proposons une approche pour la réduction d'attributs appliquée à la catégorisation de textes. Nous faisons ici une hypothèse sur la représentation des textes, en n'utilisant que les mots présents dans les textes pour les décrire. Le chapitre 7 détaille l'algorithme SCAR, avant de présenter une variation sur la méthode de sélection d'attributs pertinents RELIEF et une méthode par filtrage classique en recherche documentaire.

Les deux derniers chapitres relatent les expérimentations que nous avons effectuées sur la collection Reuters-22173 et en analysent les résultats. Le chapitre 8 met l'accent sur l'influence du vocabulaire employé pour décrire les textes, c'est-à-dire la réduction des dimensions. Au chapitre 9, nous nous intéressons à l'aspect apprentissage. Nous évaluons différents formalismes pour exprimer les règles de classification. Cela nous permet aussi d'analyser le comportement de l'algorithme CHARADE. Nous complétons notre évaluation par l'étude de l'influence des contraintes employées pour réduire la recherche lors de l'apprentissage. Notre analyse des résultats obtenus insiste sur la différence de comportements des différentes méthodes au niveau de chaque catégorie.

Nous terminons notre exposé en tirant les conclusions de nos travaux et en présentant des perspectives de recherche. En particulier, nous montrons qu'une analyse locale est nécessaire pour comprendre les interférences entre sélection, apprentissage et données textuelles.

Chapitre 1

La catégorisation de textes

À l'ère de la communication électronique, il est vital de disposer d'outils automatiques permettant d'accéder facilement à l'information. Rechercher un ouvrage dans un catalogue électronique, naviguer sur le réseau Internet pour trouver une citation, suivre des discussions sur des listes ou par courrier électronique sont des activités de plus en plus courantes. Pour trouver une information souhaitée, il faut être capable de déterminer si un document est, ou non, intéressant.

Accéder à l'information, sous forme électronique, est le rôle que se fixe la recherche documentaire. Elle s'est attachée à proposer des méthodes pour classer, archiver, analyser et diffuser des informations apparaissant dans des documents. Lorsque ces documents sont des textes, recherche documentaire et traitement du langage naturel possèdent un intérêt commun : l'analyse des textes en vue d'extraire les informations pertinentes.

De nombreuses tâches de recherche documentaire peuvent s'exprimer comme un problème de *classification* [Lew92]. Nous employons le terme classification avec sa connotation anglo-saxonne, car les termes français discrimination ou classement ne représentent qu'un aspect du processus qui nous intéresse. La catégorisation de textes correspond à un problème de classification particulier. C'est la tâche qui vise à identifier les ou les sujets dont traite un document. Plus particulièrement, on suppose qu'il existe un ensemble prédéfini de sujets possibles, que l'on cherche alors à attribuer automatiquement à des textes. Ces sujets peuvent être des mots clés [HW90, LR94, FHL$^+$91], nous parlerons dans ce cas de *catégories*, ou bien des structures plus complexes [RJ91, LPY94, RL94].

1.1 Applications de la catégorisation de textes

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

L'étiquetage automatique de documents est la première application de la catégorisation de textes. Il s'agit d'associer un ensemble de catégories à des documents en fonction des informations qu'ils comportent. C'est par exemple le cas de l'affectation de mots clés à des articles scientifiques [FHL$^+$91] ou bien l'étiquetage de dépêches financières comme le propose [HW90]. Dans un tel contexte, la catégorisation de textes est un cas particulier de l'indexation documentaire automatique en langage contrôlé.

La catégorisation peut être un support pour différentes applications. C'est par exemple un support pour la diffusion sélective d'informations. Lors de la réception d'un document, l'outil choisit à quelles personnes faire parvenir celui-ci en fonction de leurs centres d'intérêt. Ces centres d'intérêt correspondent à des profils individuels [LPY94] et sont composés de plusieurs étiquettes, par exemple. Une deuxième illustration des applications de la catégorisation vient du domaine de la presse : les agences de presse segmentent (manuellement ou automatiquement) les informations en de grandes classes et les diffusent au moyen de canaux différents. Par exemple, l'Agence France Presse offre différents services traditionnels ou des services en lignes. Parmi les services classiques, AFP Economie, qui s'intéresse à l'actualité économique et financière, est un service à la carte qui permet aux abonnés de choisir parmi des rubriques spécialisées (par exemple agriculture, sociétés ou média) tandis que AFP Sports s'intéresse aux événements sportifs [1]. Les services en ligne sont de manière analogue divisés en AFP Direct et AFP Direct Sport. AFP Direct utilise huit rubriques pour classer l'information.

Finalement, on peut filtrer automatiquement un flux d'informations pour un même utilisateur. Par exemple, Cohen [Coh96a] propose, comme application de la catégorisation, de classer automatiquement dans des fichiers les courriers électroniques arrivant dans une boîte à lettres. Par ailleurs, certains agents logiciels, qui proposent des pages trouvées sur le Web en fonction des centres d'intérêt donnés par un utilisateur, peuvent aussi être considérés comme des applications de la catégorisation de textes [HH96].

1.2 Les approches en catégorisation de textes

Nous nous intéressons plus particulièrement à la catégorisation de textes comme illustration de l'indexation fondée sur un langage contrôlé. Dans ce cadre, deux types d'approches ont été proposés. Ces approches dépendent du choix d'un modèle pour représenter le moyen d'affecter une catégorie à un document et de la manière dont est engendré ce modèle. Le modèle construit peut être compréhensible, par exemple une base de règles, comme c'est le cas des systèmes experts ou des bases de règles construites automatiquement par apprentissage. Ou bien, c'est une fonction (au sens large) numérique calculée à partir de données servant d'exemples.

Plus qu'aux caractéristiques du modèle, nous nous intéressons à la façon dont il est construit : manuellement, à l'aide de techniques à base de connaissances, ou bien automatiquement par induction à partir des données.

1.2.1 Systèmes à bases de connaissances

Les premiers systèmes de catégorisation se sont inspirés des méthodes à base de connaissances telles qu'elles sont employées pour la classification et le diagnostic.

Le système CONSTRUE [HW90] est un système dédié à un domaine particulier : la catégorisation de dépêches financières. Cependant les idées sous-jacentes sont générales et ont été appliquées ensuite à d'autres domaines [HANS90]. Ce système se découpe en deux étapes : l'extraction de concepts et l'affectation des catégories. Un concept est défini

1. Ces informations sont disponible sur le serveur Web de l'AFP http://www.afp.com/.

par une base de règles, où chaque règle représente des associations lexicales, syntaxiques ou sémantiques de mots. Ceci permet de regrouper les différentes manières d'exprimer un même concept. L'importance de chaque concept est ensuite évaluée. Une seconde base de règles combine les concepts importants pour déterminer quelles catégories affecter. On trouvait déjà une approche similaire pour l'indexation en langage contrôlé chez [VS87].

L'approche proposée par Rau et Jacobs [JR90, RJ91] se distingue de la précédente par son ancrage dans le domaine du traitement automatique du langage naturel. Le premier composant est la construction d'un lexique des sens des mots ; ce lexique contient de plus des informations sur le contexte des mots, qui permettent de choisir un sens plutôt qu'un autre. Le texte ainsi découpé est ensuite «écrémé» [2]. La définition de patrons syntaxiques ou sémantiques permet d'apparier des éléments du texte à des structures prédéfinies que l'on souhaite compléter, d'affecter des catégories à des textes ou d'extraire des portions intéressantes pour un traitement linguistique ultérieur. Cette approche est plus générique que celle de Hayes *et al.* et a été testée pour l'extraction d'information comme pour l'indexation automatique [JR93]. Une approche similaire, à base de dictionnaires et de patrons syntaxiques, est proposée par Riloff et Lehnert [RL94].

Les performances obtenues par ces différentes approches sont très bonnes, lors de l'évaluation sur des ensembles de taille restreinte. Cependant le coût pour les mettre en œuvre est prohibitif (9 hommes/an pour CONSTRUE et 2 hommes/an pour la construction du lexique décrit dans [JR93]). De plus, un inconvénient majeur de ces outils est leur sensibilité aux variations du domaine. C'est pourquoi une construction automatique d'outils de catégorisation est proposée.

1.2.2 Construction inductive à partir des données

La solution couramment adoptée pour la construction automatique d'outils de catégorisation fait appel à l'apprentissage inductif : à partir d'un ensemble de documents étiquetés manuellement, on évalue une fonction qui modélise (on dit également généralise) ces documents.

Dans [Mou96], nous avons proposé un cadre permettant de décrire la majorité des travaux relatifs à la catégorisation de textes. Ce cadre s'articule autour de trois points, illustrés à la figure 1.1 :

- une représentation raisonnable des textes,
- une technique d'apprentissage et
- l'apport de connaissances additionnelles.

Nous décrivons brièvement comment chacun de ces composants est employé dans les systèmes de catégorisation existants. Une description complète des deux premiers composants sera donnée dans des chapitres ultérieurs

2. Traduction du terme anglais *text skimming*.

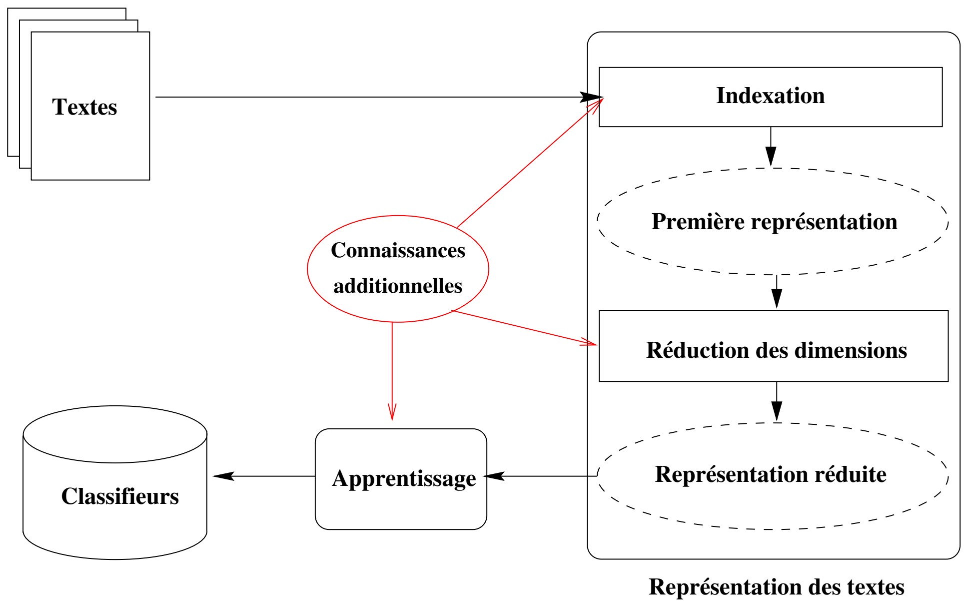


FIG. 1.1 - *Un cadre mettant en évidence les éléments importants dans un système de catégorisation de textes*

1.2.2.1 Une représentation raisonnable des textes

Avant tout autre traitement, il faut être capable de manipuler les textes de manière automatique. C'est l'un des rôles de la recherche documentaire. En particulier, l'*indexation documentaire* est le processus qui permet d'associer à chaque document une représentation. Ainsi un document est représenté par un ensemble non ordonné de termes. En général, un terme correspond à une chaîne de caractères. Chaque terme peut être brut (par exemple, le terme wells est extrait intact de la dépêche donnée à la figure 1.2) ou bien normalisé (le terme wells peut être remplacé par son singulier well). L'ensemble des termes issus de la collection de documents constitue la base de la représentation initiale des textes. Il s'agit ensuite de choisir quelle information conserver pour chaque terme : une information sur sa présence ou une information fréquentielle. Les travaux décrits dans [LR94, ADW94a, MG95, CS96] décrivent les textes comme des vecteurs booléens, c'est-à-dire qu'ils ne prennent en compte que la présence et l'absence des termes. Les approches proposées dans [FHL$^+$91, WPW95, Yan96b] font appel à une représentation basée sur la fréquence des termes.

Qu'entendre par raisonnable? Lorsque la collection de textes comporte plusieurs milliers de textes, il n'est pas rare que la représentation regroupe aussi plusieurs milliers de termes. Malgré l'évolution récente des quelques algorithmes d'apprentissage, en particulier ceux appliqués à l'extraction de connaissances dans les bases de données [HS94] ou à la biologie moléculaire [HSS93], des dimensions de plusieurs milliers de termes sont difficilement manipulables par la majorité des algorithmes d'apprentissage. Différentes approches ont donc introduit une étape supplémentaire dans la représentation des exemples : la *réduction des dimensions*. Notons que cette étape de réduction intervient après un processus d'indexation, où une première réduction est déjà réalisée, par exemple en éliminant les mots trop fréquents et les mots trop peu fréquents, ou en choisissant des termes normalisés.

Dans le contexte de la catégorisation de textes, nous définissons deux axes pour décrire les approches adoptées pour la réduction. Le premier axe s'intéresse aux termes sur lesquels portent la réduction, tandis que le second caractérise comment réaliser cette réduction. Avant de décrire ces deux axes, il est nécessaire de présenter le schéma retenu en général pour l'apprentissage. À chaque document sont associées aucune, une ou plusieurs catégories. Pour des raisons techniques puis historiques, de nombreuses approches ont choisi de construire un modèle par catégorie au lieu de construire un modèle général. Ceci influence la portée de la réduction, c'est-à-dire les termes pris en compte.

Lorsque l'on choisit de personnaliser la réduction des dimensions pour chaque catégorie, on parle alors de *réduction locale*. Une alternative consiste à décrire toutes les catégories en utilisant le même ensemble de termes, il s'agit alors d'une *réduction globale*. Les approches de [Mar61, ADW94a, BB63, Wie95] font appel à la réduction globale, tandis que [Lew92, WPW95, MRG96, SHP95, FHL$^+$91] choisissent une réduction locale. Finalement un nombre restreint de travaux [CS96, LSCP96, LC96, CMSW92] ne fait appel à aucun type de réduction particulier.

La réduction est en général mise en œuvre par deux mécanismes : le *filtrage* et la *construction*. Le filtrage vise à réduire le nombre de termes en sélectionnant les meilleurs selon un critère donné. Parmi les critères les plus fréquents se trouvent l'expertise humaine [Mar61, LPY94], l'information mutuelle [Lew92, MG95, Wie95], la mesure du χ^2 [SHP95, Wie95], la fréquence [BB63, ADW94a] ou le rang des termes [FHL$^+$91, WPW95]. L'aspect construction de la réduction est moins étudié. Au lieu de sélectionner un sous-ensemble de la représentation initiale, la réduction par construction crée de nouveaux termes en fonction des premiers. Le représentant d'une telle approche est l'indexation fondée sur la sémantique latente (ISL) [DDF$^+$90] , qui engendrent de nouveaux termes, correspondant à des combinaisons linéaires des termes de la représentation initiale. Elle est utilisé pour la catégorisation de textes dans [WPW95] et pour le routage de documents dans [SHP95].

Les travaux en catégorisation utilisent différentes techniques héritées de la recherche documentaire. Par exemple, l'ISL est à l'origine une méthode de recherche documentaire qui s'appuie sur la notion de décomposition selon les vecteur propres. Une telle approche est citée par Salton [Sal68] dès la fin des années soixante.

1.2.2.2 La construction inductive de classifieurs

Rappelons que la catégorisation est une tâche de classification. Nous appelons en conséquent les connaissances apprises par induction un classifieur. Une fois que la représentation réduite est fixée, les textes sont transformés en exemples d'apprentissage.

Différentes techniques sont possibles pour construire ces classifieurs de manière automatique. Nous distinguons les méthodes symboliques, qui permettent d'obtenir un modèle compréhensible aux approches numériques pour qui le classifieur est une fonction. De nombreuses approches numériques ont été proposées pour la catégorisation, en particulier la régression linéaire [BFL+88, FHL+91], les classifieurs Bayésiens [Mar61, Lew92], l'analyse factorielle [BB63] ou discriminante [Héb91], un algorithme implémentant la méthode des k plus proches voisins [MLW92, Yan94, LC96] ou de réseaux de neurones artificiels [WPW95, SHP95]. Quelques travaux, dont les nôtres, ont appliqué des techniques symboliques pour construire des classifieurs. Par exemple, [LR94, FHL+91, Mou96, MG95] utilisent des arbres de décision, tandis que [Coh95c, ADW94a, CS96, MG95, MRG96] engendrent des bases de règles.

Nous avons mis en évidence l'existence de nombreuses techniques d'apprentissage appliquées à la catégorisation. Nos travaux mettront l'accent sur l'application de l'apprentissage symbolique à la catégorisation de textes. Nous détaillons certains concepts mis en jeu dans le chapitre 3.

1.2.2.3 Quel est l'impact des connaissances additionnelles ?

Nous analysons à présent l'utilisation et l'impact de connaissances tout au long du processus de construction d'un système de catégorisation. Comme nous pouvons l'observer à la figure 1.1, des connaissances additionnelles peuvent intervenir dans les deux principaux composants de la catégorisation, c'est-à-dire la représentation des texte et l'induction.

Il n'existe pas une unique définition de connaissance. C'est pourquoi nous distinguons trois aspects différents. En recherche documentaire et dans les techniques d'apprentissage numériques, des connaissances peuvent être extraites des données. Elles interviennent en particulier dans la représentation des textes : choisir un modèle qui s'appuie sur les fréquence peut être considéré comme un apport de connaissances par rapport à un modèle ne s'intéressant qu'à la présence ou l'absence des termes. Nous appelons le deuxième aspect *connaissances du domaine*. Ce type de connaissances provient d'une interaction extérieure et fait référence à une application spécifique. Les dictionnaires électroniques sont par exemple source de connaissances du domaine. Finalement, contraindre la forme du système de catégorisation est la dernière facette du terme connaissance que nous souhaitons mentionner.

L'apport de connaissances additionnelles est surtout employé dans l'étape de représentation. L'emploi de la réduction locale, de l'ISL ou d'un modèle fréquentiel pour représenter les textes peut être perçu comme un ajout de connaissances extraites des données. Dans [LPY94], les auteurs font appel à des connaissances du domaine par l'utilisation d'un dictionnaire électronique pour engendrer la représentation initiale. Nous considérons que l'approche décrite par [ADW94a], qui prend en compte la structure des textes en pondérant les termes apparaissant dans le titre d'un document, est un exemple d'apport de connaissances du domaine. Finalement représenter les textes sous forme de relation est une contrainte sur le modèle appris. Cette approche est adoptée par [Coh95c] et permet de prendre en compte les positions relatives des termes, par opposition à l'approche générale qui consiste à ignorer cette information.

Peu de travaux font état de l'utilisation de connaissances durant la construction de classifieurs pour la catégorisation. L'approche décrite dans [WPW95, Wie95] fait cependant exception. Les auteurs ont regroupé les catégories en fonction de leurs caractéristiques sémantiques et utilisent ce découpage pour induire des classifieurs sur les sous-domaines définis par des groupes. L'approche de [FHL+91] utilise des connaissances du domaine pour guider l'apprentissage. Ces connaissances interviennent en aval de l'apprentissage et permettent de discriminer les catégories en fonction de consignes d'indexation prédéfinies.

La majorité des approches utilisant des connaissances, que ce soit lors de la représentation ou pour l'induction, montre que le système de catégorisation enrichi obtient de meilleures performances qu'une approche naïve. Cependant, peu de travaux ont proposés des expériences où l'apport de connaissances variaient. Notons cependant les résultats obtenus par [WPW95], qui rapportent une amélioration de 5% lors de l'emploi de l'ISL et d'un réseau hiérarchique, par rapport à un modèle booléen réduit et un réseau de neurones plat. Une augmentation du même ordre est observée par [ADW94a] lorsque les auteurs passent d'une représentation s'appuyant sur un modèle booléen et une réduction globale à une représentation fréquentielle obtenue par réduction locale et par pondération des termes apparaissant dans les titres des documents.

1.2.2.4 Limites du cadre proposé

Les limitations du cadre proposé sont de deux types : d'une part, certaines approches de catégorisation de textes n'entrent pas totalement dans le schéma, d'autre part, la diffusion sélective d'informations n'est pas entièrement modélisée.

Limitations pour les systèmes de catégorisation

Parmi les approches qui ne correspondent pas exactement au cadre de la figure 1.1, figurent celles où l'étape de réduction est absente [LSCP96, Yan94, CS96]. De manière plus marquée, l'approche de [CMSW92] remplace cette étape de réduction par une étape d'expansion de la représentation initiale : l'algorithme d'apprentissage doit manipuler quelques millions de termes, sur des super-calculateurs.

La réduction de l'ensemble des documents, par opposition à la réduction de la représentation, n'est pas prise en compte dans le cadre proposé. Une approche possible pour ce type de réduction est la technique d'échantillonnage proposée par [LC94, Yan96b] et employée par [Coh95c].

Catégorisation et diffusion sélective

Dans les applications de la catégorisation, nous avons mentionné la diffusion sélective d'information. La diffusion sélective, en recherche documentaire, recouvre à la fois le filtrage et le routage de documents. Employer la catégorisation de textes comme aide pour ces techniques est possible [LPY94]. Cependant, certains travaux s'intéressent directement à ces deux tâches de classification [Lew95, SHP95, HSP96]. Le problème à résoudre est similaire mais la classification ne correspond plus à l'affectation de catégories mais à la décision de diffuser un document à un utilisateur. De plus cela permet de prendre en compte la dimension temporelle de la diffusion d'informations. Par exemple, [LSCP96] propose un traitement au vol des documents, tandis que [BM96] propose un traitement par paquets tout en conservant un temps de diffusion efficace. Finalement le routage se caractérise par le type de réponse apporté par le système : plutôt que décider si un texte est ou non intéressant pour un utilisateur, il s'agit de fournir un certain nombre de textes, classés par degré de pertinence.

1.3 La collection Reuters-22 173 : un corpus de référence

La partie expérimentale de nos travaux s'appuie sur une base documentaire particulière : la collection Reuters-22 173, qui a servi de corpus d'expérimentation pour différentes approches évoquées précédemment. Cette collection, la base de dépêches Reuters-22 173 [3], est un ensemble de dépêches financières de l'agence Reuters. Depuis sa première utilisation comme domaine d'application du système CONSTRUE [HW90], elle a fait l'objet de nombreux travaux. Ces travaux s'orientent autour de la représentation initiale des textes [Lew92, Fin95], proposent des méthodes pour réduire le nombre d'éléments décrivant les textes [ADW94a, WPW95, Wie95, MRG96] ou bien évaluent des algorithmes inductifs, nouveaux ou classiques, dans le contexte de la catégorisation [LR94, WPW95, Gol96, Mou96, CS96, Yan96a]. Plus largement, cette collection est utilisée en recherche documentaire [San94] et en extraction de connaissances dans les bases de données textuelles [DFH96].

1.3.1 Les documents

Cette collection comporte 22 173 documents, étiquetés par des experts de l'agence Reuters en parallèle à l'élaboration du système CONSTRUE. Les documents sont tous des dépêches financières, composées d'une partie structurée et d'une partie libre (voir à la figure 1.2). La partie structurée renferme sept champs : le sujet ou catégorie de la dépêche, les pays, les personnages ayant un rôle financier, économique ou politique, les organisations, les entreprises concernées et finalement les marchés financiers. Chaque champ peut prendre plusieurs valeurs, parmi un ensemble de valeurs prédéfinies. Les catégories couvrent des aspects financiers variés tels que fusions et acquisitions d'entreprise, indicateurs économiques, matières premières, taux de change, ... C'est uniquement à ce champ que nous intéressons par la suite lorsque nous parlons de catégorie ou d'étiquette.

Certains documents ne sont pas étiquetés : quel rôle faut-il leur attribuer ? On ne sait pas si les dépêches qui ne possèdent pas de catégories n'en ont pas parce qu'aucune n'a été jugée pertinente, ou bien s'il s'agit d'un oubli de la part des experts. Les observations de S. Finch (communication personnelle) tendent à favoriser la seconde hypothèse. Face

3. Grâce P. Hayes et D. Lewis, cette collection est disponible par ftp anonyme sur le site `ciir-ftp.cs.umass.edu`, dans le répertoire `/pub/reuters1`.

PATTERN-ID 9517 **TRAINING-SET**
12-MAR-1987 18:44:48.38
TOPICS: *loan crude* **END-TOPICS**
PLACES: usa india **END-PLACES**
PEOPLE: END-PEOPLE
ORGS: worldbank **END-ORGS**
EXCHANGES: END-EXCHANGES
COMPANIES: END-COMPANIES
INDIA GETS 140 MLN DLR WORLD BANK LOAN
WASHINGTON, March 12 - The World Bank said it approved a 140 mln dlr loan for India to help lessen that country's dependence on imported oil and spur development of its own petroleum resources.
The bank said the loan will be used to boost production by injecting gas in the partially depleted Assam oil fields and to assist exploration in other areas, including drilling 10 exploratory wells.
The bank said the recipient of the 20-year loan will be Oil India Ltd (OIL), which is the smaller of two public Indian petroleum exploration and production companies.
Reuter

FIG. 1.2 - *Une dépêche extraite de la collection Reuters-22713*

à cette ambiguïté, nous avons choisi de ne pas retenir les textes non étiquetés lors de la transformation de la collection en base d'apprentissage. Une telle approche se retrouve dans les travaux de [ADW94a, Wie95, Yan96a, CS96]. La proportion de textes non étiquetés dépend de la liste des catégories valides (voir la section suivante). On peut considérer néanmoins qu'environ 40% des documents ne possèdent pas de catégorie. Nous les avons éliminés de nos expérimentations, ce qui réduit la collection à 11 664 dépêches.

On sépare habituellement la collection en deux ensembles, l'un servant de base d'apprentissage pour la construction d'un système de catégorisation, tandis que l'autre est conservé pour évaluer ce système. Nous avons adopté l'approche employée par [Lew92] : la séparation entre ces deux ensembles se fait sur la date d'émission des dépêches ; les dépêches datées avant le 7 avril 1987 composent l'ensemble d'apprentissage. Si l'on prend en compte la collection entière, il y a 14704 textes dédiés à l'apprentissage et 6746 réservés pour l'évaluation. 723 dépêches supplémentaires composent le reste du corpus et correspondent aux exemples sur lesquels le système CONSTRUE a été testé. Nous avons finalement adopté le partitionnement suivant :

- apprentissage : 7789 dépêches

- test : 3875 dépêches

Les dépêches comportent en moyenne 90 mots mais leur taille varie d'une seule phrase à plusieurs pages. Elles se distinguent également par leur forme : certaines, comme à la figure 1.2, relatent un événement financier dans un style descriptif tandis que d'autres fournissent des résultats sous forme tabulaire (voir la figure 1.3).

PATTERN-ID 15 **TRAINING-SET**
1-APR-1987 03:08:25.69
TOPICS: *earn* **END-TOPICS**
PLACES: japan **END-PLACES**
SKYLARK CO LTD <SKLK.T> 1986 YEAR
TOKYO, April 1 -
Group shr 65.44 yen vs 73.30
Net 4.48 billion vs 4.19 billion
Current 10.85 billion vs 9.77 billion
Operating 9.65 billion vs 9.54 billion
Sales 103.53 billion vs 94.39 billion
NOTE - Company forecast for current year is group shr 70.05 yen, net 4.80 billion, current 11.20 billion and sales 113 billion.
REUTER

FIG. 1.3 - *Une dépêche sous forme tabulaire*

	1	2	3	4	5	6	7	8	9	10	11	12	14	15
Total	9754	1364	310	108	67	27	10	6	7	3	3	2	2	1
Apprentissage	6580	880	189	63	39	20	6	4	4	2		1		1
Test	3174	484	121	45	28	7	4	2	3	1	3	1	2	

TAB. 1.1 - *Histogramme du nombre de catégories étiquetant les dépêches*

1.3.2 Les catégories

Les résultats publiés par P. Hayes et D. Lewis mentionnent 135 catégories, que nous appellerons catégories officielles[4]. Certaines dépêches sont étiquetées par des catégories n'appartenant pas à cette liste. C'est par exemple le cas de la dépêche 9517 à la figure 1.2, qui possède la catégorie *loan* (emprunts) qui n'appartient pas aux 135 catégories.

Le nombre de catégories affectées par dépêche varie de 1 à 15, mais le nombre moyen de catégories par document est seulement de 1.26. La table 1.1 montre la distribution du nombre de catégories, sur l'ensemble d'apprentissage, sur l'ensemble de test et sur la combinaison des deux.

Parmi ces 135 catégories, certaines n'ont pas de documents dans la partie du corpus réservé à l'apprentissage, d'autres n'ont pas de documents dans la partie dédiée à l'évaluation. Finalement, une quinzaine de catégories ne possèdent ni exemple d'apprentissage, ni exemple de test. Les tables 1.2 à 1.4 résument ces informations sur le nombre de documents étiquetés par chaque catégorie.

Nous appelons *fréquence* d'une catégorie le nombre de documents étiquetés par celle-ci. La fréquence des différentes catégories varie de façon notable. En particulier, les 5 catégories les plus fréquentes, soit moins de 4% des catégories, étiquettent plus de 72% des documents du corpus. De plus, les catégories *earn* (bénéfices des entreprises) et *acq*

4. À la première apparition d'une catégorie dans un chapitre, nous en rappelons le sens. La liste des catégories mentionnées dans ce mémoire est donnée à l'annexe C.

(fusions et achats d'entreprises) représentent à elles seules plus de la moitié du corpus.

Nous avons remarqué que le nombre moyen de catégories affectées à un document est inférieur à deux. Cependant, si l'on exclut les catégories *earn* et *acq*, ce nombre augmente de manière sensible. Nous pouvons l'observer au moyen des tables 1.2 à 1.4, où les dernières colonnes relatent le nombre de fois où la catégorie n'est pas affectée comme unique étiquette à un texte. Certaines catégories dont *dlr* (dollar), *palm-oil* (huile de palme) , *wheat* (blé) ou *corn* (maïs) sont en grande majorité affectées en conjonction avec d'autres.

Cette propriété nous amène à étudier la proximité entre les catégories. Elle se décompose en deux aspects : une proximité sémantique et une proximité de fait. La proximité sémantique correspond à un regroupement conceptuel des étiquettes, par exemple les catégories appartenant à la famille des métaux comme *gold* (or), *silver* (argent), *lead* (plomb) ou *copper* (cuivre). La proximité de fait est celle traduite dans les tables 1.2 à 1.4. Considérons par exemple la catégorie *wheat*, de fréquence 212. Elle est affectée sur l'ensemble d'apprentissage 209 fois en même temps que *grain* (céréales), 54 fois en même temps que *corn*. Si l'on cherche à regrouper les catégories en fonction des documents auxquelles elles sont affectées (c'est-à-dire que l'on classe automatiquement les individus catégories en fonction des variables documents), *wheat* est à la fois proche de *grain* et de *corn*. Si l'on s'intéresse de plus aux termes apparaissant dans les documents, *wheat* est quasiment indifférentiable de *grain*.

1.3.3 Les termes employés

Comme nous l'avons mentionné lors de la présentation des différentes approches pour la catégorisation de textes, il est nécessaire d'engendrer une représentation initiale pour manipuler les dépêches de façon automatique. Pour la collection Reuters-22173, une telle représentation est disponible en même temps que les données brutes. Elle a été créée par une succession d'étapes simples [Lew92] :

- conversion de toutes les chaînes de caractères en minuscules,
- élimination de termes appartenant à un lexique des mots très fréquents,
- et élimination des termes ayant un nombre d'occurrences inférieur à 2 sur l'ensemble de la collection.

Lorsque les termes sont des mots simples, le nombre de termes à l'issue de la deuxième étape est de 80 936. Après l'élimination de la troisième étape, il en reste 22 791 pour représenter les 22 173 dépêches.

1.3.4 Intérêt et particularités de la collection Reuters

Les différentes catégories de la collection Reuters couvrent un certain nombre de problèmes susceptibles d'être caractéristiques de la tâche de catégorisation. En particulier, il faut être capable d'induire à partir de peu d'exemples. En effet, si la catégorisation est utilisée comme un outil pour le filtrage d'informations, il n'est pas envisageable de demander à l'utilisateur final d'étiqueter des milliers de documents, ni même de lui demander un jugement sur chaque document qui lui est adressé.

Un second point intéressant est la présence de catégories qui cooccurrent fréquemment. Ce type d'informations n'est que rarement pris en compte dans les différents travaux mais semblent être une caractéristique de la catégorisation de textes, qui l'oppose à une tâche de classification classique.

Pour conclure, soulignons que les différents travaux sur la collection Reuters n'emploient pas les mêmes bases d'apprentissage et d'évaluation. D'un côté, certains conservent toutes les dépêches [LR94, CS96], tandis que d'autres n'en gardent qu'une partie [ADW94a, WPW95, Yan96a]. L'ensemble conservé dépend du choix des catégories étudiées : les catégories officielles ou les catégories étiquetant effectivement la collection.

1.4 Conclusion

Nous avons proposé un cadre général pour la catégorisation de textes et indiqué les éléments particuliers aux différentes approches pour la catégorisation. Ces approches se fondent principalement sur des méthodes numériques. Notre objectif, dans ce mémoire, est de proposer une approche symbolique, tant pour l'apprentissage que pour la réduction de l'index.

Nous avons ensuite présenté la collection Reuters-22173, en insistant sur quelques particularités. Cette collection va nous servir à évaluer notre approche et à mettre à jour les liens qui existent entre apprentissage et données textuelles.

Nous replaçons à présent la catégorisation de textes dans le contexte de l'indexation documentaire, et présentons brièvement l'apprentissage symbolique.

Catégorie	Total	Apprent.	Test	Cooccur. Total(%)	Cooccur. Apprent.(%)	Cooccur. Test(%)
earn	4075	2877	1198	1	1	1
acq	2506	1651	855	4	3	4
money-fx	827	538	289	61	61	62
grain	646	433	213	92	91	94
crude	639	388	251	36	34	37
trade	569	369	200	34	32	39
interest	530	347	183	45	45	44
wheat	316	212	104	100	100	100
ship	308	198	110	47	44	53
corn	255	176	79	100	100	100
dlr	227	131	96	96	97	96
oilseed	197	122	75	95	93	99
money-supply	192	139	53	15	12	25
sugar	191	126	65	21	23	18
gnp	169	103	66	49	42	61
coffee	151	113	38	19	18	24
veg-oil	140	87	53	73	77	66
gold	138	94	44	26	24	30
nat-gas	133	75	58	64	67	60
soybean	124	78	46	100	100	100
bop	119	75	44	71	71	73
livestock	118	73	45	81	84	78
cpi	114	69	45	30	22	42
reserves	84	54	30	35	30	43
copper	80	47	33	31	34	27
yen	79	45	34	86	100	68
jobs	78	46	32	27	20	38
cocoa	78	56	22	17	16	18
carcass	77	50	27	84	86	81
rice	71	35	36	99	100	97
iron-steel	70	40	30	30	35	23
gas	67	37	30	69	73	63
ipi	66	42	24	24	19	33
cotton	65	39	26	57	62	50
alum	63	35	28	21	11	32
barley	55	36	19	100	100	100
rubber	54	37	17	20	16	29
meal-feed	52	30	22	85	80	91
palm-oil	44	29	15	100	100	100
zinc	44	21	23	59	62	57

TAB. 1.2 - *Liste des 118 catégories. Les 15 catégories non présentées correspondent à des lignes de 0. Pour chaque catégorie, nous indiquons le nombre d'exemples du corpus, de l'ensemble d'apprentissage et de l'ensemble de test étiquetés. Les trois dernières colonnes traduisent le pourcentage de textes où la catégorie est affectée en même temps qu'une autre catégorie.*

Catégorie	Total	Apprent.	Test	Cooccur. Total(%)	Cooccur. Apprent.(%)	Cooccur. Test(%)
pet-chem	42	20	22	50	35	64
sorghum	37	23	14	100	100	100
tin	37	18	19	8	6	11
silver	37	21	16	70	81	56
lead	36	15	21	72	73	71
rapeseed	34	17	17	100	100	100
strategic-metal	33	17	16	36	41	31
wpi	32	19	13	19	26	8
fuel	29	13	16	59	69	50
orange	29	16	13	24	19	31
hog	28	16	12	96	100	92
retail	27	23	4	19	17	25
soy-oil	27	14	13	100	100	100
heat	27	14	13	44	57	31
soy-meal	25	12	13	100	100	100
stg	22	18	4	95	94	100
housing	21	16	5	14	6	40
income	19	9	10	37	22	50
sunseed	18	11	7	100	100	100
lei	17	12	5	18	8	40
lumber	16	8	8	25	12	38
dmk	15	10	5	100	100	100
tea	15	9	6	60	78	33
platinum	13	5	8	69	80	62
nickel	12	8	4	58	62	50
oat	12	6	6	100	100	100
groundnut	10	5	5	80	100	60
l-cattle	9	6	3	89	83	100
sun-oil	9	5	4	100	100	100
rape-oil	8	5	3	100	100	100
jet	8	4	4	50	50	50
instal-debt	7	5	2	0	0	0
potato	7	3	4	14	33	0
naphtha	7	2	5	86	100	80
inventories	7	5	2	57	40	100
coconut-oil	6	4	2	100	100	100
propane	6	3	3	83	100	67
coconut	6	4	2	83	100	50
austdlr	4	4	0	100	100	–

TAB. 1.3 - *Liste des 118 catégories, avec leur fréquence et leur taux d'affectation simultanée (suite)*

Catégorie	Total	Apprent.	Test	Cooccur. Total(%)	Cooccur. Apprent.(%)	Cooccur. Test(%)
plywood	4	4	0	50	50	–
sfr	4	0	4	100	–	100
cpu	4	3	1	0	0	0
tapioca	4	3	1	100	100	100
pork-belly	4	3	1	100	100	100
palmkernel	3	2	1	100	100	100
nkr	3	1	2	100	100	100
sun-meal	3	1	2	100	100	100
f-cattle	3	0	3	100	–	100
can	3	3	0	100	100	–
palladium	3	2	1	100	100	100
lin-oil	3	1	2	100	100	100
groundnut-oil	3	1	2	100	100	100
cotton-oil	3	1	2	100	100	100
dfl	3	2	1	100	100	100
nzdlr	3	2	1	100	100	100
rand	3	2	1	67	50	100
lit	3	1	2	100	100	100
linseed	2	2		100	100	–
wool	2	2	0	50	50	–
rye	2	1	1	100	100	100
copra-cake	2	1	1	100	100	100
castor-oil	2	1	1	100	100	100
cornglutenfeed	2	2	0	100	100	–
fishmeal	2	2	0	100	100	–
skr	1	1	0	100	100	–
dkr	1	1	0	100	100	–
lin-meal	1	1	0	100	100	–
cottonseed	1	0	1	100	–	100
cruzado	1	1	0	100	100	–
saudriyal	1	0	1	100	–	100
red-bean	1	1	0	100	100	–
corn-oil	1	1	0	100	100	–
peseta	1	1	0	100	100	–
ringgit	1	1	0	100	100	–
castorseed	1	1	0	100	100	–
rupiah	1	1	0	100	100	–
bfr	1	0	1	100	–	100
hk	1	0	1	100	–	100

TAB. 1.4 - *Liste des 118 catégories, avec leur fréquence et leur taux d'affectation simultanée (fin)*

Chapitre 2

Indexation documentaire

Nos travaux s'articulent autour de la phase inductive de la catégorisation de textes. Dans ce contexte, ils s'appuient sur des composants responsables de l'analyse des textes et de création d'une représentation initiale. Ces composants réalisent des tâches traditionnelles de la recherche documentaire.

De plus, une application de la catégorisation consiste à représenter des documents à l'aide d'un vocabulaire prédéfini et fixe. C'est l'objectif de l'indexation à l'aide d'un vocabulaire contrôlé.

Nous décrivons brièvement les objectifs de la recherche documentaire et la place de l'indexation dans ce contexte. Nous présentons les traitements couramment mis en œuvre, après avoir défini quelques termes employés dans la suite du chapitre.

2.1 Indexation : la première étape de tout système de recherche documentaire

Avec l'essor des communications électroniques, avoir des outils pouvant gérer de grandes quantités de données est devenu nécessaire. L'agence américaine pour les projets de recherche avancés (DARPA) en a reconnu l'enjeu et a financé deux programmes d'évaluation de systèmes de recherche documentaire : les conférences TREC (recherche documentaire) [Har94, Har95] et MUC (compréhension de messages) [Sun91]. Ces deux conférences ont un objectif commun : l'évaluation des outils qui sont le fruit des travaux de recherche depuis plus de 20 ans. Si les conférences TREC s'intéressent à la recherche documentaire en général, les conférences MUC évaluent des systèmes de compréhension du langage naturel : il s'agit d'extraire des informations pertinentes à partir de textes en déterminant non seulement l'objet des documents (comme pour la catégorisation) mais aussi les différents acteurs.

De manière succincte, un système de recherche documentaire a un objectif simple : un utilisateur spécifie un besoin d'information sous la forme d'une requête exprimée le plus souvent en langage naturel ; le système recherche dans une base documentaire les documents qui lui paraissent les plus pertinents. À cette fin, le système s'appuie sur une analyse du contenu des textes pour déterminer le degré de pertinence de chaque document relativement à la requête. La réponse qu'apporte le système à la requête de l'utilisateur est une liste de documents classée selon le degré de pertinence [vR79, SM83, Sal89].

Pour construire une base documentaire électronique, il faut tout d'abord analyser les textes pour déterminer quels éléments d'informations sont à conserver. De même, pour apporter une réponse à un utilisateur, il faut analyser sa requête avant de la comparer aux différents textes de la base documentaire. L'étape qui permet d'analyser les textes et qui en construit une représentation manipulable pour toute recherche ultérieure s'appelle *indexation*.

Analyser un texte n'est pas chose facile. Généralement, cette tâche peut être réalisée à l'aide d'outils linguistiques, statistiques ou hybrides. En recherche documentaire, ce sont plutôt les traitements statistiques qui dominent. Ceci s'explique en grande partie par l'objectif du traitement et la nature des textes manupilés. Les documents sont en effet considérés comme des vecteurs convoyant de l'information : l'important est d'avoir accès à cette information, et non d'analyser la manière dont ces informations sont présentées. Un deuxième facteur en faveur des approches statistiques ou hybrides est plutôt historique : le domaine d'application des premiers systèmes de recherche documentaire était restreint à un type de documents, par exemple des dépêches financières, des résumés d'articles scientifiques. Dans un tel contexte, les problèmes d'ambiguïté sont limités. C'est pourquoi le traitement automatique des langues n'occupe qu'une place marginale en recherche documentaire. Des collections documentaires plus variées sont apparues avec la conférence TREC. Les approches mises en œuvre restent cependant similaires à celles utilisées sur des collections plus contraintes, privilégiant les analyses statistiques ou hybrides qui ont l'intérêt d'avoir un coût de calcul moins élevé que les approches linguistiques.

2.2 Indexation : quelques définitions

Le mot *index*, «indicateur» en latin, désigne à la fois le doigt qui sert à indiquer, à montrer, et un petit objet mobile servant à distinguer un document dans un ensemble. Au XVIIe siècle, le mot *index* est employé pour désigner une table alphabétique accompagnée de références. Ce dernier sens introduit une notion de classement, que nous retrouvons dans l'acception courante du terme *indexation*. L'indexation est définie comme l'action d'attribuer à un document des marques distinctives renseignant sur son contenu, en vue de le classer. Ainsi, l'objectif de l'indexation est de permettre de distinguer les documents les uns par rapport aux autres, grâce à l'analyse de leur contenu.

Le processus d'indexation vise à extraire les idées principales de chaque document de la base documentaire. Une idée est appelée *concept informatif* [1] ou *concept*. Un concept est une entité abstraite, qui, pour être manipulable automatiquement, est représentée par une forme textuelle (mot ou expression) ou par une forme complexe (graphe, patron). Nous appelons cette représentation *unité d'indexation* ou *unité*.

L'*index* d'un document est l'ensemble des unités d'indexation affectées à ce document. L'index d'une base documentaire est l'ensemble de tous les index associés aux différents documents de la base.

On distingue deux types d'index, l'index *libre* et l'index *contrôlé*. L'index est libre lorsque les unités qui le composent sont directement extraites des documents. Par opposition, il est dit contrôlé lorsque les unités composant l'index appartiennent à un vocabulaire

1. Ce terme est introduit par [Cha82] et apparaît avec un sens similaire chez [HW90].

PATTERN-ID 406 **TRAINING-SET**
1-APR-1987 13:59:24.80
TOPICS: *acq* **END-TOPICS**
PLACES: usa france **END-PLACES**
PEOPLE: chirac delors **END-PEOPLE**
CHIRAC SEES QUICK DECISION ON CGCT
WASHINGTON, April 1 - French Prime Minister Jacques Chirac told U.S. congressmen France would announce a decision in two to three weeks on which group is to control its second largest communications firm, Compagnie Generale de Construction Telphoniques, French sources said.
They said Chirac promised that France's decision would be based on financial, economic and technical criteria, and not on political grounds.
U.S. and German companies are the leading contenders to take control of CGCT.
The Reagan administration has warned France and Germany that it may retaliate if control of the company is awarded on political grounds.
Jacques Delors, President of the European Community's Executive Commission, called last month for control to go to West Germany's Siemens AG in order to strengthen European industry.
The other leading contender is a consortium of American Telephone and Telegraph Co <T> with Philips NV of the Netherlands.
Chirac said France would make public its reasons for awarding control, the sources said. He was asked about the company's fate on two occasions, in meetings with leaders of the Senate and House of Representatives.
Reuter

FIG. 2.1 - *Un exemple de dépêche*

administration	american	announce	chirac	commission	company
congressmen	contender	criteria	decision	delors	economic
financial	french	german	grounds	industry	european
political	retaliate	public	order	leaders	strengthen

FIG. 2.2 - *Extrait de l'index libre de la dépêche de la figure 2.1*

fixé a priori. Ce type de vocabulaire est appelé langage documentaire par J. Chaumier [Cha82] et est souvent présenté sous la forme d'un *thesaurus*. Dans le contexte de la catégorisation de textes, l'ensemble des catégories prédéfinies composent un langage documentaire simple. Un exemple d'index libre est présenté figure 2.2 pour la dépêche de la figure 2.1, tandis que l'unité *acq* (fusions/achats d'entreprise) compose l'index contrôlé de cette dépêche.

En résumé, l'indexation s'applique à lier à chaque texte les unités d'indexation qui traduisent son contenu. L'identification des concepts, leur représentation sous forme d'unités et leur rattachement aux textes est une tâche délicate, qui peut être réalisée de manière manuelle ou automatique.

2.3 Indexation manuelle

Pour indexer un document, une personne, appelée l'indexeur, parcourt ce document, synthétise les idées qu'il contient et choisit les unités d'indexation qu'elle juge représentatives de ces idées. Cela nécessite que l'indexeur parcoure intégralement le document afin d'identifier les concepts informatifs de ce texte. Les concepts sont ensuite représentés par des unités d'indexation. L'un des problèmes majeurs de l'approche est le choix du vocabulaire : deux indexeurs peuvent associer des unités différentes à un même texte. Plus critique encore, l'index associé à un texte par un même indexeur peut varier en fonction des connaissances de ce dernier. De plus, l'indexeur peut utiliser des termes différents à des instants différents [Ste65]. Ce comportement peut entraîner la création d'un index non pertinent, comportant des unités soit trop spécifiques, soit trop générales.

L'emploi d'un langage documentaire résout en grande partie ces problèmes, en réduisant le nombre d'unités d'indexation et en définissant précisément le sens de chaque unité. Dès l'apparition des premières techniques documentaires, au 19e siècle, une attention particulière a été accordée à la mise au point de langages documentaires. Par exemple, en 1876, Dewey proposa une classification pour répertorier livres et écrits dans une bibliothèque. La notion moderne de thesaurus est introduite par Roget avec son dictionnaire analogique "Thesaurus of English words and phrases", qui regroupe les mots, non par ordre alphabétique, mais suivant les idées qu'ils expriment.

L'indexation ne se restreint pas à l'extraction des unités ; elle évalue aussi l'importance relative de chaque unité dans un texte. Pour traduire cette importance, on attribue, pour chaque texte, un poids aux unités de l'index [Cha82]. Généralement, les poids se répartissent entre trois valeurs : très important, d'importance moyenne ou de peu d'importance.

L'indexation manuelle est un processus long et coûteux. Tout d'abord, elle nécessite une intervention humaine : même un indexeur expérimenté doit parcourir le document à indexer, travail qui peut s'avérer plus ou moins long. Elle requiert de plus la mise au point et la maintenance d'outils linguistiques. Ces coûts et la forte augmentation de la taille des textes à traiter ont provoqué l'essor des techniques automatiques pour l'indexation.

2.4 Indexation automatique

Les techniques mises en œuvre pour l'indexation automatique diffèrent si l'on s'appuie sur un langage documentaire ou si l'index est libre. La majorité des systèmes documentaires engendrent des index libres, mais l'on observe l'évolution de l'indexation contrôlée grâce aux approches à base de connaissances et à la catégorisation de textes.

2.4.1 Indexation libre

Afin d'accéder de manière efficace à l'information contenue dans les textes, un système documentaire classique construit un index de la collection et ne mémorise que les associations entre les termes de l'index et les documents. On parle alors d'index inverse, plus utile pour la recherche que les textes eux-mêmes. La taille de cet index inverse dépend du nombre de termes composant l'index et des documents.

L'indexation manuelle tend à distinguer concept et unité. Les approches automatiques pour l'indexation ont en général tendance à confondre concept et unité. Par analogie à l'approche manuelle, nous faisons émerger deux phases. Il s'agit tout d'abord d'extraire les unités d'indexation des textes de la collection. Il faut ensuite choisir un modèle permettant de représenter l'importance de chaque unité dans un texte.

2.4.1.1 Extraction des unités d'indexation

La première étape de tout traitement des textes est une analyse lexicale. Il s'agit en effet de transformer la longue chaîne de caractères que constitue un texte en éléments manipulables de manière individuelle. Nous appelons ces éléments primitifs des mots. Ils sont obtenus à partir des textes bruts de façon assez directe : toute suite de caractères entre deux séparateurs est considéré comme un mot. Les séparateurs sont l'espace et tout signe de ponctuation. L'analyse lexicale soulève certains problèmes : comment reconnaître un signe de ponctuation (le point par exemple), les noms propres, faut-il conserver les capitales ou normaliser tous les mots en minuscules ? Ces problèmes sont en général résolus simplement : par exemple, un point est une ponctuation s'il est suivi d'un espace et d'une lettre capitale ; un nom propre se distingue en commençant par une capitale. Les solutions apportées ne sont pas exactes mais suffisent en général à obtenir un système documentaire performant.

Les unités d'indexation peuvent être de mots, mais aussi des expressions ou des structures plus complexes comme des patrons ou des graphes. Les techniques que nous présentons dans l'approche statistique considèrent les mots comme unités d'indexation. Lors de l'emploi de méthodes linguistiques ou hybrides, ces unités sont en général plus structurées.

Approche statistique

Une majeur partie des systèmes documentaires s'appuie sur une analyse statistique du contenu des textes. Cette analyse vise à déterminer les mots à conserver comme unités d'indexation.

La fréquence d'un mot est un indicateur de son intérêt

L'article de P. Luhn [Luh58] pose les bases du traitement statistique des textes : il établit que la fréquence d'une unité dans un texte indique son pouvoir de représentation du contenu de ce texte. Le mécanisme le plus simple de sélection des unités consiste à mesurer les fréquences d'apparition des unités candidates et de ne conserver que certaines d'entre elles :

- Les unités les plus fréquentes ne sont pas révélatrices du contenu du texte. Ces unités sont, par exemple, les mots *chevilles* (*stop words* en anglais) tels que les prépositions, conjonctions ou articles.

- Les unités très peu fréquentes sont aussi à éliminer, car leur présence n'est pas un indicateur pertinent du contenu du document.
- Seules les unités ayant une fréquence ni trop élevée, ni trop peu élevée, sont conservées dans l'index. Elles sont considérées comme représentatives du contenu du texte.

Il existe d'autres approches pour réduire la taille de l'index tout en essayant de conserver une bonne représentation du contenu des textes.

Utilisation d'un anti-dictionnaire

Certains mots n'apportent aucune information en dehors du contexte dans lequel ils sont employés. C'est le cas de mots tels que «le», «et» ou «de», qui n'ont pas de contenu sémantique intrinsèque. Il parait donc naturel d'éliminer de ce type de mots de l'index : on utilise alors un anti-dictionnaire pour conserver l'ensemble des mots à éliminer.

Le nombre optimal de termes composant un anti-dictionnaire varie d'une dizaine à environ 500. Les systèmes actuels sont en général dans la partie haute de ce spectre. À la différence avec l'approche basée sur la fréquence des mots, un anti-dictionnaire mémorise des informations fréquentielles mais aussi sémantiques car certains mots informatifs ayant une fréquence très élevée peuvent ainsi être conservés dans l'index.

L'utilisation d'un anti-dictionnaire ne pénalise pas les performances d'un système documentaire, tout en permettant de conserver un index inverse de taille plus restreinte [FBY92]. C'est pourquoi la majorité des systèmes utilisent cette approche.

Normalisation

Lorsque l'on extrait les unités de façon naïve, le singulier et le pluriel d'un mot sont considérés comme deux entités distinctes. Une manière de remédier à la disparité que cela entraîne pour l'index est de considérer non par les mots comme unités mais leur radical. Le terme radical ou racine a un sens précis en linguistique. Ici, ce sens est plus vague. En effet, il ne s'agit pas de retrouver la forme normale associée à un mot (par exemple l'infinitif d'un verbe) mais plutôt d'unifier toutes les formes d'une même famille. Par exemple, «naviguer», «naviguant», «navigateur», «navigation» et même «navire» peuvent être représentés par «navire».

Cette normalisation peut être réalisée de façon statistique ou linguistique. Une première solution est de tronquer les mots après n caractères. Décider du nombre de caractères reste un problème ouvert. Une seconde approche, plus fréquente, consiste à établir une liste d'affixes (suffixes et préfixes) et d'éliminer ces affixes de chaque mot, de manière récursive (voir par exemple [Lov68, Por80, Pai94, Har91]). L'élimination des affixes est souvent dirigée par des règles *ad hoc* : par exemple il convient d'éliminer le suffixe «ation» de tout mot sauf de «nation».

L'extraction de radicaux s'apparente à l'analyse morphologique, traitement linguistique qui associe à chaque mot la forme canonique qui lui correspond. Cette analyse s'appuie sur l'utilisation de lexiques d'associations *mot - forme canonique* ou de dictionnaires électroniques. Une telle approche offre un intérêt restreint pour des langues qui présentent peu de diversité lexicale, comme l'anglais, mais a été employée pour le français dans un certain nombre de travaux [CD86, Ker85, BP86, Ber90, MB91, Arc95].

L'apport de la normalisation reste incertain et dépend grandement de la langue étudiée et des outils employés. Quelque soit la langue, il apparaît clairement que l'index inverse s'en trouve réduit. Cette réduction s'effectue au détriment de l'introduction d'ambiguïtés supplémentaires, par exemple entre un verbe (*e.g.* stating devient state) et un nom (state). Les résultats empiriques sont plus mitigés. Pour la langue anglaise, utiliser cette normalisation ne dégrade pas les performances d'un système documentaire ; par contre, si certains travaux observent une amélioration sensible, d'autres ne perçoivent peu de différence entre l'emploi ou non de la normalisation des unités [FBY92].

Regroupement des mots

On a souvent fait appel à des techniques de classification automatique (*clustering*) dans le but d'améliorer les réponses apportées par un système de recherche documentaire [SJ71, vR79, SM83]. Les éléments considérés pour la classification sont les documents ou les unités d'indexation.

Nous mentionnons ici brièvement l'emploi de la classification automatique pour regrouper les mots qui sont proches dans la collection étudiée. Les travaux de K. Sparck-Jones ont étudié le regroupement de termes simples, à partir de différents moyens de classification automatique [SJ71]. Les résultats obtenus sont mitigés et ont été confirmés par des travaux ultérieurs [Lew92]. Il est à noter que les regroupements sont numériques et non sémantiques.

L'emploi de techniques de classification automatique reste controversé : le processus est en effet coûteux et n'améliore pas systématiquement l'efficacité du système de recherche documentaire, comme le montre par exemple les travaux cités précédemment.

Approches hybrides : traitement automatique de la langue et statistiques

En raison de la nature des documents manipulés, des textes, il semble naturel de coupler recherche documentaire et traitement automatique de la langue (TAL). Cependant, le succès de ces approches mixtes est mitigé. A. Smeaton justifie cet échec relatif par l'utilisation faite du TAL [Sme95].

Utilisation d'expressions : représentation superficielle

La majorité des approches font appel au TAL comme une boite noire, utilisée pour extraire des unités d'indexation autres que de simples mots (expressions et sens de mots sont des exemples courants), que ce soit dans la représentation des documents ou de la requête.

Les méthodes employées peuvent se découper en trois groupes : les méthodes s'appuyant sur des grammaires stochastiques, celles s'appuyant sur une analyse syntaxique locale et celles implémentant une analyse syntaxique globale. Les travaux de [Lew92] sont un exemple du premier groupe. Dans [EMT92, EGWH$^+$91, DG83], les expressions correspondant à des patrons syntaxiques fixés sont extraites : par exemple trouver les expressions du type *article nom adjectif*. Quelques travaux se sont appuyés sur l'analyse globale des textes : [Fag87] utilise une grammaire mais ne retient que certaines expressions pour indexer un document ; dans [SvR88], seule la requête est analysée. Finalement, notons l'approche décrite dans [FHL$^+$91] qui considère des groupe de mots contigus comme unités complexes, même si ces entités ne sont pas syntaxiquement correctes. [Lew92, Sme95] observent que l'utilisation d'expressions comme unités d'indexation n'apportent pas d'amélioration notable sur les performances du système documentaire.

Parmi les approches précédentes, en particulier [EGWH$^+$91, JC95] utilisent des outils linguistiques construits automatiquement (un thesaurus calculé à partir de la collection). D'autres approches utilisent des outils linguistiques tels que des dictionnaires [LPY94, RJ91] ou des thesaurus pour représenter les textes et raffiner la requête.

Représentation profonde des textes

Les approches précédentes traitent en général des collections de grande taille. Différents travaux se sont intéressés à une recherche documentaire conceptuelle. Certains utilisent des techniques du TAL pour extraire les concepts et les représenter sous forme plus complexe, tel que des graphes conceptuels ou des *frames*. Les travaux décrits dans [Mau91, RJ91] utilisent des composants d'analyse morpho-lexicale ou syntaxique pour extraire les concepts des documents. Les systèmes proposés sont appliqués à des domaines restreints, où ils obtiennent une précision élevée, mais ne manipulent que des collections de volume faible par rapport aux systèmes proposés précédemment. La mise en œuvre de ces systèmes s'appuie sur des grandes bases de connaissances, construites en partie manuellement ou automatiquement. Dans [MB91], les auteurs proposent une approche multi-experts de l'analyse de textes, traitant parfois des aspects sémantiques : des patrons, correspondant à des groupes nominaux ou verbaux, sont mis à jour lors de l'analyse et constituent l'index.

Ces approches s'appuient sur des techniques de représentation des connaissances évoluées et appliquent des travaux relatifs au TAL. Cependant, elles ne semblent pas répondre totalement aux besoins des systèmes documentaires, en particulier en ce qui concerne la taille des bases documentaires étudiées.

2.4.1.2 Importance relative des unités d'indexation : pondération

L'importance des unités dépend en tout premier lieu de la requête, du besoin d'informations d'un utilisateur. La majorité des systèmes font cependant un choix *a priori* de l'intérêt des unités apparaissant dans les textes, par l'intermédiaire d'une pondération. Chaque document est alors représenté par un vecteur de poids.

Le cas le plus simple est rarement employé en recherche documentaire, mais on le trouve de manière assez fréquente en catégorisation de textes : il s'agit du modèle booléen, où les unités apparaissant dans un texte ont un poids de 1, tandis que les unités absentes ont un poids nul. Lors de la recherche, seuls les documents en total accord avec la requête pourront être récupérés. L'un des inconvénients principaux de cette représentation booléenne est qu'elle ne prend pas en compte la fréquence des unités.

Différentes solutions ont été proposées pour palier cet aspect, en proposant une pondération des unités. On peut séparer chaque poids en trois composantes. Au lieu de considérer la présence des termes, B. Croft [Cro81, Cro83] propose de prendre en compte la fréquence relative d'une unité, c'est-à-dire la fréquence d'une unité dans un texte à travers un *poids significatif* de l'unité (*term significance weight*) : en effet une unité qui apparaît souvent dans un texte a de grandes chances de bien distinguer ce texte des autres.

L'importance d'une unité n'est pas uniquement liée à sa fréquence relative, mais elle dépend aussi de sa fréquence globale, c'est-à-dire le nombre de fois où elle apparaît dans la collection. Les unités ayant une fréquence globale élevée peuvent intervenir dans de nombreux textes, et possèdent en conséquence un pouvoir caractéristique relativement faible. Par contre, une unité peu fréquente apparaît dans des textes très spécifiques et apporte ainsi une information intéressante. Pour prendre en compte cet aspect, on définit une composante du poids d'unité inversement proportionelle à sa fréquence globale [SJ72].

Dans les collections documentaires classiques, les documents étaient de structure et de longueur comparables. Les collections plus récentes n'offrent pas la même homogénéité (par exemple, les dépêches de la collection Reuters comportent entre 1 ligne et quelques pages de textes). Certaines approches introduisent un facteur pour normaliser le poids d'une unité en fonction de la longueur du document. Finalement, le poids de l'unité i dans le document j peut être s'exprimer de la façon suivante :

$$w_{ij} = \frac{f(f_{ij})log(N/n_i)}{g(|D_j|)}$$

où f_{ij} est la fréquence de l'unité i dans le texte j, n_i, la fréquence globale de l'unité i, N le nombre total de documents dans la collection et $|D_j|$ la longueur du document j. Le choix des fonctions f et g reste expérimental. Dans le cadre classique de la recherche documentaire, on a montré de manière expérimentale la supériorité du modèle pondéré par rapport au modèle booléen. Mentionnons brièvement un autre modèle pour la pondération des unités, qui est fortement dépendante du modèle de recherche en aval. Plutôt que de déterminer le degré de similarité entre la requête et un document, le modèle probabiliste évalue la probabilité qu'un document apporte une réponse pertinente à la requête. Nous renvoyons le lecteur aux articles suivants [Mar61, SM83, Fuh89, Cro83, Tur90] pour plus de détail sur l'indexation probabiliste.

2.4.2 Indexation contrôlée

Pendant des décennies, l'indexation manuelle s'est appuyée sur des langages documentaires. L'évolution de la taille des collections et la difficulté à maintenir les langages documentaires ont favorisé l'essor des techniques automatique d'indexation libre.

Depuis quelques temps, cependant, on a vu éclore différents travaux autour de l'indexation en langage contrôlé, en particulier avec le renouveau de la catégorisation de textes. Au chapitre précédent, nous avons présenté les travaux de [JR93] et [FHL+91], comme relevant de la catégorisation. En fait, ils font explicitement référence à l'indexation en langage contrôlé. L'approche de [FHL+91] utilise de plus un thesaurus électronique.

Les travaux de C. Jacquemin sont un autre exemple d'indexation contrôlée [JR94]. Ils mélangent TAL et utilisation d'un thesaurus : à partir des termes d'un thesaurus, l'auteur engendre des patrons syntaxiques locaux issus de dérivations des termes du thesaurus, afin d'indexer une base documentaire.

2.5 Qualité de l'indexation

Que l'index d'une collection soit engendré manuellement ou automatiquement, il est nécessaire d'évaluer sa qualité. Cette évaluation apparaît fortement subjective : deux indexeurs, qui n'affectent pas les mêmes unités à un document, ne porteront pas le même jugement sur un index créé par une tierce personne.

Certains auteurs ont proposé deux critères qualitatifs pour évaluer (et contrôler) l'indexation [Sal89, Cha82] : la *spécificité* et l'*exhaustivité*. La *spécificité* des unités d'indexation fait référence au degré de généralité des unités employées dans l'index, tandis que l'*exhaustivité*, encore appelée profondeur de l'indexation, mesure le nombre d'unités composant l'index et leur degré d'importance.

Ces deux critères sont en fait dépendant de l'objectif du système dans lequel est intégrée l'indexation. Par exemple, si l'on souhaite obtenir des informations très précises, on favorisera une indexation à la fois spécifique et exhaustive. Si le système est un outil de diffusion sélective de dépêches, l'indexation peut être assez générale, lorsque seules quelques grandes catégories existent.

L'exhaustivité et la spécificité restent des mesures informelles, difficilement quantifiables. C'est pourquoi une évaluation quantitative de l'indexation est en général effectuée au travers du système documentaire global : si l'indexation est bonne, on peut s'attendre à une augmentation des performances du système documentaire.

2.6 Stratégies de recherche en recherche documentaire

Nous concluons ce chapitre en présentant différentes stratégies de recherche employées en recherche documentaire . De façon schématique, on peut considérer trois éléments dans un système documentaire :

– la représentation des documents et de la requête, qui est assurée par l'indexation,
– le mécanisme de recherche,
– la prise en compte d'informations pour raffiner la recherche.

Le mécanisme de recherche est sans doute l'aspect de plus important d'un système de recherche : c'est à lui que revient la tâche de déterminer les documents pertinents pour une requête. On distingue traditionnellement trois modèles de recherche, qui s'appuient sur une représentation des textes comme des ensembles non ordonnés d'unités :

– le modèle booléen, qui recherche les documents qui s'apparient de manière exacte avec la requête,

- le modèle vectoriel, qui recherche les documents ayant un fort degré de similarité avec la requête,
- et le modèle probabiliste qui évalue la probabilité qu'un document soit pertinent.

Dans un cadre booléen, la requête est une expression logique utilisant les connecteurs ET, OU et NON. Une recherche efficace peut être mise en œuvre grâce à l'index inverse ; elle est encore souvent utilisée dans les systèmes de recherche bibliographiques. Ce modèle possède différents inconvénients. La recherche n'est pas robuste : elle dépend largement des unités employées dans la requête. D'autre part, le modèle booléen ne permet pas de prendre en compte l'importance des unités ni de classer les documents en fonction de leur pertinence.

Le modèle vectoriel pallie les deux derniers défauts du modèle booléen, en prenant en compte les poids des unités dans les textes et en classant les documents par degré décroissant de similarité. Si l'on représente la requête et les documents comme des vecteurs, la mesure de similarité la plus simple est le produit scalaire entre deux vecteurs. Il existe de nombreuses autres mesures [vR79, Sal89]. Ce modèle reste tout de même sensible aux unités employées dans la requête.

Pour rendre ce modèle moins sensible aux unités employées, de nombreux travaux proposent d'augmenter la requête. Cette expansion peut être guidée par un thesaurus ou par des associations entre termes calculées sur la collection documentaire (les regroupements de termes peuvent être employés à cet effet). Il s'agit alors d'exprimer le même concept avec des termes différents. Une seconde solution est de prendre en compte des jugements de pertinence donnés par l'utilisateur ayant formulé la requête. Une première recherche prend place ; l'utilisateur marque les textes qu'il juge pertinents. Cette information est ensuite utilisée pour reformuler la requête, en ajoutant ou retirant des unités, en modifiant les pondérations des éléments de la requête : on parle alors de reformulation automatique de la requête (*relevance feedback*) [Roc71]. La redondance dans l'index de la requête joue ainsi un rôle important pour renforcer la robustesse des système documentaires.

2.7 Conclusion

Nous avons brièvement introduit les objectifs de la recherche documentaire et les techniques mises en œuvre pour y répondre. Nous avons surtout accordé notre attention à l'indexation documentaire, car elle est étroitement liée à la catégorisation de textes. En effet, la catégorisation de textes correspond à un type d'indexation particulier, l'indexation avec un vocabulaire contrôlé. Ensuite, tout système de catégorisation est amené à manipuler des documents, et donc à les représenter. C'est précisément le rôle de l'indexation libre.

Nous présentons les différentes techniques d'apprentissage mises en œuvre dans le chapitre suivant, en supposant qu'une représentation a été choisie. Nous discuterons des problèmes liés à la représentation adoptée en recherche documentaire dans le chapitre 4, lorsque l'on veut utiliser des algorithmes classiques en apprentissage.

Chapitre 3

Apprentissage automatique et classification

Nous décrivons à présent la composante inductive de la catégorisation de textes. Nos travaux s'orientent autour de l'apprentissage automatique symbolique, c'est pourquoi nous présentons plus particulièrement cette facette de l'apprentissage.

3.1 Apprentissage et intelligence artificielle

L'apprentissage est une aptitude qui intrigue les scientifiques car elle est difficile à appréhender de manière précise. En fait, la compréhension des mécanismes mis en œuvre dépend du point de vue adopté : un psychologue, qui étudie le comportement lié à l'apprentissage, le définit comme la modification du comportement par la pratique, l'entraînement ou l'expérience ; un enseignant s'intéresse à l'apprentissage effectué par ses élèves, qui doivent acquérir des connaissances par étude, instruction ou expérience. Pour l'informaticien, la régularité et la reproductibilité des expériences sont centrales : il cherche à identifier les conditions nécessaires à l'apprentissage. Cette contrainte éloigne l'apprentissage automatique du comportement humain où une telle régularité est pour ainsi dire inexistante [Nat91].

Avant de modéliser le phénomène d'apprentissage, replaçons l'apprentissage automatique dans un cadre plus général, l'Intelligence Artificielle (IA). Dans l'un des articles fondateurs de l'IA [Tur59], Turing introduit l'apprentissage comme caractéristique essentielle à toute machine dont on dit qu'elle est intelligente et aborde les questions relatives aux modèles à adopter. La notion d'intelligence artificielle suscite en elle-même bien des questions. Définir la notion d'intelligence est largement en dehors de l'IA ; c'est pourquoi l'on s'intéresse le plus souvent aux machines qui imitent le comportement cognitif humain : peu importe si la machine ne pense pas réellement dès lors qu'elle en donne l'impression.

Les travaux actuels en apprentissage se scindent en deux catégories. Certaines recherches s'inscrivent directement dans la définition de l'intelligence artificielle proposée par Turing. C'est le cas des travaux sur la découverte [Len82, LS83] ou encore les travaux sur les algorithmes génétiques [Hol75, Gol89]. Mais la majorité des travaux poursuivent un objectif plus pragmatique. Le succès et l'industrialisation des systèmes experts ont rendu urgente la conception d'outils d'aide à l'acquisition de connaissances. Si, dans le cadre des systèmes experts, la programmation se ramène à l'expression de connaissances sous une forme déclarative, la mise en évidence des connaissances n'en demeure pas moins délicate. Certains outils peuvent extraire, d'eux-mêmes, les connaissances à partir d'exemples. Les contraintes que doivent satisfaire ces outils sont donc en retrait par rapport aux exigences formulées par Turing.

L'apprentissage symbolique n'est pas la seule approche possible pour extraire des connaissances à partir de données. De nombreux travaux statistiques ou connexionnistes ont des objectifs semblables. Malheureusement, l'hétérogénéité des formalismes et la différence des traditions rendent toute comparaison délicate. Certains travaux ont accordé un intérêt particulier à la construction d'un modèle mathématique du phénomène d'apprentissage [Nat91, GPH90, Val84], qui inclut à la fois l'apprentissage symbolique et les approches du connexionnisme et de l'analyse des données.

3.2 Exemples, concepts et descriptions

Lorsque l'on veut faire apprendre un concept à une machine, il faut lui décrire les différentes *entités* qu'elle doit manipuler. Par exemple, si l'on s'intéresse à la catégorisation de textes, ces entités seront des textes (cf. figure 3.1) ; si l'on veut reconnaître les ballons rouges, ces entités sont des objets proches du concept `ballon`. Souvent, le choix de l'entité à manipuler fige les *propriétés* décrivant cette entité. Par exemple, pour décrire un texte, on s'attache aux mots apparaissant dans ce texte ; pour trouver le concept `ballon rouge`, les propriétés de forme, taille et couleur peuvent être pertinentes. Chaque entité est alors caractérisée par les valeurs que prennent ces propriétés. Pour représenter un texte, on peut ne considérer que l'absence ou la présence des mots, ou bien leur fréquence dans le texte ; la propriété `couleur` prend ses valeurs dans l'ensemble {`rouge`, `vert`, `bleu`, `jaune`}.

En apprentissage, ces entités décrites pour la machine sont appelés *exemples*. Nous nommons *description* l'ensemble des associations *propriété – valeur* propre à un exemple. Par exemple, la description d'une dépêche comporte la propriété «la chaîne said est présente», qui prend la valeur *vrai* pour les dépêches 502 et 530 et la valeur *faux* pour la dépêche 529 (les dépêches sont données à la figure 3.1).

L'apprentissage symbolique inductif est aussi appelé *apprentissage de concept* , puisqu'il s'agit de déterminer le *concept* implicite à un ensemble d'exemples. On définit l'*extension* d'un concept comme la fonction caractéristique qui prend la valeur *vrai* pour les objets auxquels le concept s'applique et la valeur *faux* pour les autres objets. On parle d'*exemple positif* d'un concept si la valeur de la fonction caractéristique vaut *vrai* pour cet exemple ; l'exemple est dit *négatif* dans le cas contraire. Par exemple, les dépêches 502 et 530 sont des exemples positifs du concept «possède la catégorie *interest* (taux d'intérêts)» . L'*intention* d'un concept est l'ensemble des propriétés caractérisant le concept. C'est donc une description de ce concept. Le concept «possède la catégorie *interest*» peut être décrit par «la chaîne prime est présente». Ainsi, l'objectif de tout algorithme d'apprentissage est de donner la description d'un concept à partir des descriptions d'un sous-ensemble de l'extension de ce concept.

PATTERN-ID 502 **TRAINING-SET**
1-APR-1987 15:54:21.89
TOPICS: corp-news *interest* **END-TOPICS**
PLACES: usa **END-PLACES**

TCF BANKING AND SAVINGS HIKES PRIME RATE
MINNEAPOLIS, April 1 - TCF Banking and Savings FA $\boxed{\text{said}}$ it is raising its prime rate to 7-3/4 pct from 7-1/2 pct effective today.
Reuter

PATTERN-ID 529 **TRAINING-SET**
1-APR-1987 16:18:21.96
TOPICS: *earn* **END-TOPICS**
PLACES: canada **END-PLACES**

LAIDLAW <LDMFA> SPLITS STOCK, HIKES PAYOUT
Toronto, April 1 -
Three for two stock split
Pay and record date subject to shareholder confirmation May four
Qtly dividend 5-1/4 cts vs four cts
Pay May 15
Record May one
Note: dividends declared on pre-split shares.
Laidlaw Transportation Ltd
Reuter

PATTERN-ID 530 **TRAINING-SET**
1-APR-1987 16:19:04.90
TOPICS: corp-news *interest* **END-TOPICS**
PLACES: usa **END-PLACES**

SECURITY PACIFIC, PROVIDENT NATIONAL LIFT PRIME
NEW YORK, April 1 - Security Pacific National Bank and Provident National Bank $\boxed{\text{said}}$ they increased their prime lending rates a quarter point to 7-3/4 pct.
They $\boxed{\text{said}}$ the move is effective immediately.
Reuter

FIG. 3.1 - *Exemple élémentaire composé des dépêches* 502, 529 *et* 530 *issues du corpus Reuters*

3.3 Un modèle pour l'apprentissage

Le modèle que nous présentons ne s'appuie par sur une formalisation mathématique, à la différence des travaux mentionnés ci-dessus (voir [Nat91] par exemple). Il met l'accent sur les flux d'informations qui parcourent les systèmes d'apprentissage, plus qu'il ne détaille les techniques effectivement mises en œuvre. Ceci nous permet de reformuler les approches classiques tout en explicitant les particularités de quelques systèmes. Nous emploierons le terme *apprenant* [1] pour désigner les systèmes et algorithmes d'apprentissage. Le modèle est construit pas à pas, à partir d'un mécanisme élémentaire : la généralisation.

3.3.1 Généralisation : une étape nécessaire

Une solution triviale pour décrire un concept est de construire la disjonction des descriptions correspondant aux exemples positifs du concept. Elle présente cependant un certain nombre d'inconvénients. La description construite ne prend pas en compte les exemples négatifs. De plus, c'est un *apprentissage par cœur*, qui ne fait aucune généralisation [Mit82], car on mémorise la description de chaque exemple positif. Or ce type d'apprentissage est très éloigné de l'apprentissage humain, qui fait intervenir, entre autres, abstraction et analogie. C'est pourquoi un certain nombre d'apprenants s'appuient sur un mécanisme de généralisation : le concept appris est exprimé par une description généralisant les exemples positifs.

La détection empirique de régularités est une application directe de la généralisation. Elle vise à construire une base de connaissances en partant d'un ensemble d'exemples. C'est dans ce cadre que nous décrivons quelques techniques de généralisation.

Une approche simple pour la généralisation est mise en œuvre lorsque la description des exemples est exprimée en logique des propositions. La généralisation permet de trouver les propriétés communes aux exemples. Si les exemples sont exprimés sous forme de conjonctions de propositions, l'intersection des différentes propositions constitue une généralisation de ces exemples : $\mathsf{said} \wedge \mathsf{it}$ et $\mathsf{said} \wedge \mathsf{they}$ se généralisent en said. La technique mise en œuvre par un tel mécanisme est l'élimination d'un facteur conjonctif : soient P_1 et P_2 deux propositions, la proposition P_1 est plus générale que $P_1 \wedge P_2$.

Lorsque les exemples sont exprimés au moyen de la logique des prédicats, il faut ajouter à cette élimination la réification (transformation en variables) des constantes. Par exemple, étant donnés un prédicat *avant*, prédicat à deux places indiquant si un mot précède un autre mot, et les constantes said, it et they, $avant(\mathsf{said}, \mathsf{they})$ et $avant(\mathsf{said}, \mathsf{it})$ se généralisent en $avant(\mathsf{said}, X)$, où X est une variable. Soient P un prédicat à n places, $\vec{a}$ un vecteur de n constantes et $\vec{x}$ un vecteur comprenant au moins une variable, le terme $P(\vec{x})$ est plus général que $P(\vec{a})$. On appelle *généralisé* la description résultant d'un processus de généralisation.

Ces mécanismes de base permettent de formuler une première définition de l'apprentissage, illustrée par la figure 3.2 :

> *Étant données n descriptions d'exemples positifs et négatifs d'un concept C, trouver une description de C, c'est-à-dire une généralisation qui soit cohérente avec tous les exemples.*

1. Le terme *apprenant* a été choisi par analogie au terme anglais *learner*, fréquemment utilisé dans la littérature.

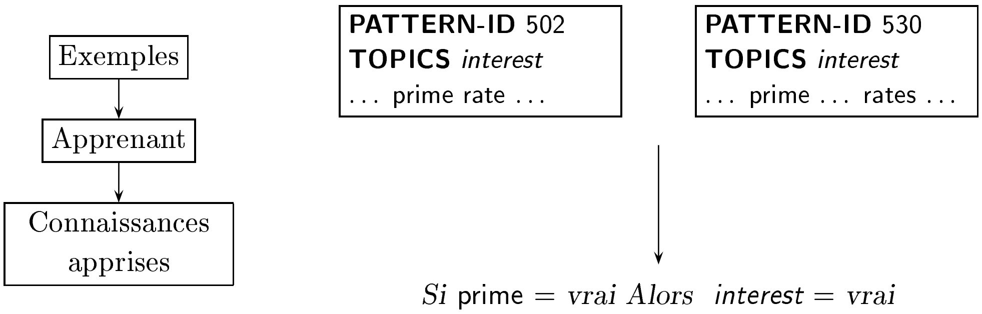


FIG. 3.2 - *Un premier modèle pour l'apprentissage s'appuyant sur la généralisation*

À partir des trois dépêches de la figure 3.1, une description du concept «possède la catégorie *interest*» traduit la propriété «le mot prime est présent»; la catégorie *earn* (bénéfices des entreprises) est décrite par «le mot dividend est présent». Notons que nous considérons les expressions «le concept «possède la catégorie *X*»» et «la catégorie *X*» comme synonymes.

3.3.2 Présupposé d'apprentissage

Les descriptions associées aux exemples permettent de définir de nombreux généralisés, qu'il est impensable d'envisager dans leur intégralité. Par exemple, considérons les dépêches 502 et 530 de la figure 3.1. Suivant les propriétés que l'on juge pertinentes, on obtient les généralisés suivants, parmi d'autres :

- une dépêche est constituée d'un titre et d'une histoire (cette information peut être indiquée par des marqueurs)
- les dépêches ont été rédigées le premier avril (April 1)
- le nombre de mots dans une dépêche est inférieur à 100
- les dépêches concernent les USA
- le mot prime apparaît dans le titre

Le nombre de généralisés, ou plus généralement la complexité du traitement, dépend de l'expressivité des descriptions des exemples. Plus l'on autorise une représentation des connaissances expressive, plus les algorithmes nécessaires pour manipuler ces connaissances sont complexes [LB85, Car89]. Le langage de description des exemples est appelé *langage des instances*. Dans la majorité des travaux en apprentissage symbolique, la logique (des propositions ou des prédicats) sert de base au langage des instances. Par exemple, les systèmes ID3 [Qui83], CHARADE [Gan87], CN2 [CN89] décrivent les exemples à l'aide de la logique des propositions. Les représentations vectorielles employées par les approches connexionnistes ou statistiques, comme la méthode des plus proches voisins, sont assimilées à la logique des propositions. En effet, chaque vecteur correspond à une

Dépêches Unités	502	529	530
tcf_banking_and_savings	v	f	f
hikes	v	v	f
prime	v	f	v
rate	v	f	f
said	v	f	v
pct	v	f	v
effective	v	f	v
laidlaw	f	f	v
splits	f	v	f
stock	f	v	f
payout	f	v	f
split	f	v	f
pay	f	v	f
shareholder	f	v	f
confirmation	f	v	f
qtly	f	v	f
dividends	f	v	f
cts	f	v	f
vs	f	v	f
shares	f	v	f
security pacific	f	f	v
provident	f	f	v
national bank	f	f	v
lift	f	f	v
lending	f	f	v
rates	f	f	v

Les propriétés retenues pour décrire les dépêches sont des unités d'indexation, mélangeant mot et expression. La valeur v associée à une unité indique la présence de celle-ci dans le texte.

TAB. 3.1 - *Représentation en logique des propositions des dépêches* 502*,* 529 *et* 530

description conjonctive où les littéraux élémentaires sont de la forme *attribut* = *valeur*. Un autre choix pour le langage des instances est la logique des prédicats. Ce type de langage est adopté par les systèmes FOIL [Qui90] et GOLEM [MF90], instances de systèmes de programmation logique inductive. La table 3.1 et la figure 3.3 montrent deux descriptions possibles des dépêches de la figure 3.1.

T. Mitchell reformule la généralisation comme une recherche dans un espace, appelé *espace de généralisation* ou *espace des hypothèses* [Mit82]. L'espace engendré par le langage des instances est vaste. Un parcours exhaustif de cet espace serait très coûteux. La reformulation de Mitchell propose donc de restreindre le nombre d'états étudiés dans cet espace.

Une première restriction possible est de contraindre la forme autorisée des généralisations. On se donne alors un langage de description des généralisations, appelé *langage des hypothèses*. Mitchell distinguait initialement le langage des instances et le langage des hypothèses. Dans la pratique, cependant, il y a souvent confusion entre les deux :

Nous adoptons une représentation relationnelle proposée par [Coh95c]. Chaque unité d'indexation correspond à un prédicat à deux places, indiquant la dépêche étudiée et la position du mot dans le texte, par exemple *prime*(502,3). Les prédicats *interest* et *earn* indiquent si les documents possèdent les catégories *interest* ou *earn*.

interest(502). *interest*(530). *earn*(529).

hikes(502,2).	*hikes*(529,4).	*prime*(502, 3).	*prime*(502,8).	*prime*(530,4).
prime(530,10).	*rate*(502, 4).	*rate*(502,9).	*rates*(530,12).	*said*(502,7).
said(530,9).	*said*(530,14).	*vs*(529,14).	*cts*(529,13).	*cts*(529,15).
pct(530,13).	*pct*(502,10).	...		

On peut de plus préciser les positions relatives des mots entre eux. Les mots sont représentés par leur position. La relation *proche* est rendue symétrique par sa définition en extension.

avant(1,2).	*avant*(2,3).	*avant*(3,4).	...	*avant*(n,$n+1$).
proche(1,3).	*proche*(2,4).	*proche*(3,5).	...	*proche*(n,$n+2$).
proche(3,1).	*proche*(4,2).	*proche*(5,3).	...	*proche*($n+2$,n).

FIG. 3.3 - *Représentation relationnelle des dépêches* 502, 529 *et* 530

les composants du langage des instances sont des composants élémentaires du langage des hypothèses. Parmi les langages des hypothèses employés classiquement en apprentissage, on trouve des arbres de décision, comme dans ID3 [Qui83], des listes de décision [PH90, Riv87, CN89], ou des systèmes de règles, comme dans INDUCE [DM81], AQ [MMHL86] ou CHARADE [Gan87]. Un exemple d'arbre de décision sera donné pour le concept «possède la catégorie *acq* (fusions/achats d'entreprises)» à la figure 6.2. La table 6.1 donnera un système de règles pour le concept «possède la catégorie *wheat* (blé)» : les littéraux apparaissant dans les règles sont des éléments du langage des instances.

Souvent, l'utilisation d'un langage des hypothèses ne suffit pas à restreindre l'espace de généralisation. Il est alors nécessaire d'ajouter des contraintes qui limitent la combinatoire de la recherche dans cet espace. L'ensemble de ces contraintes forment le *présupposé d'apprentissage* [Mit82], qui se décompose à son tour entre stratégie d'exploration et critères de choix entre plusieurs généralisations possibles.

La majorité des apprenants inductifs symboliques explorent l'espace de généralisation du général vers le spécifique : une description vide est peu à peu spécialisée jusqu'à l'obtention d'une généralisation cohérente des exemples du concept.

À chaque étape, il faut choisir dans quelle direction poursuivre le parcours. Ceci est résolu par différentes approches heuristiques. Les algorithmes construisant des arbres de décision, mais aussi certains apprenants de règles (comme FOIL ou RIPPER) choisissent à chaque étape l'attribut qui maximise un critère numérique, par exemple le gain d'information ou information mutuelle[2]. C'est une application de la méthode du gradient (*hill-climbing*) : les choix effectués ne sont jamais remis en cause pendant l'exploration. Les systèmes comme AQ ou CN2 conservent plus d'informations entre les étapes. Au lieu de sélectionner une unique généralisation à spécialiser, un certain nombre de descriptions potentiellement intéressantes sont conservées à chaque étape. C'est le principe de l'*étoile*, qui implémente une recherche en faisceau (*beam search*).

Un second groupe d'heuristiques entre en jeu lors de l'arrêt de l'exploration. Les premiers apprenants s'attachaient à trouver des généralisations cohérentes avec tous les exemples. C'est le cas de AQ, ID3 ou CHARADE. La robustesse d'une telle approche est remise en cause lorsque les exemples proviennent d'applications réelles. Certains systèmes s'arrêtent alors dès qu'un critère n'est pas satisfait, par exemple, lorsque le taux d'erreur sur la base d'apprentissage ne diminue pas de manière suffisante.

Le processus d'apprentissage met à présent en jeu non seulement des exemples du concept recherché, mais aussi un présupposé d'apprentissage, décomposé en deux parties : le langage des hypothèses, qui contraint la forme de la généralisation recherchée, et des heuristiques de parcours propres à chaque apprenant. Ceci nous permet de reformuler le modèle pour l'apprentissage (cf. figure 3.4) :

Étant donnés

- *un concept*
- *des exemples positifs et négatifs de ce concept*
- *un présupposé d'apprentissage (langage des hypothèses, heuristiques)*

Trouver une description du concept, qui soit compatible[3] *avec les exemples et construite en respectant le présupposé.*

3.3.3 Connaissances du domaine

On a souvent reproché à ID3 de ne pas pouvoir prendre en compte les connaissances du domaine, en particulier les relations implicites de la représentation et les relations sémantiques entre les différentes propriétés. L'apport de connaissances permet d'enrichir le langage des instances. La contrepartie d'un langage plus expressif est l'augmentation de la complexité associée à l'exploration de l'espace de généralisation. Prenons l'exemple de la figure 3.1. L'énoncé «les dépêches concernent les USA» relatif aux dépêches 502 et 530 ne peut être induit que si l'on sait que Minneapolis et New York sont des villes des États Unis. Si l'on ajoute de plus la dépêche 529, rédigée à Toronto, une caractéristique commune aux trois dépêches est leur provenance : l'Amérique du Nord. Les relations implicites entre les différentes valeurs des propriétés sont illustrées par exemple par la fréquence des mots. On veut pouvoir apprendre qu'une dépêche possède la catégorie *interest* si le mot said

2. Une fiche technique sur l'information mutuelle est donnée en annexe A.

3. Le terme compatible est employé pour exprimer la notion de cohérence dans la mesure où le présupposé d'apprentissage est respecté.

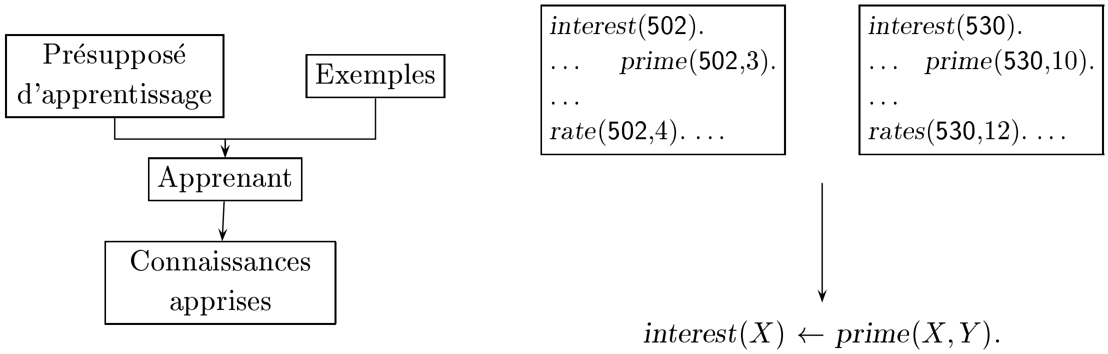


FIG. 3.4 - *Modèle avec présupposé d'apprentissage. L'exemple utilise un langage des instances et un langage des hypothèses relationnels.*

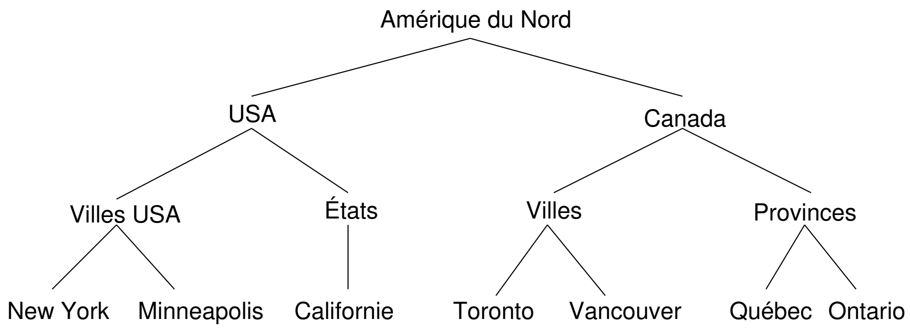


FIG. 3.5 - *La hiérarchie de sommet* Amérique du Nord *permet de définir les valeurs possibles de l'attribut* lieu_de_rédaction, *en prenant en compte des connaissances géographiques (limitées) pour le continent nord-américain.*

apparaît au moins une fois (une fois pour 502 et deux fois pour 530) dans le texte : on explicite ainsi la relation d'ordre entre les entiers.

Les connaissances du domaine sont définies dans le langage des instances. Mais elles n'ont d'application que lorsque le langage des hypothèses et le présupposé d'apprentissage permettent de les manipuler. Deux mécanismes sont habituellement proposés. Le premier s'applique aux relations entre les valeurs d'une propriété lorsqu'elles peuvent s'exprimer comme une hiérarchie conceptuelle (cf. figure 3.5). Il s'agit de remonter dans la hiérarchie conceptuelle définissant cette propriété : dans la hiérarchie définie à la figure 3.5, la description «la dépêche est rédigée aux USA» est plus générale que «la dépêche est rédigée à New York». Le second mécanisme porte sur les propriétés définies avec une relation d'ordre. Dans ce cas, on étend les bornes des expressions mettant en jeu ces propriétés. Les descriptions «le mot said apparaît une fois» et «le mot said apparaît deux fois» se généralisent en «la fréquence du mot said est comprise entre 1 et 2» ou bien en «le mot said apparaît au moins une fois».

Cet emploi de connaissances dans la génération de généralisés intervient principale-

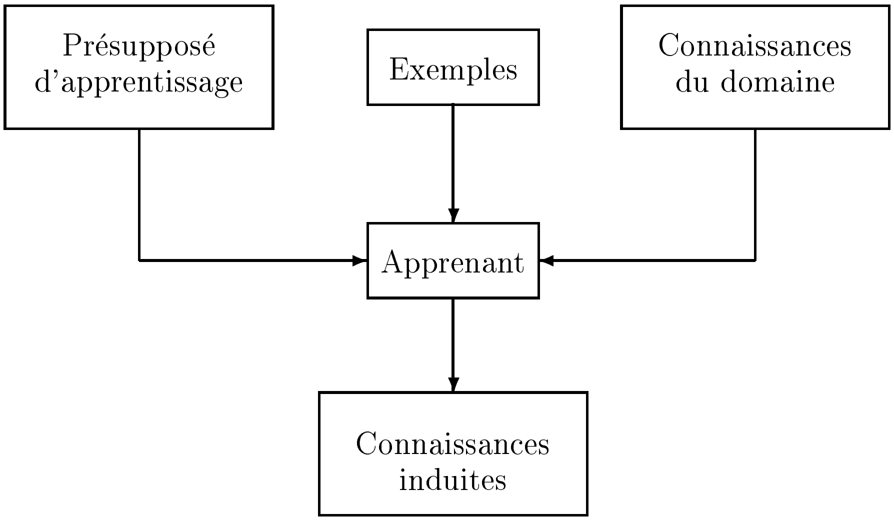


FIG. 3.6 - *Modèle avec connaissances du domaine*

ment dans les apprenants symboliques, dont le langage des hypothèses s'appuie sur des règles de décision. Il existe d'autres mécanismes, proposés pour les systèmes numériques. Dans le système KBANN, de [TS94], les auteurs expriment des connaissances sous forme logique et les introduisent dans un réseau de neurones. [NH93] propose différents modèles statistiques de bruit. Ces a priori sont ensuite pris en compte dans le calcul d'un classifieur Bayésien[4]. Dans cette dernière approche, les modèles représentent des connaissances explicites données à un apprenant numérique.

Une troisième définition de l'apprentissage, illustrée par le schéma 3.6, étend notre vue de l'apprentissage présentée précédemment.

Il existe un lien étroit entre le présupposé d'apprentissage et les connaissances du domaine. En effet, le présupposé inductif décrit la forme des connaissances acquises et la manière dont l'apprenant parcourt l'espace de généralisation pour obtenir ces connaissances. Ce lien est tellement fort que la majorité des travaux fait peu de différence entre présupposé et connaissances. Par exemple, dans [CM93], Clark et Matwin utilisent des connaissances du domaine, représentées par des modèles, pour limiter l'exploration de l'espace des hypothèses. Les approches de la programmation logique inductive utilisent aussi des connaissances pour limiter et guider l'exploration de l'espace des généralisés. Un cas particulier est de spécifier des contraintes entre divers éléments du langage des instances comme le font [PH90] et [Mic83]. On peut par exemple définir la relation $après(X,Y)$ en fonction de la relation *avant* ou encore la relation $entre(X,Y)$ par l'expression $entre(X,Y) \leftarrow X \leq Y, proche(X,Z), proche(Z,Y)$. Certains travaux ont considéré une théorie du domaine pour guider l'exploration [Coh92]. [PK92, MO94, BG93] proposent d'intégrer apprentissage inductif et apprentissage par explication en donnant une théorie du domaine initiale : il s'agit à la fois de guider l'exploration de l'espace des hypothèses tout en affinant les connaissances de la théorie donnée.

4. Brièvement, un classifieur Bayésien apprend un concept par estimation de la probabilité d'appartenance d'un exemple à ce concept, au vu de ses différents descripteurs.

3.3.4 Incrémentalité

Jusqu'ici, nous avons présenté l'apprentissage comme un processus direct, où l'on part de peu de connaissances pour en acquérir de nouvelles. Certains auteurs, dont [Utg89, MMHL86], ont proposé un apprentissage pas à pas, qui incorpore les connaissances acquises à chaque pas de l'apprentissage. Dans ce contexte, les exemples sont présentés un à un ; l'algorithme d'apprentissage raffine les connaissances apprises aux étapes précédentes en fonction du nouvel exemple. On parle alors d'apprentissage incrémental.

L'aspect incrémental est important pour les tâches de diffusion sélective telles que le routage ou le filtrage ; en revanche, la catégorisation de textes est généralement considérée comme une tâche en différé. Cependant, des travaux récents [LSCP96, CS96] appliquent des apprenants numériques incrémentaux à la catégorisation et au routage de documents.

3.3.5 Évaluation

Comme pour tout système, il faut être capable d'évaluer les performances d'un système d'apprentissage, c'est-à-dire la qualité des connaissances apprises. On peut distinguer deux aspects : une évaluation en cours d'apprentissage et une évaluation finale.

Le présupposé d'apprentissage correspond en partie à un moyen d'évaluation lorqu'il permet de décider l'arrêt ou la poursuite de l'apprentissage d'un concept. En particulier, certains apprenants évaluent le taux d'erreurs réalisé par le concept appris jusque là pour arrêter l'apprentissage. Ce taux d'erreur est calculé sur l'ensemble d'apprentissage ou sur l'ensemble de validation (partie de l'ensemble d'apprentissage réservée à cette fin). C'est le cas de RIPPER [Coh95a], de C4.5 [Qui92] ou de CART [BFOS84]. Notons que l'optimisation du taux d'erreurs est un présupposé particulier. L'approche proposée dans RIPPER va plus loin puisque le concept appris peut être remis en cause de manière ponctuelle en cours de construction ou totalement par une étape d'élagage.

L'évaluation finale peut être effectuée avec l'aide d'un expert, ou de manière automatique. Un expert peut émettre un jugement sur le concept appris lorsque celui-ci est exprimé sous une forme compréhensible, par exemple des règles de décision. Mais plus généralement l'évaluation est réalisée automatiquement : à partir d'un ensemble d'évaluation, différents critères mesurent les performances d'un système d'apprentissage. Le taux de classification correcte est le critère le plus fréquent, mais l'on peut aussi s'intéresser à la qualité du concept appris, par exemple la taille du concept ou sa compréhensibilité. Nous détaillerons l'évaluation finale d'un système d'apprentissage dans un chapitre ultérieur.

3.4 Discussion

Nous avons présenté les composants importants d'un système d'apprentissage symbolique. Pour la majorité d'entre eux, il existe un équivalent pour les systèmes d'apprentissage numériques. Par exemple, les apprenants numériques représentent les exemples comme des vecteurs, ce qui correspond à un langage des instances propositionnel. Le concept appris se représente sous forme de fonction. Les contraintes sur l'expression de cette fonction correspondent au langage des hypothèses : par exemple, on cherche à engendrer des classifieurs linéaires.

Notre vision de l'apprentissage est relativement simple et pragmatique : nous nous intéressons à l'apprentissage comme un outil. C'est pourquoi nous étudierons plus particulièrement l'influence du présupposé d'apprentissage lors de la construction de classifieurs dédiés à la catégorisation de textes.

Pendant de nombreuses années, la communauté de l'apprentissage a mis l'accent sur l'élaboration d'algorithmes plutôt que sur l'aspect réellement applicatif. Depuis quelques années, cette tendance semble s'inverser. Par exemple, des techniques classiques d'apprentissage sont appliquées à la découverte dans les bases de données, où il s'agit d'extraire par exemple des associations du type «les personnes qui achètent du jus d'orange achètent aussi des yogourts.» au sein d'une base de données relationnelle. Que ce soit la catégorisation ou la découverte dans les bases de données, ce nouveau type d'applications pose problème aux apprenants symboliques comme aux apprenants numériques. Dans le prochain chapitre, nous nous intéressons aux problèmes liés à la dimension et à leurs solutions.

Chapitre 4

Réduction des dimensions

La majorité des approches pour la catégorisation de textes présentées au chapitre 1 réduisent l'index initial lors la transformation des textes en exemples d'apprentissage. Pourquoi effectuer ce traitement ? Certains des algorithmes employés pour cette tâche ne sont pas capables de gérer des dizaines de milliers d'attributs : ils sont confrontés par exemple à des problèmes de sur-apprentissage, ne disposent pas d'assez d'exemples pour apprendre à étiqueter des textes de manière fiable ou sont calculatoirement trop coûteux.

Les motivations pour réduire l'index correspondent aux limitations de la composante apprentissage des approches en catégorisation. Différents travaux en apprentissage ont aussi souligné la nécessité de diminuer les dimensions d'un problème pour certaines applications. C'est l'objectif des travaux sur la sélection d'attributs pertinents.

Dans ce chapitre, nous analysons les approches pour la sélection d'attributs en apprentissage et replaçons la réduction des dimensions des systèmes de catégorisation dans ce contexte.

4.1 Applications en grandes dimensions de l'apprentissage

La communauté de l'apprentissage symbolique a donné naissance à de nombreux algorithmes, et a montré leur efficacité d'abord sur des cas d'école, regroupant bases construites artificiellement et données réelles. Le point commun de la majorité de ces applications est leur dimension : elles sont de taille relativement restreinte, au maximum un millier d'exemples décrits par une trentaine d'attributs. Certains travaux ont étudié des cas complexes, par exemple en apprenant à classer des images satellite [SCF$^+$95, FWD93]. La catégorisation de textes et l'extraction de connaissances dans les bases de données (ECBD) sont les deux applications réelles et en grandes dimensions des techniques d'apprentissage.

4.1.1 Données de grandes dimensions

La taille des données manipulées est un problème récurrent en apprentissage, que ce soit dans un cadre empirique ou théorique. Quand peut-on dire d'une application qu'elle est de grandes dimensions ? Nous définissons un espace à deux dimensions dans lequel positionner les différentes applications de l'apprentissage. Le premier axe de cet espace correspond au nombre d'attributs, le second au nombre d'exemples. Comme nous l'avons vu dans le chapitre 3, le nombre d'attributs a une influence sur le langage des hypothèses et donc sur la taille de l'espace de généralisation. Plus le nombre d'attributs est élevé, plus l'apprentissage est *a priori* complexe. Le nombre d'exemples a moins d'influence sur la complexité de l'apprentissage. En effet, les exemples contraignent l'exploration de l'espace de généralisation : par exemple, on ne développe la feuille d'un arbre de décision que lorsqu'il existe un nombre minimal d'exemples dans cette feuille. Cependant, le manque d'exemples peut dégrader la reconnaissance du concept : comment calculer de manière fiable les probabilités nécessaires à la construction d'un classifieur Bayésien en l'absence d'un nombre suffisant d'observations ?

Dans cet espace attribut-exemple, le comportement des apprenants en grandes dimensions évolue selon trois directions, par rapport aux cas d'école usuels :

- le nombre d'attributs reste stable, le nombre d'exemples augmente (région 1 de la figure 4.1).

  Depuis quelques années, il existe une collection de bases d'apprentissage, réelles ou artificielles, qui servent pour l'évaluation des nouvelles approches. Certaines bases ont des caractéristiques semblables : lorsque le nombre d'exemples est élevé (entre 300 et 5 000), le nombre d'attributs dépasse rarement la cinquantaine [MA94].

- le nombre d'attributs et le nombre d'exemples augmentent de façon relativement proportionnelle (région 2 de la figure 4.1).

  Ce cas correspond principalement aux applications potentielles de l'apprentissage à l'extraction de connaissance dans les bases de données (ECBD). Dans [HS94], une application de DBLearn s'appuie sur des centaines de milliers d'exemples décrits par environ 22 000 attributs. Même si l'on s'attend à manipuler des bases de données gigantesques (se mesurant en giga et tera-octets), on reste face à un problème où la proportion entre attributs et exemples est semblable au cas précédent.

- le nombre d'exemples augmente nettement moins rapidement que le nombre d'attributs (région 3 de la figure 4.1).

  Une telle configuration est rare dans les applications classiques de l'apprentissage, mais elle est classique dans les applications en recherche documentaire. Par exemple, une base standard de recherche documentaire, la base CACM, qui regroupe 1425 résumés d'articles de la revue *Communication of the ACM*, est indexée par plus de 40 000 unités (syntagmatiques) distinctes [Lew92].

Que ce soit en catégorisation de textes, en apprentissage ou en ECBD, il existe peu de travaux réellement menés sur de telles dimensions. Citons cependant les travaux récents en catégorisation [CS96, LSCP96] utilisant des bases de plusieurs milliers de documents, représentés par des dizaines de milliers d'unités. Les travaux décrits dans [CMSW92] (catégorisation de textes) et dans [CS93] (biologie moléculaire) utilisent des calculateurs parallèles pour manipuler plusieurs dizaines de milliers d'attributs.

La majorité des travaux en grandes dimensions cherche plutôt à diminuer la complexité du problème. On trouve là un point commun entre catégorisation et ECBD.

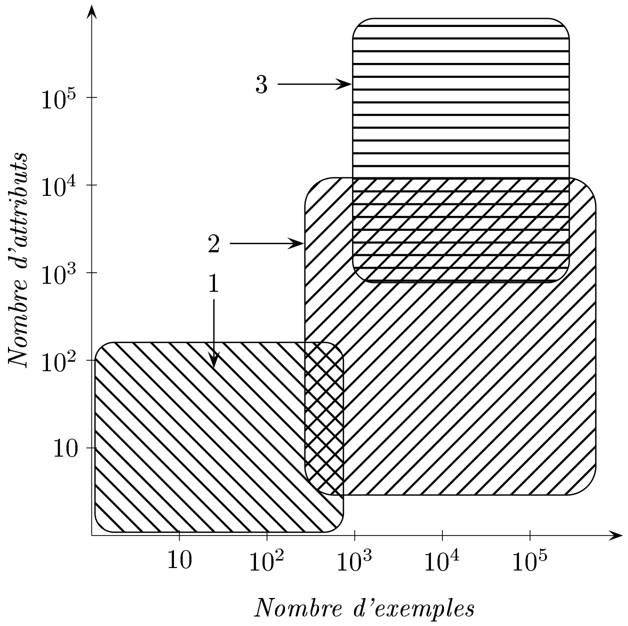


FIG. 4.1 - *Représentation plane des problèmes de dimension pour l'apprentissage.*

4.1.2 Parallèle entre catégorisation de textes et extraction de connaissances dans les bases de données

En tant qu'applications des techniques d'apprentissage, la catégorisation de textes et l'extraction de connaissances présentent diverses similitudes et une différence majeure.

Nous commençons par cette différence : l'objectif du système. L'objectif d'un système de catégorisation est de minimiser le nombre d'erreurs réalisées par le système. En ce sens, c'est une application classique de l'apprentissage. L'extraction de connaissances a un tout autre but : il s'agit de présenter les connaissances extraites sous une forme compréhensible par des humains [MK95]. La performance du système d'apprentissage n'est alors que secondaire.

Les ressemblances entre ECBD et catégorisation sont les traits qui les opposent à des applications d'apprentissage plus classiques. Les bases de données représentent une source d'exemples d'apprentissage gigantesque. Il est difficile d'envisager la transformation de la totalité d'une base de données en base d'apprentissage. C'est pourquoi une première étape consiste à sélectionner les parties de la base à étudier. Cet aspect peut être mis en correspondance avec les techniques d'échantillonage [LC94, Yan96b] pratiquées parfois pour la catégorisation.

Après avoir choisi les exemples «naturels», il faut les représenter, car l'apprentissage manipule des exemples pour lesquels une représentation a déjà été choisie. Pour la catégorisation, c'est le rôle de l'indexation de fixer une représentation des textes. Les bases de données comportent des éléments structurés, faciles à représenter, et des éléments libres, comme du texte. Transformer une base de données en base d'apprentissage suppose aussi que l'on a choisi les tables sur lesquelles travailler.

En somme, d'un point de vue fonctionnel, catégorisation de textes et ECBD suivent le même schéma : sélection des exemples, sélection des attributs et apprentissage. Nous détaillons à présent la sélection d'attributs, telle qu'elle est réalisée en apprentissage.

4.2 Réduire le langage des instances

La sélection d'attributs évoquée ci-dessus est un cas particulier de la réduction du langage des instances. En effet, étant donnés n attributs, on peut représenter chaque exemple comme un point dans un espace à n dimensions. La sélection d'attributs consiste à projeter les exemples orthogonalement au sous-espace construit à partir du langage réduit. Si l'on ne projette plus orthogonalement, mais selon une direction quelconque, la réduction construit de nouveaux attributs en fonction des anciens. On parle alors de construction d'attributs et non de sélection.

La sélection et la construction ont d'abord été étudiées en analyse des données, avant de faire l'objet de recherches en apprentissage. Nous présentons les travaux de la communauté d'apprentissage, en soulignant l'apport des travaux antérieurs en analyse de données et en reconnaissance des formes.

4.2.1 Comment le langage des instances influence l'apprentissage

Nous avons vu que le langage des instances représentaient une composante importante du présupposé d'apprentissage, au même titre que le langage des hypothèses. Ces langages ont donc une influence non négligeable sur l'apprentissage, en particulier dans les cas suivants.

Le langage des instance ne permet pas de distinguer les différentes instances. [Qui86a] et [Ren88] ont introduit la notion de langage des instances *adéquat*. Dans un tel langage, deux instances ayant la même représentation appartiennent à la même classe. Il semble évident qu'un apprenant aura du mal à apprendre le concept «appartient à la classe X» si le langage des instances n'est pas adéquat. Ceci est en particulier vrai pour les apprenants qui cherchent à apprendre une représentation du concept cohérente avec tous les exemples.

Le langage des hypothèses ne couvre pas la définition du concept. Le langage des instances et le langage des hypothèses sont intrinsèquement liés. Si le concept que l'on souhaite apprendre s'exprime simplement dans un formalisme relationnel, il sera complexe à apprendre si les exemples sont décrits en employant un langage attribut–valeur. En effet, un apprenant propositionnel aura tendance à exprimer le concept relationnel en extension. Le second aspect est propre au langage des hypothèses : il est parfois impossible d'exprimer un concept dans le langage choisi. Par exemple, un classifieur linéaire ne peut pas approcher le concept de sphère.

Les données violent le présupposé sur lesquels l'apprenant s'appuie. Certains apprenants supposent que les données possèdent des propriétés connues. Lorsque les exemples ne vérifient pas ces propriétés, rien n'assure que l'apprenant trouvera une bonne description du concept. Par exemple, employer un moyen d'optimisation tel qu'une descente de gradient peut mener à un optimum local, plutôt qu'à l'optimum global. Une seconde illustration est le cas des classifieurs Bayésiens. Afin de simplifier les calculs, on fait hypothèses d'indépendance entre les attributs. Cette hypothèse est rarement vérifiée lorsque l'application est un cas réel.

Les langages des instances et des hypothèses engendrent un espace des hypothèses trop vaste. L'apprentissage correspond à une recherche dans l'espace de généralisation. La taille de cet espace dépend du nombre d'attributs constituant le langage des instances et du langage des hypothèses. Par exemple, si l'on considère n attributs booléens, le nombre de monomes conjonctifs possible est 3^n, le nombre d'expressions logiques sous forme normale disjonctive avec au plus k littéraux par termes conjonctifs [1] est en $\mathcal{O}(n^k)$ et le nombre d'arbres de décision est 2^{2^n} [Pit94]. L'intuition voudrait que plus on ait d'attributs, donc d'informations sur le concept, meilleur soit l'apprentissage. Cependant, plus le nombre d'attributs augmente, plus l'apprentissage est complexe. En fait ce problème de complexité est au centre d'une grande partie des travaux en théorie de l'apprentissage, en particulier l'apprentissage *pac* (probably approximatively correct) [Val84, Nat91, Bou92] dont l'un des objectifs est de déterminer des bornes minimales sur le nombre d'exemples nécessaires pour apprendre un concept, lorsque le cardinal du langage des instances est donné [Hau88, Paz90, Kad91].

Les données sont bruitées. Dès que les apprenants sont confrontés à des données réelles, le problème du bruit se pose. Il n'en existe pourtant pas de définition précise en apprentissage. Par exemple, Quinlan [Qui86a] traite de ce sujet sans jamais le définir de manière précise. Le bruit inclut aussi bien des valeurs d'attributs manquantes ou erronées que la présence d'attributs redondants. Différents apprenants ont proposé une gestion des valeurs manquantes, par exemple ASSISTANT ou C4.5, et prennent en considération une certaine notion du bruit (cf. CN2, IPEC-DNF ou GROVE). Détecter quels attributs sont redondants [LS94a], c'est-à-dire expriment la même propriété, est délicat et intervient dans la définition du langage des instances. Certains algorithmes sont plus sensibles que d'autres à cette corrélation entre attributs et singularisent plusieurs attributs au lieu d'un unique attribut plus pertinent ; c'est le cas de la méthode des k plus proches voisins, par exemple.

La réduction du langage des instances n'apporte pas de solutions à la totalité des points abordés. Par exemple, transformer un langage non adéquat en langage adéquat ne peut se faire qu'en ajoutant des informations. De même, modifier le langage des instances ne permet pas d'apporter une solution à un langage des hypothèses ne permettant pas d'apprendre le concept. La construction et la sélection d'attributs proposent des éléments de réponse pour limiter la redondance entre attributs ou pour diminuer la taille de l'espace de généralisation.

1. Un système de règles de production peut être converti en une formule logique sous forme normale disjonctive lorsque le concept est binaire.

4.2.2 La réduction des dimensions : l'apport de l'analyse des données

L'analyse des données, encore appelée statistique descriptive ou statistique exploratoire, a pour objectif l'analyse et la présentation de données. Elle s'oppose à la statistique classique dans la mesure où elle n'émet aucune hypothèses sur les données [HL77]. En effet, la statistique classique fait des hypothèses sur des lois de probabilités ou sur des modèles et se sert des exemples (échantillons) pour estimer les paramètres des lois de probabilités ou des modèles. L'analyse de données, quand à elle, n'émet aucun hypothèse et utilise les exemples pour trouver le modèle qui ne dépend que de la structure des données.

[HL77] sépare la visualisation des données de l'analyse à proprement parler. L'analyse discriminante [Cel90], un des sous-domaines de l'analyse des données, a des préoccupations semblables à l'apprentissage automatique. Dans les faits, choisir des variables (attributs) employées pour décrire des exemples est un problème identique au choix du langage des instances en apprentissage [Sap90a]. Les techniques de visualisation et d'analyse permettent d'apporter une réponse à ce problème de représentation lorsque le nombre de variables est important.

La visualisation a pour objectif de représenter les exemples dans un espace de dimension inférieure à la dimension de l'espace d'origine. Si l'on veut obtenir une représentation graphique des exemples, la dimension retenue sera généralement 2 ou 3.

Chaque exemple est représenté par un point dans l'espace engendré par les attributs. L'ensemble des exemples forme un nuage de points et peut être caractérisé par l'inertie du nuage. Plus les exemples sont semblables, plus l'inertie est petite. L'analyse en composantes principales permet de déterminer les axes qui minimisent l'inertie de différents nuages de points. Ces axes principaux peuvent être utilisés pour réduire le langage des attributs. En effet, si l'on affecte un nouvel attribut à chaque axe, cet attribut est une combinaison linéaire des attributs d'origine, qui capture une certaine redondance entre les attributs.

La classification automatique représente un autre moyen de réduire les dimensions d'un problème. Elle permet de regrouper des individus (des exemples en apprentissage) ayant des caractéristiques voisines, en maximisant l'inertie entre les groupes d'individus et en minimisant l'inertie propre à chaque groupe. De nombreuses méthodes ont été proposées pour regrouper les éléments : les nuées dynamiques [Did71], les approches hiérarchiques (voir [Sap90b] pour un survol), l'analyse relationnelle [Mar84]. Pour réduire le langage des instances, on peut considérer les attributs comme les individus à regrouper en fonction des valeurs qu'ils prennent sur les exemples. On associe ensuite à chaque classe un représentant, et l'on substitue ce représentant à tous les éléments de la classe.

Nous avons jusqu'ici présenté des techniques permettant de construire de nouveaux attributs. L'analyse de données s'intéresse aussi à la sélection d'attributs. Différents travaux [Jam85, Kit74] proposent des méthodes pour sélectionner un sous-ensemble d'attributs en utilisant un critère numérique pour estimer la qualité d'un attribut ou d'un ensemble d'attributs. Ces méthodes, reprises en apprentissage, sont en partie l'objet de la section suivante.

4.2.3 La réduction du langage des instances en apprentissage

La sélection d'attributs pertinents est un domaine de recherche en évolution depuis quelques années, qui a donné naissance à des méthodes se distinguant des approches issues de la reconnaissance des formes [Kit86], tout en conservant certaines techniques. Le deuxième aspect de la réduction, la construction d'attributs, correspond aux différents travaux sur l'induction constructive, où il s'agit de modifier le langage des instances en fonction du processus inductif. Le premier objectif de l'induction constructive n'est cependant pas une réduction des dimensions, mais une reformulation plus appropriée des exemples.

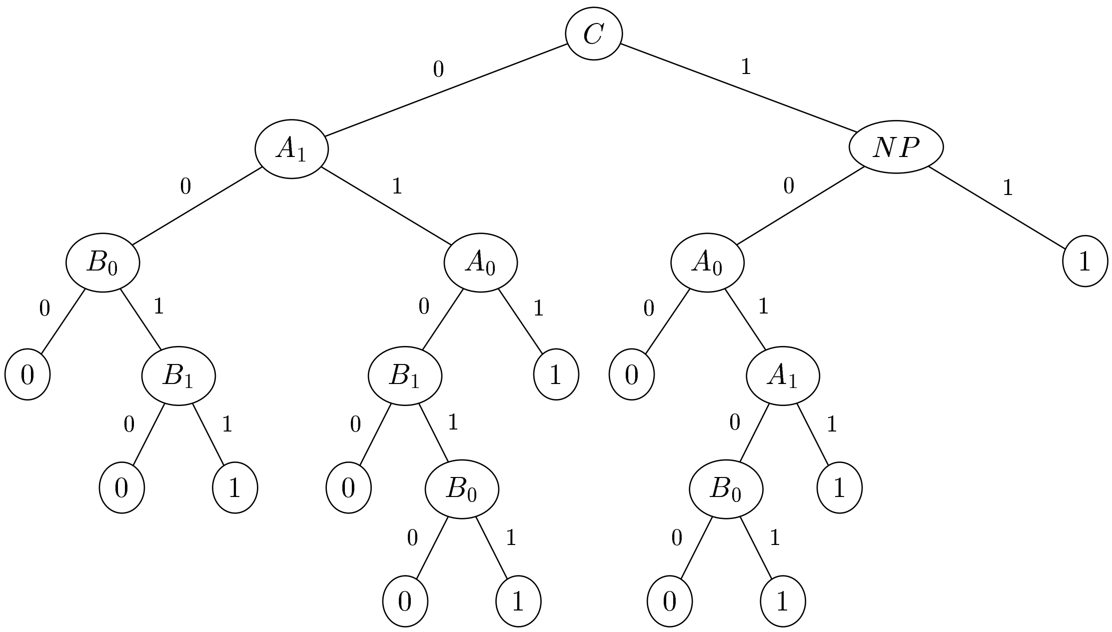
4.2.3.1 Sélection d'attributs

Certains apprenants, tels que les arbres de décision ou les algorithmes s'appuyant sur la méthode des plus proches voisins, réagissent mal en présence d'attributs non pertinents [KS95] : le modèle appris est plus complexe et les performances (par exemple le taux d'erreur) se dégradent.

Reprenons l'exemple introduit par Kohavi [JKP94], où C4.5 ne parvient pas à retrouver l'arbre de décision correspondant au concept recherché en raison de la présence d'attributs non pertinents. On construit une base d'apprentissage artificielle ayant les propriétés suivantes :

- le concept recherché est la fonction booléenne $(A_0 \wedge A_1) \vee (B_0 \wedge B_1)$.
- l'attribut noté NP a une distribution aléatoire uniforme et l'attribut noté C est présent en même temps que la classe 75 fois sur 100.

L'arbre produit par C4.5 est le suivant :



La présence de l'attribut C, corrélé à la classe, et de NP, non pertinent, conduit à un arbre qui est plus complexe que le concept recherché (le sous-arbre gauche).

Le but de la sélection est trouver un *bon* ensemble d'attributs, c'est-à-dire un ensemble d'attributs qui permet d'augmenter les performances de l'apprenant ou bien qui engendre une meilleure représentation du concept appris (par exemple, un arbre de décision plus compact).

On peut distinguer deux tendances pour construire l'ensemble recherché et pour en évaluer la qualité. Dans un premier cas, on sélectionne les attributs ayant une bonne qualité individuelle. L'hypothèse sous-jacente est qu'un ensemble composé d'attributs pertinents sera un bon ensemble. Dans le deuxième cas, sélectionner un ensemble d'attributs est considéré comme un problème de recherche dans l'espace constitué des parties de l'ensemble d'attributs [Lan94].

La sélection, conçue comme une exploration heuristique de l'ensemble des parties de l'ensemble des attributs, doit effectuer les choix suivants :

- le point de départ pour l'exploration,
- une stratégie d'exploration,
- une fonction pour évaluer la qualité d'un ensemble ou d'un attribut,
- un critère d'arrêt.

Où commence la recherche ?

Choisir l'état initial dans l'espace des ensembles d'attributs conditionne le sens de l'exploration. Plusieurs stratégies sont possibles :

- partir de l'ensemble vide et ajouter des attributs un à un. On parle alors de *sélection en avant*. C'est l'option adoptée par [AD91] dans leur système FOCUS et par [LS94a, Car93, Sch93].
- partir de l'ensemble de tous les attributs et en retirer un à chaque pas de l'exploration [LS94b, TWK94]. On parle d'*élimination en arrière*.
- partir d'un ensemble quelconque d'attributs [Ska94, Doa92].

Algorithmes employés pour l'exploration

On peut difficilement envisager un parcours exhaustif de l'espace des sous-ensembles : pour 100 attributs, cela revient à étudier plus de 10^{30} combinaisons possibles. Une approche quasi-exhaustive a cependant été retenue par [AD91, Sch93].

Lorsque la sélection cherche à trouver un sous-ensemble optimal, il est possible d'employer des algorithmes gloutons : à chaque étape, on étudie une modification locale de l'ensemble optimal courant. L'ajout ou le retrait d'un attribut de l'ensemble est irréversible ; il s'agit d'une recherche unidirectionnelle, ascendante ou descendante [CF94, JKP94, AB94]. Une autre approche glouton consiste à sélectionner les attributs avec remise en cause à chaque étape. La recherche n'est plus effectuée dans une seule direction, mais en alternant ajout et retrait d'un attribut.

Dans les approches où l'on évalue la qualité d'un attribut et non la qualité d'un sous-ensemble, on peut considérer que la recherche est exhaustive dans l'espace des attributs (et non pas des parties de l'ensemble des attributs) : en effet, on calcule la fonction d'évaluation pour chaque attribut et on les ordonne en fonction de la valeur obtenue [KR92, Kon94, Doa92].

Évaluation de la qualité d'un ensemble

L'exploration présentée ci-dessus fait nécessairement appel à l'évaluation d'un ensemble d'attributs. [JKP94] différencie deux méthodes d'évaluation : les méthodes de filtrage, qui sélectionnent les attributs de façon indépendante de l'algorithme d'apprentissage, et les méthodes encapsulées, qui font appel à l'algorithme inductif pour chaque état candidat.

Modèle de filtrage

Le modèle de filtrage s'appuie principalement sur les propriétés des données pour sélectionner les attributs. C'est un traitement en amont du processus inductif, qui n'interfère pas avec l'apprenant. Il est employé par les approches évaluant les attributs individuellement [KR92, Kon94, Doa92] ou bien dans des approches construisant des ensembles d'attributs pas à pas [AB95, Car93, AD91].

Modèle encapsulé

Le modèle de filtrage est sujet à critique dans le sens où il ne permet pas de trouver un ensemble optimal[2], pour un apprenant donné [JKP94, KS95]. L'idée derrière le modèle encapsulé, introduit dans [JKP94], est de prendre en considération l'apprenant dans la fonction d'évaluation [CF94, Ska94, LS94b, LS94a].

La manière la plus simple pour prendre en compte l'apprenant dans la fonction d'évaluation est d'utiliser l'apprenant lui-même : pour chaque ensemble d'attributs, on met en œuvre une validation croisée [JKP94, CF94] pour estimer la performance de l'apprenant lorsque les exemples sont décrits à partir de cet ensemble. Le résultat obtenu sert à comparer les sous-ensembles entre eux.

Critères d'arrêt

La fonction d'évaluation ne permet pas d'arrêter la recherche d'un bon ensemble. Ce rôle est assuré par un critère d'arrêt : l'ensemble vérifiant ce critère est proposé comme résultat de la sélection.

Lorsque la fonction d'évaluation suit le modèle encapsulé, le critère d'arrêt est en général lié à l'amélioration de cette fonction. Le parcours de l'espace des parties de l'en-

2. Un ensemble est optimal lorsqu'il permet d'obtenir la meilleure performance avec un apprenant donné. La notion d'optimalité est donc dépendante de l'apprenant et de la base d'apprentissage.

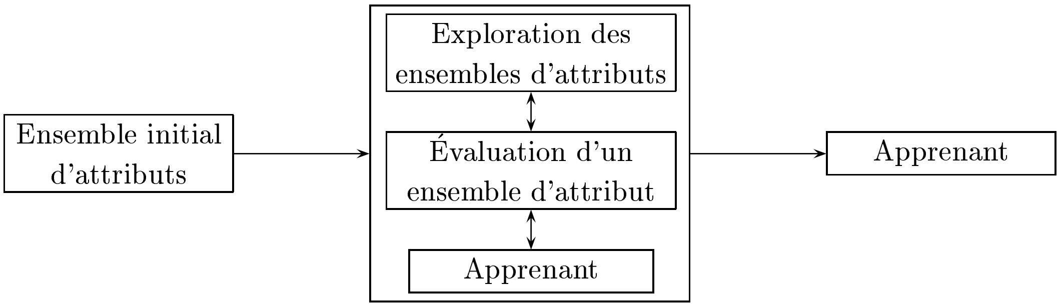


Fig. 4.2 - *Modèle encapsulé*

semble des attributs est interrompu lorsque la fonction d'évaluation se stabilise [JKP94, AB94, LS94b, LS94a] ou n'évolue plus de manière significative.

Parmi les différents critères utilisés dans les approches utilisant le filtrage, nous retenons les deux principaux : la cohérence avec la base d'apprentissage [AD91, Car93, Sch93] et l'utilisation de seuils [Doa92, KR92]. La cohérence assure que le langage des instances sera adéquat : en effet, toute instance ayant la même description a la même classe. Les approches qui reposent sur un ordre linéaire entre les attributs emploient deux types de seuil : un nombre maximal d'attributs à conserver et la valeur minimale de la fonction d'évaluation.

4.2.3.2 Construction d'attributs

La sélection d'attributs ne permet pas de résoudre tous les problèmes posés par un ensemble d'attributs non pertinents. Par exemple, [LS94a] montre que le problème de la redondance n'est pas résolu par la sélection seule. Certains travaux en apprentissage automatique, consacrés à l'induction constructive, se sont intéressés à la modification du langage des instances au cours de l'apprentissage.

L'induction constructive permet de construire de nouveaux attributs en fonction d'attributs initiaux. Différentes sources de connaissances peuvent être employées pour parvenir à ce but. Dans ce contexte, l'apprentissage est considéré comme une double exploration. Il s'agit à la fois de trouver une hypothèse dans l'espace des généralisations et de déterminer quel est cet espace des généralisations, c'est-à-dire quel langage des instances construire.

L'exploration de l'espace de généralisation correspond en fait à la problématique classique de l'apprentissage automatique. La construction d'un nouveau langage des instances s'effectue par application successive d'opérateurs d'expansion, qui ajoutent de nouveaux attributs, et d'opérateurs de contraction, qui éliminent des attributs existants.

Ces opérateurs peuvent être appliqués à un ou plusieurs attributs et produire un ou plusieurs autres attributs : un exemple d'opérateur est la conjonction, appliquée à des attributs booléens pour construire un nouvel attribut. La construction du langage des instances est par ailleurs guidée par différentes connaissances sur l'application de ces opérateurs. [WM94] propose quatre sources de connaissances :

- *les données.* L'ensemble des exemples est étudié dans le but de trouver des relations entre attributs. Cette recherche consiste en l'application successive d'opérateurs. Un exemple de construction de nouveaux attributs est la discrétisation des attributs numériques.
- *les hypothèses.* L'exploration de l'espace des généralisés engendré par le langage des instances initial permet de décrire une approximation du concept. Les hypothèses engendrées sont étudiées et donnent naissance à de nouveaux attributs. Par exemple, le système FRINGE [PH90] construit un arbre de décision à partir d'exemples, utilise les chemins de la racine à une feuille pour créer de nouveaux attributs, combinaisons booléennes des anciens. Ainsi, ces nouveaux attributs explicitent des relations implicites entre attributs.
- *le domaine.* L'application des différents opérateurs est guidée par des connaissances données par un expert.
- *les multiples stratégies.* Les trois approches précédentes peuvent être utilisées simultanément.

Par rapport à la sélection d'attributs, l'induction constructive permet de détecter des descriptions redondantes des exemples, en mettant en évidence des relations entre attributs. Cependant, elle n'apporte pas de solutions lorsque le langage des instances n'est pas adéquat. Le seul recours pour lever une ambiguïté entre deux instances est de fait appel à une aide extérieure. L'induction constructive dirigée par les hypothèses est sujette aux mêmes limitations que l'apprentissage symbolique en grandes dimensions, puisqu'elle fait appel à l'induction.

4.3 Conclusion : réduction pour les application en grandes dimensions

Nous concluons ce chapitre par la comparaison entre la sélection en apprentissage et la réduction des dimensions en recherche documentaire. De plus nous proposons une extension au modèle de filtrage pour l'appliquer aux problèmes de grandes dimensions en intégrant différents points de vue.

4.3.1 Réduction des dimensions en recherche documentaire

Certaines approches proposées pour réduire l'index en catégorisation de textes sont un cas particulier de la sélection d'attributs, puisqu'elles correspondent à un filtrage des attributs en fonction d'un critère numérique évaluant l'intérêt de chaque attribut. Les attributs (unités d'indexation) sont ordonnés en fonction de ce critère et seuls les n meilleurs sont conservés pour décrire les documents. Les critères employés sont relativement simples : il s'agit par exemple de l'information mutuelle [LR94, WPW95] ou même de la fréquence [ADW94a]. Wiener [Wie95] a étudié différents critères mais ne conclut pas sur la supériorité de l'un d'entre eux.

Certaines approches ne se contentent pas de réduire le nombre d'unités par simple projection. Elles cherchent à éliminer la redondance entre les différentes unités. Il s'agit des approches s'appuyant sur l'indexation par sémantique latente utilisées dans [WPW95, SHP95, YW96]. L'indexation fondée sur la sémantique latente est en fait une variation de l'analyse en composantes principales : les nouvelles unités d'indexation correspondent aux axes factoriels et remplacent les unités existantes.

4.3.2 Une proposition pour le filtrage en grandes dimensions

Dans un cadre classique d'apprentissage, l'approche encapsulée de la sélection d'attributs s'est révélée plus efficace que l'approche par filtrage [AB95, JKP94]. Cependant, ce modèle encapsulé ne peut apporter de réelles solutions pour l'apprentissage en grandes dimensions. En effet, nous partons de l'hypothèse que l'apprentissage est trop complexe dans la dimension d'origine. Or, dans le cas de l'élimination en arrière, la sélection commence par évaluer l'ensemble des attributs par validation croisée. Le problème de l'apprentissage en grandes dimensions est répliqué lors de la sélection.

Cependant, l'idée sur laquelle repose l'approche encapsulée nous semble intéressante, à savoir prendre en compte le présupposé de l'apprenant. Dans certains cas, ce présupposé peut être partiellement mis en œuvre sans faire appel à l'apprenant lui-même. C'est pourquoi, nous proposons d'étendre l'approche par filtrage pour intégrer différentes sources de connaissances :

- le présupposé de l'apprenant
- des connaissances extraites des données, telles que la distribution des différents attributs à travers les différentes classes
- des connaissances *a priori* sur le domaine. Dans le cadre de la catégorisation, ce peut être des connaissances linguistiques (synonymes, lemmatisation) ou bien des connaissances sur les étiquettes, sous la forme d'une hiérarchie conceptuelle.

Notre approche pour la réduction de dimension s'appuie donc sur la sélection et non sur la construction d'attributs. Ceci est en particulier motivé par notre intérêt pour l'apprentissage symbolique. En effet, les approches permettant de construire des attributs en recherche documentaire s'appuient sur une représentation numérique des exemples. Ce type de représentation est difficilement manipulable par les apprenants symboliques habituels. Au chapitre 7, nous présenterons trois approches pour la réduction des dimensions prenant en compte différents aspects, telles que des hypothèses sur les données ou le présupposé de l'apprenant.

Chapitre 5

Évaluation

L'évaluation permet d'apprécier la capacité d'un système à effectuer la tâche pour laquelle il a été construit. Elle doit de plus prendre en compte le contexte dans lequel le système est utilisé. Par exemple, en recherche documentaire, l'utilisateur est une composante importante du système. Pourtant, peu de travaux prennent en compte cet aspect [Sar95]. En général, on juge la qualité d'une stratégie de recherche en conduisnat une simulation informatique. On construit des requêtes artificielles et l'on détermine manuellement les documents pertinents dans une collection documentaire donnée. Les performances du système documentaire sont calculées par comparaison aux jugements de pertinence manuels.

Un système de catégorisation de textes est construit grâce à des techniques d'apprentissage, mais est utilisé dans un environnement de recherche documentaire. La mise en œuvre de l'évaluation dans ces deux domaines diffère, même si les éléments importants de l'évaluation sont semblables :

- une tâche à réaliser ; dans le cas de la catégorisation de textes, le système doit affecter des catégories prédéfinies, ce qui ne peut être indépendant de la collection de documents.
- un jugement de pertinence ; dans le cas de l'apprentissage, un tel jugement est donné dans la base d'exemples. En recherche d'informations, il peut être émis séparément.
- une mesure de la performance du système, qui doit prendre en compte l'objectif du système.
- une méthode pour réaliser l'évaluation.
- une méthode pour comparer différents systèmes.

Pour décrire les différences entre apprentissage et recherche documentaire lors de l'évaluation d'un système, nous mettons l'accent sur deux éléments : la mesure de qualité et les méthodes pour réaliser cette évaluation. Nous concluerons ce chapitre sur une brève discussion des techniques employées pour comparer les performances de différents systèmes.

5.1 Évaluation en recherche documentaire

Les systèmes de recherche documentaire ont des objectifs différents selon que l'on effectue une recherche *ad-hoc* ou une diffusion sélective. L'évaluation de ces systèmes tient en général compte de ces différences, tout en s'appuyant sur une simulation informatique à partir d'une collection documentaire donnée a priori. Pour évaluer un système de recherche *ad-hoc*, on en estime la performance en construisant des requêtes artificielles : on essaie de retrouver les documents qui sont pertinents pour chaque requête dans une collection documentaire préalablement étudiée et jugée par des experts. Cependant, lorsque l'on manipule des collections de grande taille (comme celles de TREC), tous les documents (plusieurs centaines de milliers) ne peuvent pas être étudiés par des experts. On considère alors que les exemples non étudiés ne sont pertinents pour aucune requête [1].

L'évaluation d'un système ayant pour tâche la diffusion sélective d'informations est différente dans la mesure où l'on doit accorder un rôle plus important aux documents. En effet, rappelons que la diffusion sélective, qui regroupe routage et filtrage de documents, cherche à répartir des documents selon leur pertinence relative à un nombre prédéfini de profils. En ce sens, c'est une tâche plus proche de la classification que de la recherche ad-hoc, puisque l'on centre l'évaluation autour des documents et non des profils/requêtes.

L'évaluation d'un système de catégorisation s'apparente à celui d'un système de filtrage, et dans une moindre mesure à l'évaluation du processus d'indexation. Nous présentons brièvement la méthode adoptée pour l'évaluation puis les mesures de performance habituelles en recherche documentaire et plus particulièrement pour la catégorisation.

5.1.1 Utilisation d'un ensemble de test

La catégorisation de textes, tout comme le filtrage de documents, est un problème de classification binaire, c'est-à-dire que la décision prise par le système doit être oui ou non. En cela, elle s'oppose à la majorité des tâches de recherche documentaire, dont le routage de documents, où l'on cherche à déterminer le degré de pertinence des documents par rapport à une requête.

Cette différence permet d'envisager une technique d'évaluation proche de celles de la classification. On dispose d'une base documentaire, qui inclut documents et profils. Comme dans le cadre plus général de la recherche documentaire, les documents ont été jugés manuellement et l'on dispose de ces jugements experts. La collection est divisée en deux ensembles. Le premier ensemble est employé pour construire le système, tandis que le second, appelé ensemble de test sert à l'évaluation du système construit.

Pour chaque document de l'ensemble de test, le système affecte une ou plusieurs catégories. Chaque décision prise par le système peut alors être comparée au jugement de pertinence associé préalablement au document. On évalue la performance globale du système à l'aide de différentes mesures, que nous détaillons dans la section suivante.

5.1.2 Mesures d'évaluation

Nous considérons dans un premier temps que l'on ne cherche à affecter qu'une catégorie. Nous proposons ensuite les moyens d'aggréger les résultats obtenus pour n catégories.

1. Cette position introduit du bruit dans la collection documentaire, ce qui peut être important pour la catégorisation de textes.

Attendu

Déduit

	C	$\neg C$
C	a	b
$\neg C$	c	d

TAB. 5.1 - *Résumé des n décisions prises par un système de catégorisation. C correspond au concept «le document possède la catégorie C» tandis que ¬C traduit «le document ne possède pas la catégorie C».*

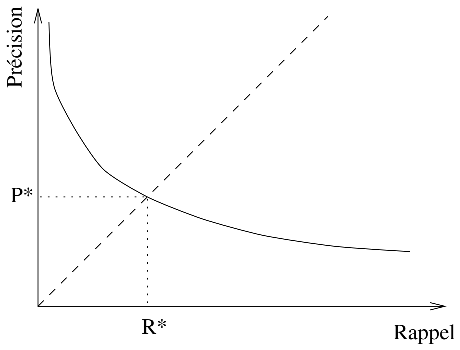


FIG. 5.1 - *Courbe rappel/précision et point moyen*

5.1.2.1 Mesures classiques pour une requête/catégorie

Deux mesures, le *rappel* et la *précision*, sont à la base des évaluations en recherche d'informations et ont été adaptées pour la catégorisation de textes [Lew91]. Le rappel mesure la largeur de la recherche et correspond à la fraction des documents retrouvés par le système. La précision mesure la qualité de la recherche : c'est la fraction de décisions correctes sur toutes les décisions prises.

Si l'on résume les résultats du système sous la forme d'une table de contingence (cf. table 5.1), où sont croisées les décisions du système avec les jugements de pertinence, le rappel R et la précision P s'expriment de la façon suivante et varient entre 0 et 1 :

$$R = \frac{a}{a+c} \quad \text{et} \quad P = \frac{a}{a+b}.$$

Utiliser une mesure à deux variables pour évaluer un système mais surtout pour comparer deux systèmes est délicat. Généralement, on observe que la croissance du rappel entraîne la diminution de la précision. Une manière pratique pour visualiser ces deux mesures est de tracer la courbe avec pour abscisses le rappel et pour ordonnées la précision.

On peut remplacer le rappel et la précision par une unique mesure qui les résume. Une première mesure, courante dans les évaluations en recherche documentaire, par exemple sur les données de TREC, est la précision moyenne . On fixe différents niveaux de rappel (par exemple tous les 10%) et on mesure la précision à ces niveaux. Il est parfois difficile d'obtenir un taux donné de rappel ; dans ce cas, la précision est évaluée par interpolation.

On calcule la précision moyenne comme la moyenne équipondérée de la précision aux différents taux de rappel. Cette mesure est en général employée pour les systèmes qui classent les documents par ordre de similarité avec la requête. Une deuxième mesure, employée en catégorisation, est le point moyen. Il se calcule directement à partir de la courbe : c'est le point où le rappel et la précision ont la même valeur (voir figure 5.1).

Cependant, l'utilisation du point moyen comme mesure a été remis en cause par [Lew95] car il traduit plus une propriété de la courbe que du système. Lewis propose comme alternative l'emploi d'une mesure utilisée en recherche d'informations, la mesure F_β [vR79] :

$$F_\beta(P,R) = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

Cette mesure permet de favoriser l'une ou l'autre variable, en faisant varier β. Elle reflète ainsi les propriétés du système : par exemple, si le système a pour objectif de trouver des informations pointues, on favorisera la précision au détriment du rappel en fixant $\beta < 1$. Ceci n'est possible ni avec la précision moyenne, ni avec le point moyen.

Cependant, changer de mesures d'évaluation soulève le problème de la comparaison avec les différents travaux passés. Cependant, nous avons la propriété suivante :

PROPRIÉTÉ 5.1 *Soit $(R^\star, P^\star)$ les valeurs du rappel et de la précision au point moyen, on a*

$$F_\beta(R^\star, P^\star) = P^\star = R^\star.$$

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

Cette propriété permet de comparer des évaluations réalisées à l'aide du point moyen à celles réalisées à l'aide de F_β. La comparaison est à sens unique. En effet, soit un système A utilisant le point moyen $P^\star$ comme mesure d'évaluation, et un système B utilisant le meilleur F_β, noté $F_\beta^\dagger$; si $F_\beta^\dagger$ est inférieur à $P^\star$, on peut conclure que le système A est meilleur que le système B. Le cas contraire ne nous apporte aucune information pour déterminer quel est le meilleur système.

5.1.2.2 Passage à n catégories

Jusqu'ici, nous avons considéré l'affectation d'une unique catégorie. Quelle attitude prendre lors de l'évaluation d'un système manipulant plusieurs catégories ? La solution adoptée est d'effectuer une moyenne des résultats obtenus pour chaque catégorie, en distinguant deux approches : donner un poids égal à toutes les catégories, on parle alors de *macro-moyenne* ou bien donner un poids proportionnel à la fréquence de chaque catégorie, il s'agit de la *micro-moyenne*.

La micro-moyenne calcule les mesures rappel, précision et F_β de façon globale : si l'on considère les tables de contingences associées à chaque catégorie, cela revient à sommer les cases a, b et c de chaque catégorie pour obtenir la table de contingence globale. Les différentes mesures sont ensuite calculées à partir des valeurs cumulées. Cette micro-moyenne accorde donc un poids important aux catégories ayant beaucoup d'exemples. La performance d'un système dépend en partie de sa capacité à traiter les catégories les plus fréquentes.

L'autre choix, la macro-moyenne, consiste à évaluer chaque requête ou catégorie de manière indépendante. La performance globale du système est calculée en faisant la moyenne des mesures individuelles, que ce soit la précision moyenne ou F_β. Les différentes requêtes ou catégories ont ainsi la même importance. Lorsque l'on étudie des catégories plutôt que des requêtes, on est confronté à un problème spécifique : le nombre de documents pertinents. En effet, il est courant d'avoir des catégories de fréquence très faible. Dans ce cas, il n'est pas rare de ne retrouver aucun texte ayant cette catégorie et ne pas l'affecter à tort, c'est-à-dire que le dénominateur de la précision (ou du rappel) est nul. Quelle valeur donner au quotient indéfini ? On propose généralement 0 ou 1 mais il n'existe aucun critère pour employer une valeur plutôt que l'autre.

La majorité des approches pour la catégorisation utilise soit la précision macro-moyennée [WPW95, Yan96b], soit le point moyen ou F_β [ADW94a, CS96, MRG96] micro-moyennés. Remarquons pour conclure que la propriété 5.1 reste vérifiée pour la micro-moyenne mais ne l'est pas pour la macro-moyenne.

5.2 Évaluation en apprentissage

Nous abordons à présent l'évaluation d'un système d'apprentissage. Comme précédemment, nous présentons la méthode et les mesures de performance.

5.2.1 Méthodes d'évaluation

Kohavi établit une revue des techniques d'évaluation pour l'apprentissage [Koh95]. Nous retenons les deux méthodes les plus fréquemment mises en œuvre: la validation croisée et l'utilisation d'un ensemble de test.

L'utilisation d'un ensemble de test est l'approche la plus simple pour évaluer un système d'apprentissage. Cet ensemble est composé d'exemples qui n'ont pas servi à la construction du système. Pour chaque exemple, on sait de plus s'il appartient ou non au concept recherché. Chaque exemple est donné en entrée du système et l'on compare la sortie avec la valeur attendue du concept («possède la catégorie X»). Ceci permet de mettre à jour l'une des quatre cases de la table de contingence illustrée table 5.1. Cette table sert ensuite à calculer la performance du système.

Un inconvénient principal de cette méthode est la sensibilité de l'évaluation à l'ensemble de test choisi. La validation croisée permet de s'en affranchir. Étant donné un ensemble de n exemples, une validation croisée en m plis divisera ces n exemples en m ensembles égaux. L'apprentissage est réalisé sur $m - 1$ ensembles et le test est effectué sur l'ensemble restant. Cette procédure est répétée m fois. À la fin du traitement, chacun des n exemple a été vu une fois (et une seule) comme exemple de test. L'évaluation de la performance se fait de façon analogue à celle présentée pour un unique ensemble de test, en cumulant les déductions faites par toutes les étapes [Koh95]. L'avantage de cette méthode est de diminuer la sensibilité des résultats face à un ensemble de test fixé. Elle est employée de plus en plus souvent dans l'évaluation de différents apprenants. Son coût restreint cependant son utilisation à des problèmes d'apprentissage de taille limitée : par exemple le projet Statlog [KFS95] ou les travaux décrits par [KS95] utilisent un unique ensemble de test lorsque la base d'apprentissage[2] dépasse 3000 exemples.

5.2.2 Mesures de performance

Quelque soit la méthode d'évaluation choisie, la performance est évaluée à l'aide de la même mesure : la précision. Cette précision ne correspond pas à la même mesure qu'en recherche documentaire[3]. En effet, en recherche documentaire, on s'intéresse principalement aux documents pertinents, tandis qu'en apprentissage, il convient de bien classer tous les exemples. La précision en apprentissage reflète le taux d'erreur réalisé par le système : si l'on se reporte aux informations résumées dans la table de contingence (à la table 5.1), le cœfficient d intervient avec le même poids que le cœfficient a. Plus formellement, Kohavi [Koh95] définit la précision en apprentissage comme la probabilité qu'un exemple choisi aléatoirement soit bien classé par le système. Cette précision P_n est estimée à partir d'un ensemble de n exemples de test :

$$P_n = \frac{a+d}{n}.$$

La précision réelle du système peut être calculée en observant que chaque exemple de l'ensemble de test est un test de Bernouilli : la prédiction est correcte ou elle ne l'est pas. Si S est le nombre de classifications correctes sur l'ensemble des n exemples, pour n assez grand, la distribution de S/n est approximativement normale. La moyenne de cette loi correspond à la précision réelle. La variance de cette loi indique les variations des résultats obtenus. C'est une information importante, qui est de plus en plus prise en compte dans les travaux expérimentaux.

5.3 L'approche que nous avons retenue

L'évaluation doit prendre en compte les objectif du système. Un système de catégorisation se concentre sur l'affectation de catégories et n'utilise l'information «ne possède pas la catégorie» que pour limiter le nombre d'erreurs. La mesure correspondant au taux d'erreur en apprentissage ne reflète pas cet objectif, puisqu'elle prend en compte les exemples négatifs. Prenons un exemple simple. Supposons qu'il y ait 1000 textes pour évaluer un système. Parmi ces 1000 textes, seuls 100 textes possèdent la catégorie recherchée. Considérons de plus que l'on décide par défaut de ne pas affecter la catégorie. Le taux d'erreur maximal estimé, réalisable par le système, est 10% (aucun texte ne s'est vu attribuer la catégorie), or le système n'a permis de catégoriser aucun texte.

2. La base d'apprentissage comprend à la fois les exemples pour entraîner l'apprenant et les exemples de test.

3. Les termes anglais correspondant à ces deux mesures sont différents : *precision* en recherche documentaire et *accuracy* en apprentissage.

Nous avons par conséquent choisi d'employer une mesure parmi celles de la recherche documentaire. Nous avons retenu la mesure F_β, car elle permet d'évaluer les apports respectifs du rappel et de la précision. De manière générale, nous privilégierons la micro-moyenne pour résumer les performances sur plusieurs catégories.

La méthode que nous employons pour évaluer un système de catégorisation fait appel à un ensemble de test. Nous avons rejeté la validation croisée en raison de son coût. Tout d'abord, les applications textuelles sont des applications de grandes dimensions. Or les applications en grandes dimensions de l'apprentissage n'emploient pas la validation croisée. De plus, la catégorisation est un processus plus complet que l'apprentissage, puisqu'il est nécessaire de représenter les documents. Une validation croisée pour la catégorisation requiert donc une représentation des exemples d'apprentissage différente à chaque pli. Finalement, la nature des données ne nous a pas semblé bien se prêter à la validation croisée. En effet, l'une des hypothèses de cette approche est que les classes sont distribuées de manière équitable. Or les bases textuelles n'ont pas une telle uniformité. Prenons l'exemple de la collection Reuters-22 173, certaines catégories sont affectées à moins de 5 documents tandis qu'une des catégories étiquette 30% de la collection.

Dans la suite de nos travaux, nous évaluerons donc différents mécanismes pour la catégorisation en utilisant un ensemble de test.

5.4 Comparaison entre différents systèmes de catégorisation : tests statistiques

Nous concluons ce chapitre par une brève discussion sur les tests statistiques qui permettent de savoir si deux ou plusieurs systèmes ont des résultats significativement différents. L'objectif d'un test statistique est de déterminer si les différences observées ont réellement un sens ou bien si elles sont dues au hasard.

Lorsque l'on cherche à comparer deux méthodes X et Y, on suppose que l'on dispose d'un ensemble de valeurs X_i et Y_i, $i \in [1, n]$ correspondant aux performances de X et de Y sur un échantillon composé de n individus. Pour effectuer la comparaison entre les méthodes X et Y, les valeurs X_i et Y_i ont peu d'importance, c'est pourquoi on s'intéresse principalement à la différence $D_i = Y_i - X_i$. On peut appliquer différents tests, tels que le t-test, le test du signe (*sign test*) (cf. [HW73] ou [Sap90b, Chap. 15]).

Nous nous sommes intéressée plus particulièrement au test du signe. Les hypothèses sont les suivantes :

- les variables D_i sont mutuellement indépendantes
- on peut comparer les X_i et les Y_i

Dans ce cas, on définit s^+ et s^- comme le nombre d'individus pour lesquels $D_i > 0$ et $D_i < 0$. Les cas où $X_i = Y_i$ sont ignorés. Sous l'hypothèse que $p(D_i > 0) = p(D_i < 0) = 1/2$, s^+ suit une distribution binomiale $\mathcal{B}(n, 1/2)$. Le test statistique employé revient à comparer $T = \frac{2s^+ - n}{\sqrt{n}}$ à la loi normale.

Pour appliquer ce test, il nous faut définir les individus sur lesquels mesurer la performance du système. En apprentissage, ces individus sont les exemples. Cependant, le cas de la catégorisation est plus complexe car il peut y avoir plusieurs catégories correctes pour un même exemple. De plus, nous avons opté pour une évaluation de type recherche documentaire. La recherche documentaire, quant à elle, étudie les requêtes [Hul93]. Par transposition, nous considérons les catégories comme individus sur lesquels sont effectuées les comparaisons.

Dans notre application à la collection Reuters-22 173, nous avons été confrontée aux deux problèmes suivants :

- certaines catégories ne comportent que quelques exemples de test. Les résultats obtenus sont donc très variables. La différence entre les approches est-elle cependant significative ?
- pour plus d'un tiers des catégories, les méthodes obtiennent le même résultat (en termes de F_1). Les tests classiques éliminent ces ex-æquo. Vu la forte proportion d'ex-æquo, ne pas les prendre en compte modifie le sens du test : au lieu de chercher si une méthode est meilleure qu'une autre, on cherche si, en cas de différence entre les méthodes, ces différences sont significatives.

À notre connaissance, il n'existe pas de solutions à ces deux problèmes. Les résoudre est hors de portée de notre étude. Nous effectuerons une comparaison entre les méthodes, sans juger si les différences obtenues sont significatives.

5.5 Conclusion

Dans ce chapitre, nous avons présenté les méthodes pour évaluer un système de catégorisation, selon que l'on se place du point de vue de la recherche documentaire ou de l'apprentissage. Nous avons opté pour une évaluation propre à la recherche documentaire, car elle traduit mieux les propriétés du système : ceci n'a d'ailleurs un impact que sur le critère employé car l'évaluation fondée sur un ensemble de test est employée pour les problèmes de dimensions importantes en apprentissage.

La dernière partie de ce chapitre est consacrée aux tests statistiques permettant d'estimer si la différence entre deux approches est significative. Lors de la mise en place de ces tests, nous avons été confrontée à des cas de figure qui mettaient en cause la validité de ces tests. Lors des comparaisons expérimentales rapportées dans les prochains chapitres, nous considérerons qu'en général les différences ne sont pas significatives.

Chapitre 6

Apprenants étudiés pour la catégorisation

Au cours des chapitres précédents, nous avons présenté la catégorisation de textes, et plus généralement l'indexation automatique. Nous avons choisi de considérer la catégorisation de textes comme une application de l'apprentissage automatique. En particulier, nous nous intéressons à l'influence des présupposés d'apprentissage, introduits au chapitre 3, dans l'application à la catégorisation.

Nous décrivons à présent quatre apprenants que nous emploierons dans la partie expérimentale de nos travaux. Le choix de ces apprenants est motivé par leurs présupposés respectifs : CHARADE, un apprenant de règles, ID3, qui construit des arbres de décision, NaiveBayes, un classifieur Bayésien et IB, un algorithme implémentant la méthode des plus proches voisins. Les algorithmes ID3, NaiveBayes et IB sont issus de la bibliothèque de programmes MLC++ [KJL$^+$94].

6.1 CHARADE, un apprenant de règles modulable

Définissons tout d'abord ce qu'est un apprenant de règles [1]. C'est un apprenant qui décrit le concept appris à l'aide d'une formule logique sous forme normale disjonctive, ou comme un système de règles lorsque le concept peut prendre plusieurs valeurs. Ce type d'apprenants s'est développé à partir des travaux de Michalski [ML78].

Nous présentons tout d'abord l'algorithme CHARADE, puis nous le situons parmi les quelques apprenants de règles représentatifs des travaux passés et présents, en accordant un intérêt particulier à l'algorithme AQ. Notre présentation des différents travaux s'articule autour du présupposé d'apprentissage.

6.1.1 CHARADE : théorie et pratique

L'idée sur laquelle s'appuie CHARADE est la détection empirique de régularités [Gan87, Gan92]. De manière intuitive, une régularité correspond à une corrélation entre littéraux observée sur un ensemble d'exemples. Dans la terminologie adoptée par CHARADE, un *descripteur* est un triplet associant un attribut et une valeur à l'aide d'un opérateur de comparaison (par exemple $\mathsf{prime} \geq 1$). Lorsque tous les exemples décrits par la description d_1 le sont aussi par la description d_2, on observe que d_1 implique d_2

1. Les termes anglais consacrés aux apprenants de règles sont *k-DNF learners* ou *rule learners*.

sur l'ensemble d'apprentissage. On utilise l'induction pour généraliser cette régularité à l'ensemble des exemples qui n'ont pas encore été étudiés (au reste du monde). L'apprentissage symbolique peut s'exprimer comme une détection empirique de régularités : la description d_2 correspond à la valeur associée au concept que l'on veut apprendre. Par exemple, à partir des dépêches de la figure 3.1, nous induisons que la présence du mot prime implique que la dépêche possède la catégorie *interest* (taux d'intérêts).

6.1.1.1 Un cadre théorique

La particularité de CHARADE est sa vision mathématique de la détection de régularités. L'approche s'appuie sur deux treillis distributifs. Un premier treillis, noté $\mathcal{D}$, représente l'espace des descriptions et mémorise l'ensemble des descriptions possibles. Le deuxième treillis $\mathcal{E}$ représente l'ensemble des parties de l'ensemble d'apprentissage.

EXEMPLE 6.1 Soit la base d'apprentissage suivante :

$$
\begin{aligned}
&E_1 : d_1 \wedge d_2 \wedge d_3 \wedge d_4\\
&E_2 : d_1 \wedge d_2 \wedge d_4 \wedge d_5\\
&E_3 : d_1 \wedge d_2 \wedge d_4\\
&E_4 : d_1 \wedge d_2 \wedge d_3 \wedge d_4 \wedge d_6
\end{aligned}
$$

Les éléments $\{E_1\}$, $\{E_2, E_4\}$ appartiennent au treillis $\mathcal{E}$ et les descriptions d_1, $d_2 \wedge d_3$ et $d_1 \wedge d_2 \wedge d_4$ appartiennent au treillis des descriptions $\mathcal{D}$.

Définir un treillis nécessite de définir une relation d'ordre partiel entre éléments du treillis. Sur le *treillis des exemples* $\mathcal{E}$, la relation d'ordre est la relation induite par l'inclusion. La relation d'ordre sur le treillis des descriptions $\mathcal{D}$ est la relation de généralité : le plus petit minorant de deux éléments a et b correspond au plus petit généralisé de ces éléments, noté $a \curlywedge b$; le plus petit majorant entre ces deux éléments est noté $a \curlyvee b$ et correspond à la conjonction logique $a \wedge b$. Pour tout couple a et b d'éléments de $\mathcal{D}$, nous avons donc $a \curlywedge b \leq a \leq a \curlyvee b$ et $a \curlywedge b \leq b \leq a \curlyvee b$. Les relations d'ordre sur $\mathcal{E}$ et sur $\mathcal{D}$ sont notées $\leq$.

On introduit deux fonctions entre les treillis $\beta : \mathcal{E} \to \mathcal{D}$ et $\gamma : \mathcal{D} \to \mathcal{E}$, qui sont utilisées pour détecter des régularités sur l'ensemble d'apprentissage. La fonction β associe à chaque élément du treillis $\mathcal{E}$, l'ensemble d'exemples E, le plus petit généralisé des descriptions des éléments de E, c'est-à-dire la description commune à tous les exemples de E.

$$\beta(E) = \curlywedge_{e \in E} \beta(\{e\}), \tag{6.1}$$

où $\beta(\{e\})$ correspond à la description de l'exemple e. Reprenons l'exemple 6.1 ; nous avons $\beta(\{E_1\}) = d_1 \wedge d_2 \wedge d_3 \wedge d_4$ et $\beta(\{E_2, E_4\}) = d_1 \wedge d_2 \wedge d_4$.

L'application γ calcule l'ensemble des exemples couverts par une description, élément de $\mathcal{D}$, à savoir le plus grand élément de $\mathcal{E}$ subsumé par cette description.

$$\gamma(d) = \{e \in \mathcal{E} | \beta(\{e\}) \geq d\} \tag{6.2}$$

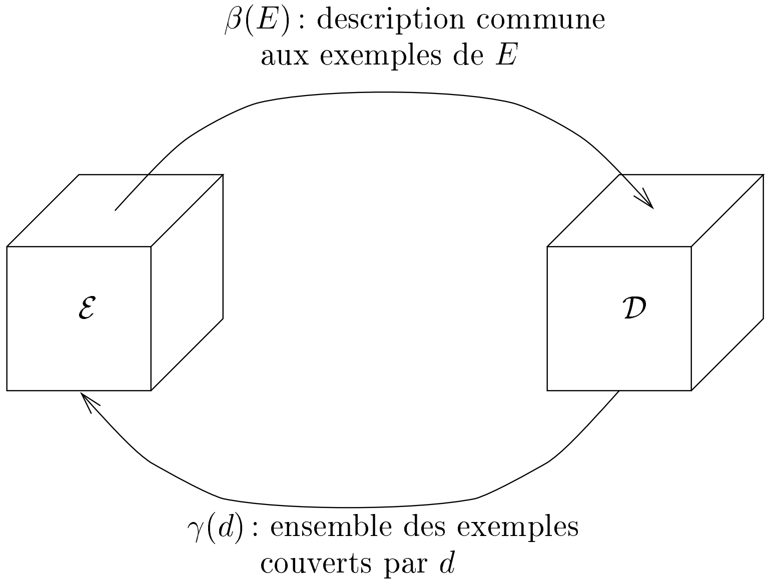


FIG. 6.1 - *Fonctions β et γ entre les treillis*

Pour l'exemple précédent, nous avons $\gamma(d_1 \wedge d_3) = \{E_1, E_4\}$.

Ces deux fonctions forment une connection de Galois entre les treillis $\mathcal{E}$ et $\mathcal{D}$. Les propriétés des treillis et de la connection permettent de détecter des corrélations entre les descriptions des exemples [Gan87, Gan92]. On introduit à cet effet la notion de différence entre éléments du treillis des descripteurs.

DÉFINITION 6.1 *La différence "b moins a", notée b/a, désigne le plus petit élément x tel que $a \curlyvee x \geq b$.*

Dans ce cadre, nous disons qu'une régularité est observée lorsque :

$$\exists d, d' \in \mathcal{D}, \beta \circ \gamma(d)/d = d' \tag{6.3}$$

et, nous pouvons induire la règle $d \rightarrow d'$. L'exemple 6.1 donne lieu à la création des règles $d_3 \rightarrow d_1 \wedge d_2$ ou $d_1 \wedge d_2 \rightarrow d_4$.

6.1.1.2 De la théorie à la pratique

Le formalisme mathématique des treillis permet d'introduire un certain nombre de contraintes sur l'apprentissage. Il permet ainsi de définir partiellement le présupposé d'apprentissage. À l'origine, les travaux de J.-G. Ganascia ont introduit cette formalisation dans un cadre plus général que la classification. Nous ne présentons ici que la facette classification de CHARADE, en distinguant le langage des instances, le langage des hypothèses et le présupposé d'apprentissage.

Le langage des instances que manipule CHARADE est propositionnel ; les attributs numériques et les attributs symboliques sont valides. La représentation des exemples par des ensembles de descripteurs (formellement par des descriptions) est plus souple qu'une représentation vectorielle. En effet, on peut omettre de donner une valeur à certains attributs sans que cela pose problème lors de l'apprentissage ; on peut aussi associer plusieurs valeurs au même attribut, par exemple les descripteurs *étiquette* = *earn* (bénéfices des entreprises) et *étiquette* = *interest* peuvent décrire le même exemple.

Le concept appris est exprimé sous forme d'un système de règles. Chaque règle est constituée d'une description en prémisse et d'une description en conclusion. Les descripteurs intervenant dans la description en prémisse sont des descripteurs apparaissant dans les exemples ou bien une de leurs généralisations. Considérons l'attribut *lieu_de_rédaction* introduit à la section 3.3.3. Une prémisse peut comprendre le descripteur *lieu_de_rédaction* = Minneapolis qui apparaît dans la description d'un exemple, ici la dépêche 502 de la figure 3.1, ou bien le descripteur *lieu_de_rédaction* = USA, qui est une généralisation du descripteur précédent. En général, la description composant la conclusion d'une règle correspond à un descripteur unique. Cependant, rien n'interdit d'avoir une conjonction de descripteurs en conclusion. Cela permet d'apprendre des concepts structurés ou des concepts multi-valués[2].

Pour chaque règle engendrée, la description en prémisses est un élément du treillis des descriptions. Ce treillis $\mathcal{D}$ correspond à l'espace de généralisation exploré par CHARADE. La taille du treillis est importante si l'on le considère en son intégralité. Par exemple, il y a 3^n nœuds dans le treillis lorsque les exemples sont décrits à l'aide de n attributs booléens. Il est donc nécessaire de mettre en œuvre des contraintes pour réduire l'exploration : c'est le rôle du présupposé d'apprentissage.

Le choix effectué dans la mise en œuvre de CHARADE est simple : on parcourt le treillis en une seule passe, du plus général vers le plus spécifique. Ce parcours est une exploration quasi-exhaustive, en largeur d'abord. Ce type d'exploration permet d'assurer que l'on étudie les descriptions les plus générales en premier.

Le parcours du treillis peut être restreint en observant diverses propriétés des fonctions β et γ (voir [Gan87]). Nous donnons ici la propriété qui permet de réduire l'exploration dans le cadre de la classification, compte tenu de l'objectif du système : généraliser les descriptions des exemples.

PROPRIÉTÉ 6.1 *Étant données les descriptions a, b et c, si l'on a induit $a \to c$, spécialiser a est inutile, c'est-à-dire que l'on ne cherche pas de règle du type $(a \wedge b) \to c$.*

Pour que cette propriété soit applicable, nous devons choisir quand produire une règle à partir d'un nœud. La condition respectée dans CHARADE est la cohérence avec les exemples : ainsi, une règle $d \to c$ n'est engendrée que lorsque tous les exemples couverts par la description d appartiennent à la même classe.

En fait, le rôle des exemples est double : ils permettent naturellement de réduire l'exploration, mais peuvent entraîner une exploration plus large lorsque l'on veut un

2. Un concept est multi-valué s'il peut prendre plusieurs valeurs pour le même exemple. C'est le cas des catégories affectées aux dépêches.

apprentissage redondant. Les exemples contraignent l'exploration de l'espace de généralisation, dans la mesure où il est inutile de spécialiser une description d qui ne couvre pas d'exemples ; en effet, toute spécialisation de d ne couvrira pas non plus d'exemples.

Il peut être parfois intéressant d'avoir un apprentissage redondant [GDP91] pour réduire les erreurs de classification liées à des données bruitées. L'apprentissage redondant revient à généraliser un même exemple à l'aide de plusieurs règles. Cette notion de redondance est incluse dans CHARADE. Elle a un contrecoup sur l'exploration de l'espace de généralisation. On introduit le paramètre λ pour contrôle la *redondance* du système. S'il est supérieur à 1, cela signifie que CHARADE recherchera à engendrer des règles de manière à ce que chaque exemple soit couvert au moins λ fois.

Nous nous intéressons pour finir aux deux autres paramètres que nous avons employés lors des expériences décrites dans les chapitres 8 et 9. Le *fractionnement*, noté f, permet de restreindre le parcours du treillis. Il définit le pourcentage[3] d'exemples qu'une règle doit couvrir pour être engendrée. Ceci a une influence sur le parcours dans la mesure où les descriptions ne couvrant pas le nombre minimal d'exemples requis ne sont pas spécialisées. L'utilisation du fractionnement peut être considérée comme un élagage a priori du système de règles appris, à l'aide d'un critère simple, la couverture des règles.

Le dernier paramètre est la profondeur maximale du treillis, notée k : CHARADE engendre des règles ayant au maximum k littéraux en prémisses. En limitant ainsi la profondeur de l'exploration, [Gan87] a montré que CHARADE était polynômial en fonction du nombre d'attributs et linéaires en fonction du nombre d'exemples. L'algorithme de CHARADE, écrit en pseudo-code, est donné à l'annexe B.

6.1.2 CHARADE et les apprenants de règles

CHARADE possède certaines propriétés qui nous ont semblé intéressantes pour une application à la catégorisation de textes. Nous présentons brièvement quelques apprenants de règles par comparaison avec CHARADE avant d'énoncer ces propriétés.

Généralement, on distingue les apprenants de règles qui s'appuient sur la logique des propositions des apprenants qui utilisent la logique des prédicats. Ces derniers sont des systèmes tels que FOIL [Qui90], FOCL [PK92] et GOLEM [MF90] qui illustrent les travaux sur la *programmation logique inductive*. Nous nous intéressons maintenant aux apprenants propositionnels, dont CHARADE fait partie.

6.1.2.1 Langage des instances et langage des hypothèses

Le langage des instances des différents apprenants regroupe à la fois le type des attributs autorisés et la façon dont sont représentés les exemples. En général, les différents systèmes représentent chaque exemple sous la forme d'un vecteur. Les attribut doivent alors prendre une unique valeur pour un même exemple : on ne peut pas omettre de donner la valeur d'un attribut, à moins de définir une valeur spéciale ; on ne peut pas non plus donner plusieurs valeurs au même attribut. En comparaison, la représentation

3. Ce pourcentage est relatif à chaque classe et non calculé globalement sur l'ensemble d'apprentissage.

$$\begin{aligned}
\mathsf{wheat} = v \;\&\; \mathsf{farm} = v &\rightarrow \mathit{wheat} = v \\
\mathsf{wheat} = v \;\&\; \mathsf{commodity} = v &\rightarrow \mathit{wheat} = v \\
\mathsf{bushels} = v \;\&\; \mathsf{export} = v &\rightarrow \mathit{wheat} = v \\
\mathsf{wheat} = v \;\&\; \mathsf{agriculture} = v &\rightarrow \mathit{wheat} = v \\
\mathsf{wheat} = v \;\&\; \mathsf{tonnes} = v &\rightarrow \mathit{wheat} = v \\
\mathsf{wheat} = v \;\&\; \mathsf{winter} = v \;\&\; \mathsf{soft} = f &\rightarrow \mathit{wheat} = v
\end{aligned}$$

TAB. 6.1 - *Un système de règles décrivant le concept «possède la catégorie wheat», d'après [ADW94a]*

ensembliste adoptée par CHARADE est plus souple. Par ailleurs, le type des attributs autorisés varie selon les apprenants. Par exemple, FRINGE et GREEDY3 [PH90] manipulent uniquement des attributs booléens. La majorité des apprenants de règles accepte des attributs ordonnés (la taille d'un ballon est petit, moyenne ou grande), énumérés (la couleur du ballon est jaune, rouge ou verte) ou numériques[4]. Les algorithmes AQ, CN2 ou CHARADE utilisent en plus des attributs hiérarchiques, comme l'attribut *lieu_de_rédaction* introduit précédemment. Le système RIPPER [Coh96b] introduit la notion d'attributs ensemblistes (*set-valued attributes*). Cette notion est également présente dans CHARADE et peut être rapprochée de la possibilité de définir plusieurs valeurs pour un même attribut dans CHARADE.

Le langage des hypothèses contraint la forme des généralisations possibles. Ici, ces généralisations sont exprimées à l'aide d'un système de règles ou d'une formule logique sous forme normale disjonctive[5]. Par exemple, le système Swap-1, utilisé dans [ADW94a], engendre le système de règles donné à la table 6.1 pour affecter la catégorie *wheat* (blé). Certains apprenants contraignent de plus la forme des systèmes de règles. C'est le cas des listes de décision [Riv87], employées également dans les apprenants présentés par [PH90] et dans CN2. Au lieu d'avoir un système où chaque règle a une importance égale, une liste de décisions correspond à une cascade de règles : si la première condition est vérifiée par l'exemple, alors on s'arrête, sinon, on essaie la condition suivante.

Le composant élémentaire d'une règle est appelé littéral (ce qui correspond à un descripteur pour CHARADE). C'est un triplet ⟨attribut opérateur valeur⟩. Ce triplet est souvent une généralisation des couples attribut-valeur apparaissant dans la description des exemples, où l'opérateur est un opérateur de comparaison. Le système RIPPER introduit un opérateur supplémentaire $\in$ pour manipuler les attributs ensemblistes[6]. En général, la valeur affectée est une valeur unique ; dans le système AQ, la valeur peut contenir une disjonction, dite interne, de valeurs élémentaires, par exemple *couleur* $=$ *bleu* $\vee$ *rouge*. Ces littéraux sont engendrés au cours de l'exploration de l'espace de généralisation.

4. Les attributs numériques sont discrétisés de manière à être traités comme des attributs ordonnés. Ce traitement des attributs numériques est souvent coûteux, car il engendre de nombreux descripteurs pour chaque attribut.

5. Il n'y a équivalence que lorsque le concept est traduit par une classe binaire et que le système de règles ne conclut que sur une des valeurs de la classe. Dans un cas plus général, une formule correspond au groupe de règles possédant la même conclusion.

6. RIPPER introduit en fait un littéral du type $x \in A$, où x est une valeur et A l'attribut.

6.1.2.2 Exploration de l'espace des hypothèses

Une caractéristique commune à la majorité des apprenants de règles est la mise en œuvre d'un algorithme de couverture. L'idée de cet algorithme est simple : on construit une règle qui couvre un certain nombre d'exemples ; ces exemples sont marqués comme déjà couverts et l'on réitère la construction d'une règle jusqu'à ce que tous les exemples soient couverts. Cette technique est adoptée par AQ, CN2 et Charade. Un cas particulier de l'algorithme de couverture est la stratégie *séparer pour régner* : au lieu de marquer les exemples, on les retire de l'ensemble d'apprentissage, et l'on s'arrête lorsque ce dernier est vide. C'est une méthode également mise en œuvre dans RIPPER et pour les listes de décisions [Riv87, PH90].

Construire une règle revient à adopter une stratégie pour choisir les littéraux et pour décider d'arrêter la construction. On distingue deux approches pour choisir les littéraux : une approche glouton par descente de gradient et une approche en faisceau. La première méthode est proche des choix adoptés lors de la construction d'arbres de décision : on construit une conjonction en partant de la conjonction vide, en ajoutant pas à pas le meilleur littéral selon un critère que l'on optimise. Ce critère correspond par exemple à l'information mutuelle ou à l'entropie [Coh95b, PH90]. L'approche en faisceau est à l'origine adoptée par Michalski sous le nom de l'algorithme *étoile*. À la différence de l'approche précédente, on conserve m descriptions prometteuses à chaque étape, m étant la taille de l'étoile. Les littéraux employés pour spécialiser les descriptions de l'étoile sont ceux apparaissant dans l'exemple que l'on cherche à couvrir pour AQ [7], et tous les descripteurs possibles pour CN2. Ces systèmes engendrent donc des règles par explorations successives de l'espace de généralisation. La stratégie d'exploration adoptée par Charade peut être vue comme une unique descente dans l'espace de généralisation avec une étoile de taille infinie.

On distingue également deux stratégies pour arrêter la construction d'une règle : la cohérence et un taux d'erreur minimal. Un système comme AQ arrête la construction de la règle en cours dès qu'une règle ne couvre que des exemples appartenant à la même classe : on dit alors que la règle est cohérente ; c'est aussi la stratégie adoptée par Charade. Lorsque les données sont bruitées, la cohérence est une condition forte, qui peut entraîner un sur-apprentissage. En réponse, le système CN2 arrête la construction de la règle en s'appuyant sur un critère numérique.

D'autres techniques permettent d'éviter le sur-apprentissage : il s'agit de l'élagage des règles. RIPPER et les systèmes décrits par [PH90] proposent d'élager chaque règle après sa construction, puis d'élaguer la base de règles globalement. C4.5Rules s'appuie sur un élagage après construction. Ces techniques d'élagage sont semblables aux techniques appliquées aux arbres de décision et visent à diminuer la complexité de la règle de classification apprise sans augmenter le taux d'erreur.

7. Une version simplifiée de l'algorithme AQ est donnée à l'annexe B.

6.1.2.3 CHARADE : avantages et inconvénients

Pourquoi avoir choisi CHARADE pour la catégorisation de textes ? Ce choix peut paraître contre-intuitif. En effet, c'est un algorithme a priori coûteux puisqu'il parcourt l'espace de généralisation de manière quasi-exhaustive. De plus Il engendre des règles cohérentes. Nous avons mentionné la sensibilité des apprenants de règles cohérentes au sur-apprentissage, en particulier lorsque les données sont réelles.

Deux caractéristiques de CHARADE ont influencé notre choix : la redondance et la souplesse du langage des instances et du langage des hypothèses. Dans notre esprit, la redondance, comme elle est perçue pour l'apprentissage, pouvait être utile dans une application à des données textuelles. Les textes sont structurés en paragraphe, chaque paragraphe convoyant une idée. Un apprentissage redondant permettrait de trouver les catégories relatives à chaque paragraphe. Il apparaît clairement que cette intuition ne peut être vérifiée de manière directe, en raison de la complexité inhérente aux applications textuelles. Nous évaluerons l'intérêt de la redondance pour la formulation usuelle de la catégorisation de textes au chapitre 8.

C'est surtout la souplesse liée à l'emploi d'un treillis des descriptions qui a motivé notre choix. En effet, elle nous permet d'explorer l'impact du langage des instances et du langage des hypothèses. On peut par exemple favoriser certains descripteurs par rapport à d'autres en omettant de les introduire dans la description des exemples. C'est ainsi que nous pouvons décrire les textes uniquement en fonction des unités qui les indexent. Comme CHARADE engendre des hypothèses à partir de la description des exemples apparaissant dans la base d'apprentissage, les règles induites seront aussi contraintes à n'exprimer que la présence des unités. Le deuxième aspect est celui que l'on retrouve dans RIPPER : la notion d'attributs multi-valués. CHARADE permet de considérer des attributs multi-valués, voire plusieurs attributs, pour représenter un concept alors que RIPPER n'a pas une telle propriété. Par exemple, utiliser CHARADE rendrait possible l'affectation simultanée de deux catégories sans opter pour une binarisation du problème.

6.2 Construction d'arbres de décision

La construction d'arbres de décision n'est pas, à proprement parler, une technique récente d'intelligence artificielle, puisque des travaux d'analyse de données et de théorie des questionnaires [Pic72] ont étudié la construction d'arbres de décision dans les années 60. Toutefois, l'aspect symbolique des résultats engendrés par cette approche est indéniable : un arbre de décision représente une connaissance structurée, intelligible lorsque la taille de l'arbre est raisonnable. Deux systèmes ont plus particulièrement marqué les travaux sur la construction d'arbres de décision : ID3 [Qui83, Qui86b] dans la communauté de l'I.A. et CART [BFOS84], dont l'origine est statistique.

Nous avons succintement présenté le présupposé relatif à la construction d'arbres à la section 3.3.2. Nous reprenons et détaillons cet aspect. Les exemples sont représentés sous forme vectorielle : une description élémentaire est une conjonction de couples attribut-valeur. Comme pour les apprenants de règles, nous singularisons l'attribut indiquant si un exemple appartient ou non au concept cherché. Cet attribut est appelé la classe.

Les nœuds de l'arbre produit représentent des tests (par exemple, «le mot prime est-il présent ?» ou «quelle est la fréquence du mot prime ?») tandis que les branches de l'arbre sont étiquetées par les différents résultats de ces tests. Les feuilles sont des nœuds où un ensemble d'exemples vérifie un critère d'arrêt, par exemple, tous les exemples de l'ensemble possèdent la même valeur pour la classe. L'étiquette associée à chaque feuille dépend de l'apprenant. L'arbre présenté figure 6.2 a été obtenu avec le système IND [Bun90] ; il est tronqué à la profondeur 3. Les tests sont du type «le mot X est-il présent ?» et le résultat de ces tests prent la valeur t (*vrai*) ou f (*faux*).

La procédure générale est ensuite inchangée : c'est une approche récursive, s'appuyant sur une stratégie *diviser pour régner* et la méthode de descente de gradient, ce qui suppose que les choix effectués à un moment ne sont pas remis en cause ultérieurement.

Construction d'arbres de décision

1. *Si tous les exemples vérifient le critère d'arrêt, l'arbre est réduit à une feuille, étiquetée par les valeurs possibles pour la classe.*
2. *Sinon*
   (a) *sélectionner un test,*
   (b) *diviser l'ensemble d'apprentissage en sous-ensembles où chaque ensemble correspond à un résultat possible du test. Les différents résultats sont mutuellement exclusifs.*
   (c) *appliquer la même procédure à chaque sous-ensemble (retour en 1)*

Les différents apprenants se distinguent par les solutions adoptées tant pour choisir un test que pour arrêter la construction. Il faut dans un premier temps déterminer ce que l'on considère dans le test, puis comment choisir le meilleur test. On distingue deux cas : le test porte sur un unique attribut, comme dans ID3, ou bien le test porte sur plusieurs attributs, comme dans CART. Le choix de l'attribut qui figure dans le test se fait après le calcul exhaustif d'un critère de préférence. Différents travaux se sont intéressés à ce critère. Quinlan [Qui83] propose d'employer le gain d'information (ou information mutuelle), puis le ratio de gain [Qui86b] pour pallier certains défauts du gain d'information. Fayyad et Irani [FI90, FI92] proposent un autre critère tandis que Mingers évalue les différentes mesures proposées dans la littérature [Min89b]. Dans la majorité des approches s'appuyant sur ID3, les attributs ne peuvent apparaître qu'une fois dans une même branche de l'arbre construit. Les tests portant sur plusieurs attributs sélectionnent principalement des combinaisons de ces attributs. Différentes approches sont proposées pour trouver ces combinaisons. Le système CART [BFOS84] effectue une recherche déterministe à base d'heuristiques ; le système SADT [HS92] utilise une recherche non déterministe s'appuyant sur le recuit simulé. Dans [MKS94], le système OC1 combine les deux approches précédentes : une recherche heuristique permet de trouver un minimum local, tandis qu'une stratégie aléatoire permet d'en sortir. Un critère de préférence est toujours nécessaire pour évaluer quel est le meilleur test. Les critères employés sont

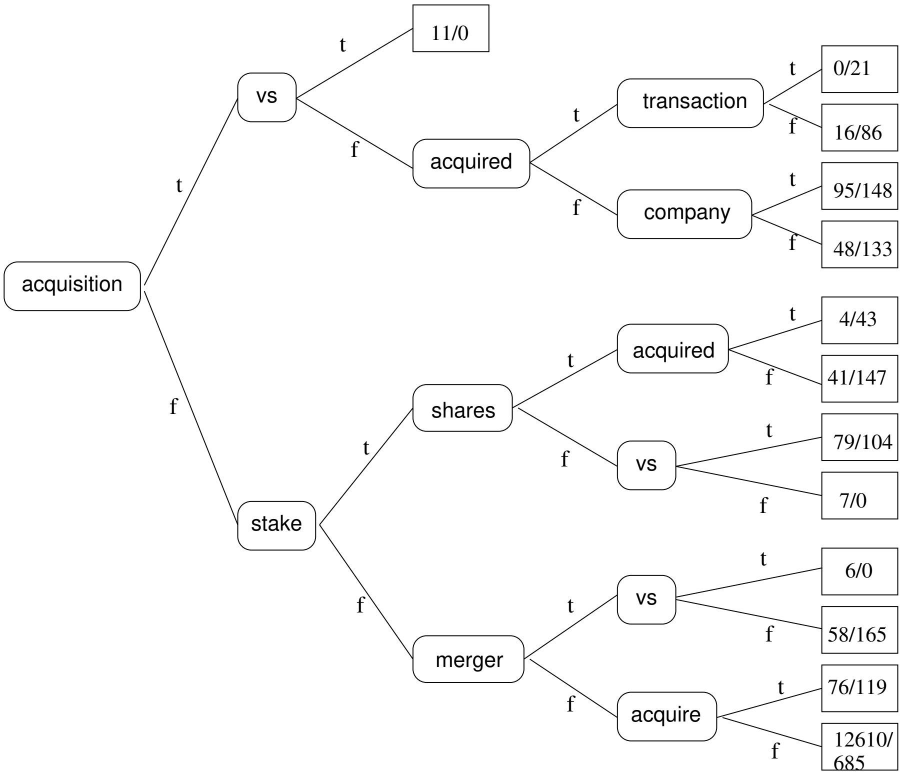


Fig. 6.2 - *Un arbre de décision décrivant le concept «possède la catégorie* acq*», d'après [LR94]. Les feuilles sont étiquetées par le nombre d'exemples négatifs/positifs sur l'ensemble d'apprentissage.*

similaires à ceux décrits par [Min89b]. [MKSB93] propose différents critères, mais aucun n'est significativement meilleur.

Le critère d'arrêt dépend souvent des propriétés du système. ID3 cherche une description cohérente avec les exemples : la construction ne s'arrête que si tous les exemples apparaissant à un nœud possèdent la même valeur pour la classe. Une telle condition n'est pas forcément vérifiée, si bien que la construction peut s'arrêter en raison de l'absence de nouveaux attributs à prendre en compte dans le test. Comme l'apprentissage de règles cohérentes, cette approche est sensible au sur-apprentissage. Différents mécanismes ont été mis en œuvre pour pallier ce problème. Il s'agit d'élaguer l'arbre pendant ou après la construction. Le système IND [Bun90] ne partitionne un nœud que s'il contient un nombre minimal d'exemples. On peut aussi choisir d'arrêter la construction lorsqu'une condition n'est pas vérifiée. Cette condition peut être l'absence d'amélioration du critère de préférence employé pour le partitionnement ou bien le taux de reconnaissance sur les exemples d'apprentissage (par exemple dans Assistant [CKB87]).

Le taux de reconnaissance (ou son dual le taux d'erreur) est à la base des techniques d'élagage a posteriori des arbres. [Min89a] propose une revue des différentes techniques. Ce mécanisme d'élagage est incorporé dans la majorité des systèmes [Qui92, PH90, MKS94, BFOS84] et repose sur le principe suivant. L'élagage est réalisé sur une base de validation, habituellement différente de la base d'apprentissage. Chaque sous-arbre est évalué ; si la classification d'un sous-arbre a n'est pas significativement différente de celle d'un sous-arbre réduit à la racine de a, le sous-arbre a est remplacé par sa racine.

Les méthodes construisant des arbres de décision sont en général confrontées à des problèmes similaires aux apprenants de règles, lorsqu'elles manipulent des attributs numériques. De plus, à l'exception de EG2 [Nuñ91], les approches construisant des arbres de décision ne permettent pas de manipuler des attributs hiérarchiques.

6.3 Apprenants numériques

Il existe de nombreuses techniques inductives pour la classification et pour la prédiction dans le cadre de l'analyse de données et des statistiques. Nous décrivons succinctement deux approches, que nous avons utilisé lors de nos expériences : un classifieur Bayésien et la méthode des plus proches voisins.

On peut considérer que le langage des instances des apprenants numérique est propositionnel, puisqu'il s'agit d'un formalisme vectoriel. Les attributs sont en général de type numérique (quantitatif), ou bien symbolique (qualitatif). Comme pour les méthodes construisant des arbres de décision, la prise en compte de connaissances du domaine ne peut se faire au moyen d'attributs hiérarchiques. Elles doivent être incluses dans la représentation choisie pour les exemples. On ne peut pas toujours parler de langage des hypothèses, en particulier pour la méthode des plus proches voisins. On peut cependant considérer comme langage des hypothèses la forme de la fonction recherchée, comme c'est le cas pour l'apprentissage Bayésien ou les classifieurs linéaires.

6.3.1 Classifieurs Bayésiens

Nous avons choisi un classifieur Bayésien [DH73, LIT92] pour établir un point de référence lors de nos expériences. Dans [LR94], Lewis emploie un classifieur similaire, et trouve que les arbres de décision obtiennent des performances légèrement meilleures.

Lors de la phase d'apprentissage, un classifieur Bayésien mémorise une matrice de probabilités conditionnelles des valeurs associées à chaque attribut A connaissant la valeur associée à la classe C : $p(A = a_i|C = c_j)$. Ces probabilités sont en général estimées en fonction du nombre de fois où l'attribut A et la classe C co-occurrent sur l'ensemble d'apprentissage. Il s'agit en quelque sorte d'une mémorisation des informations contenues dans les exemples.

Pour classer un nouvel exemple, on applique la règle de Bayes [Jam85] :

$$p(C = c_j|\mathbf{x}) = \frac{p(\mathbf{x}|C = c_j)p(C = c_j)}{\sum_n p(\mathbf{x}|C = c_n)p(C = c_n)},$$

où $\mathbf{x} = (A_1 = a_1, \ldots, A_n = a_n)$ représente la description (conjonctive) du document x.

$$\begin{aligned}\log p(C = \mathsf{acq}|D) = \quad & -2.187+ \\ & \log(5.879 \times \mathsf{acquire} + 0.840)+ \\ & \log(5.398 \times \mathsf{acquisition} + 0.795)+ \\ & \log(2.609 \times \mathsf{shares} + 0.715)+ \\ & \log(5.414 \times \mathsf{stake} + 0.826)\end{aligned}$$

Fig. 6.3 - *Un exemple de classifieur Bayésien pour la catégorie* acq *(fusions/achats d'entreprises), issu de [LR94]. Le classifieur est construit à partir de quatre attributs. On utilise le* log *par souci de clarté.*

Pour calculer les probabilités $p(\mathbf{x}|C = c_j)$, on suppose que les attributs A_j sont indépendants. Cette règle se réécrit alors :

$$p(C = c_j|\mathbf{x}) = \frac{\prod_i p(A_i = a_i|C = c_j)p(C = c_j)}{\sum_n \prod_i p(A_i = a_i|C = c_n)p(C = c_n)} \tag{6.4}$$

On calcule alors la probabilité pour chaque classe et l'on affecte au document x la classe ayant la plus forte probabilité. Si l'on considère la formule 6.4, on s'aperçoit qu'une probabilité $p(A_i = a_i|C = c_j)$ nulle entraîne que $p(C = c_j|\mathbf{x}) = 0$. Pour contourner ce problème, la version de NaiveBayes que nous avons utilisée substitue la valeur $1/2m$ à ces probabilités, m étant le nombre d'exemples dans la base d'apprentissage.

Dans une application à la catégorisation de textes, on peut représenter les textes comme des vecteurs booléens. Chaque attribut et chaque classe n'a que deux valeurs possibles. On affecte donc la catégorie C à un document $\mathbf{x}$ si $p(C = 1|\mathbf{x})$ est supérieur à un seuil p. Un exemple de classifieur Bayésien est donné à la figure 6.3.

6.3.2 Un apprenant s'appuyant sur la méthode des plus proches voisins

La méthode des plus proches voisins connaît depuis quelques années un essor important en apprentissage, sous le terme apprentissage fondé sur les instances (*instance-based learning*) [Das90, AKA91]. Cette méthode est souvent qualifiée de paresseuse, dans la mesure où la partie calculatoire est différée à la phase de test, au lieu de prendre place lors de l'apprentissage comme dans les approches précédentes.

L'apprentissage se restreint à une mémorisation des exemples. Comme cette mémorisation est coûteuse en espace mémoire, certains travaux ont cherché à réduire le nombre d'exemples à mémoriser [KA88]. Cette réduction s'effectue

- par filtrage, lorsque l'on choisit de ne mémoriser que les instances les plus représentatives,
- ou par agglomération, lorsque l'on mémorise la moyenne de quelques instances, tout en conservant d'autres instances originales.

Pour classer un exemple inconnu, il faut fixer quelle mesure employer pour évaluer la distance entre deux exemples. Une telle mesure correspond par exemple à la distance

Euclidienne. Pour chaque exemple, on calcule les n exemples mémorisés les plus proches ; on lui affecte ensuite la classe majoritaire parmi les n exemples. Le choix du nombre de voisins n reste empirique : il peut être fixé, par tâtonnements, en fonction du nombre de variables pour décrire un exemple et du nombre d'exemples d'apprentissage.

6.4 Conclusion

Nous avons choisi quatre apprenants pour les appliquer à la catégorisation de textes. Ces apprenants sont des outils dont nous cherchons à évaluer les caractéristiques pour la catégorisation de textes.

Ces caractéristiques sont en particulier liées au présupposé d'apprentissage. C'est pourquoi, nous opposons un algorithme mettant en œuvre une exploration par couverture, CHARADE, à un algorithme s'appuyant sur une stratégie *diviser pour régner*, ID3. Les exemples seront représentés en utilisant le même formalisme : une représentation vectorielle booléenne. ID3 est une version simple des arbres de décision, que la majorité des travaux en apprentissage n'emploient plus. Nous avons préféré ID3 à C4.5 en raison des parallèles que l'on peut établir avec CHARADE : cohérence des hypothèses, absence d'élagage, difficulté à manipuler les attributs numériques. De plus, le nombre de paramètres à régler est moindre, ce qui est un avantage lorsque les dimensions des données sont importantes.

Avant d'évaluer les caractéristiques de ces différents apprenants, nous revenons sur la réduction du langage des instances et proposons une méthode dédiée à la catégorisation de textes.

Chapitre 7

Réduire l'index en rendant implicite l'absence

Le chapitre 3 a montré le rôle du présupposé lors de l'apprentissage. Dans le chapitre 4, dans le cadre d'un apprentissage symbolique, nous avons étudié les problèmes propres à la catégorisation de textes, en particulier ceux posés par la taille du vocabulaire mis en jeu lors d'une telle application.

Les travaux que nous présentons ici sont articulés autour de l'apprentissage de règles, tel qu'il est réalisé par CHARADE. Nous avons exploité la flexibilité de cet apprenant pour évaluer un présupposé particulier lors de la construction inductive de systèmes de catégorisation.

Nous proposons dans un premier temps SCAR, un algorithme de réduction du langage des instances qui s'appuie sur différents présupposés. Cet algorithme est comparé de manière empirique à d'autres approches de réduction du langage des instances par filtrage numérique.

7.1 Les présences seules : un présupposé d'apprentissage

Que ce soit en recherche documentaire ou en apprentissage, les exemples naturels, ici les textes, doivent être décrits pour la machine. L'approche généralement adoptée repose sur une représentation vectorielle des textes : étant donné l'ensemble des unités d'indexation, un texte est représenté par un vecteur, où chaque élément de ce vecteur indique le poids de l'unité correspondante dans le texte. Lorsque nous choisissons une représentation booléenne des textes, le poids associé à une unité présente est 1, tandis qu'une unité absente d'un texte a un poids nul.

Du point de vue de l'apprentissage, cette représentation vectorielle correspond à un langage des instances booléen. Qu'entendons-nous alors par *absence par défaut* ? Nous proposons une alternative à la représentation vectorielle en apprentissage : une représentation ensembliste. Le langage des instances reste inchangé, c'est toujours l'ensemble des unités composant l'index de la collection de textes. Par contre, la description individuelle des exemples se trouve modifiée, puisqu'un texte n'est décrit que par les unités qui le composent. La dépêche 502 de la figure 3.1 est représentée par la description : tcf_banking_and_savings = v & hikes = v & prime = v & rate = v & said = v & pct = v & effective = v, et non plus par le vecteur proposé table 3.1.

Faire de l'absence une valeur par défaut conduit à modifier la formulation des exemples.

Mais cela a aussi une conséquence sur l'exploration de l'espace de généralisation : l'absence par défaut devient donc un présupposé d'apprentissage additionnel. Dans le contexte de l'apprentissage de règles, nous n'explorons pas, par exemple, les hypothèses qui suivent les modèles $\mathsf{wheat} = v\ \&\ \mathsf{winter} = v\ \&\ \mathsf{soft} = \mathrm{f} \rightarrow \textit{wheat} = v$ (blé) et $\mathsf{vs} = \mathrm{v} \rightarrow \textit{acq} = \mathrm{f}$ (fusions/rachats d'entreprises). En somme, nous imposons une contrainte sur la forme des généralisations à engendrer et nous nous servons de cette contrainte pour réduire le langage des instances.

On peut observer la distinction entre les rôles de la présence et de l'absence en recherche d'informations. La représentation vectorielle adoptée pour la catégorisation de textes associe en général un poids nul aux unités n'apparaissant pas dans un texte. Les classifieurs linéaires sont par nature additifs et ne sont donc pas affectés par les unités absentes. Toutes les méthodes en recherche documentaire n'ignorent pas l'absence : l'indexation probabiliste utilise de manière explicite l'absence des unités pour représenter des textes. Lors de la recherche d'un document, la requête est comparée aux documents de la collection ; les mesures de similarité employées sont souvent normalisées par rapport à la longueur des textes. L'absence joue alors un rôle indirect.

Remarquons finalement que le rôle donné à l'absence des mots relève du statut de la négation dans le raisonnement et la représentation en I.A. On peut, par exemple, associer l'absence par défaut à l'hypothèse du monde clos : si une propriété n'est pas vérifiée, alors elle est fausse.

7.2 SCAR : un algorithme de réduction employant l'absence par défaut

Nous proposons à présent un algorithme de réduction, qui instancie le modèle décrit au chapitre 4 et s'appuie sur le présupposé d'absence par défaut. Cet algorithme constitue la deuxième étape de la transformation des textes en exemples d'apprentissage, dans le schéma des systèmes de catégorisation donné à la figure 1.1. Lorsque nous avons conçu SCAR, nous nous sommes attachée à construire un algorithme de réduction dédié à CHARADE. Nous verrons cependant que les choix effectués permettent de rattacher SCAR à AQ aussi bien qu'à CHARADE.

7.2.1 Les différentes caractéristiques entrant dans la réduction du langage des instances

La réduction du langage des instances s'articule autour de deux pôles : la construction d'attributs et la diminution du nombre d'attributs. Nous présentons ici SCAR comme outil pour réduire le nombre d'attributs : la réduction mise en œuvre incorpore les caractéristiques du modèle encapsulé dans une technique par filtrage. En particulier, le langage des hypothèses, allié à l'absence par défaut, est un fil directeur lors du choix des attributs. On peut cependant envisager d'utiliser SCAR pour construire de nouveaux attributs, de manière similaire aux travaux décrits dans [PH90]. Nous n'avons pas explorer cette direction.

7.2.1.1 Présupposé de l'apprenant

Nous avons décrit le modèle encapsulé dans la section 4.2.3, tout en motivant son inadéquation pour les données textuelles : il fait en effet appel à l'apprenant pour évaluer la qualité d'un ensemble d'attributs, ce qui peut s'avérer aussi coûteux que l'apprentissage lui-même. Comme ce modèle s'avère plus efficace qu'un filtrage a priori dans un contexte classique d'apprentissage, nous souhaitons l'évaluer dans le cadre de la catégorisation. En conséquence, nous adoptons l'idée que la réduction du langage des instances doit prendre en compte les propriétés de l'apprenant. Nous proposons donc un algorithme de filtrage qui incorpore le présupposé d'apprentissage de CHARADE.

Nous nous intéressons en particulier à l'aspect du présupposé qui permet de restreindre le parcours de l'espace de généralisation. Ceci comprend à la fois le langage des hypothèses et les heuristiques de parcours mises en œuvre par des apprenants de règles tels que AQ ou CHARADE. Nous retenons les caractéristiques suivantes :

- Le concept appris est défini par un ensemble de règles. L'exploration de l'espace de généralisation est descendante, des règles les plus générales aux règles les plus spécifiques. On évolue dans cet espace par application d'opérateurs logiques : une règle est spécialisée par ajout d'un terme conjonctif en prémisse, un système par ajout d'une règle. Lorsque le langage des instances est un langage propositionnel, booléen, ce mécanisme est simple.

- Une règle n'est créée que si tous les exemples qui vérifient ses prémisses ont la même conclusion. L'hypothèse obtenue est alors cohérente avec les exemples. La cohérence est une contrainte forte dans le cas où les données comportent du bruit, car elle peut entraîner un sur-apprentissage. En effet, lorsqu'une règle générale couvre exemples et contre-exemples, il est nécessaire de la spécialiser pour exclure les contre-exemples. Il résulte de cette spécialisation des règles qui sont trop proches des données.

- Lorsque deux règles sont concurrentes, l'apprenant choisit la règle que vérifie un nombre maximal d'exemples. En effet, plus la couverture d'une règle est grande, plus elle a de chances d'être générale. Des apprenants tels que RIPPER ou C4.5Rules appliquent un critère plus complexe pour comparer les règles, par exemple le taux d'erreur lors de l'apprentissage.

7.2.1.2 L'absence par défaut : présupposé et connaissances sur les données

L'absence par défaut peut être employée pour contraindre la forme des règles engendrées par l'apprenant. Elle correspond aussi à un présupposé pour la réduction. Nous cherchons à engendrer des hypothèses où seuls les descripteurs relatifs à la présence interviennent. Dès lors, la méthode de réduction ne doit pas retenir des unités qui apporteraient une information par leur absence.

L'absence par défaut est une propriété des mécanismes de manipulation des textes, comme l'illustre l'adoption de l'index inverse pour archiver une collection documentaire (cf. chapitre 2). Nous cherchons en effet à déterminer ce qui est important dans les textes, lorsque l'on décide d'affecter une catégorie. La réduction s'attache à conserver les attributs dont la présence permet de caractériser les textes possédant une catégorie, ainsi que ceux dont la présence permet de discriminer une catégorie des autres catégories.

7.2.1.3 Connaissances du domaine : une élimination a priori d'unités d'indexation

Les approches statistiques pour l'indexation libre, présentées au chapitre 2, ont mis au point différents mécanismes de réduction de l'index. En particulier, il existe un certain nombre d'heuristiques pour éliminer brutalement des unités jugées a priori inintéressantes [Sal89], par exemple l'emploi d'un anti-dictionnaire (cf section 2.4) ou l'exclusion des mots de très faible fréquence sur l'ensemble de la collection.

Dans le cadre de la catégorisation de textes et du routage, différentes solutions sont avancées pour réduire l'index de la collection. Y. Yang propose une analyse préalable qui permet d'éliminer les unités trop fréquentes ou redondantes [Yan95, YW96]. Les travaux rapportés dans [SHP95] visent aussi à limiter la redondance dans la description des textes [1]. Ces travaux sont en fait des approches complètes pour la réduction du langage des instances.

Notre approche est plus simple et peut être qualifiée de «sens commun». Dès le début du traitement, nous excluons les unités dont la fréquence locale pour une classe est inférieure à la fréquence globale pour toutes les classes. Nous pensons en effet que de telles unités n'ont pas de pouvoir discriminant entre les catégories.

7.2.2 Mise en œuvre

Nous voulons prendre en compte les propriétés de l'apprenant lors de la réduction. Pourquoi, dans ce cas, ne pas utiliser un algorithme inductif ayant ces propriétés pour réaliser la sélection ? Dans un cadre général, cette approche est impensable [2]. Mais le présupposé que représente l'absence par défaut nous permet d'envisager un algorithme inductif pour réduire le langage des instances.

Différents travaux en sélection d'attributs s'appuient sur des algorithmes inductifs pour sélectionner les attributs à conserver. Par exemple, dans [Car93], la sélection par filtrage est réalisée au moyen d'arbres de décision. Les attributs apparaissant dans les nœuds de l'arbre induit sont conservés comme pertinents pour décrire les exemples d'apprentissage.

Nous adoptons un point de vue similaire. Scar induit des règles qui discriminent ou caractérisent les différentes catégories. Les attributs apparaissant dans les prémisses

1. La notion d'unités redondantes est assez vague. Cependant, l'indexation par sémantique latente considère comme redondantes les unités qui se regroupent autour d'un même axe factoriel.

2. À notre connaissance, le seul apprenant de règles capable de traiter de telles dimensions est RIPPER [Coh96b]. Cet algorithme recherche d'ailleurs principalement des descripteurs positifs pour former les règles.

des règles apprises ont donc un bon pouvoir discriminant ; la création de règles est une alternative à la notion de conflit introduite par [AD91].

7.2.2.1 Décomposition en sous-problèmes

La réduction du langage des instances, tout comme la sélection d'attributs en apprentissage, peut être perçue comme un problème de recherche dans un espace. En général, cet espace est l'ensemble des parties de l'ensemble des attributs composant le langage des instances [Lan94]. Dans le cadre de la catégorisation de textes, cette recherche peut se décomposer en sous-problèmes. Dans un premier temps, nous adoptons une approche locale pour la réduction. Puis nous tirons parti de l'absence par défaut.

Réduction locale à chaque catégorie

L'objectif de la catégorisation est d'affecter 0, 1 ou plusieurs catégories parmi n à un texte. Ce type de classification est peu commun en apprentissage symbolique, où l'on affecte en général une unique valeur parmi n. Un moyen de résoudre ce problème est de considérer que l'affectation d'une catégorie à un texte ne dépend que de la représentation de ce texte et non des autres catégories qui lui sont affectées. Ceci permet de transformer un mécanisme d'affectation de n catégories en n processus d'affectation d'une catégorie.

La réduction tire parti de cette transformation. Au lieu de retenir le même ensemble d'attributs pour toutes les catégories, nous réduisons le langage des instances en fonction de chacune des catégories. C'est une approche communément adoptée en catégorisation de textes et que nous avons présentée section 1.2.2.1.

Une critique que l'on peut adresser à cette réduction locale est qu'elle semble faciliter la tâche de l'apprenant puisque l'on définit un langage des instances local, regroupant les attributs jugés intéressants pour une catégorie.

L'absence par défaut conditionne le parcours de l'espace de généralisation

L'hypothèse d'absence par défaut fait de la catégorisation de textes un problème atypique en apprentissage. Les exemples sont en effet décrits par des unités représentant un sous-ensemble du langage des instances ; de plus, on ne cherche à caractériser qu'une valeur de la classe : «possède la catégorie X».

Remarquons tout d'abord que l'absence par défaut n'accorde pas le même rôle aux exemples positifs et aux exemples négatifs. On réduit le langage des instances dans l'optique de caractériser et discriminer les exemples positifs, tandis que les exemples négatifs ne servent qu'à assurer la discrimination de la catégorie.

Nous pouvons ensuite observer que deux textes qui ne possèdent aucune unité d'indexation en commun ne pourront pas être généralisés par une même règle. Trouver les attributs qui caractérisent ces deux exemples revient à trouver les attributs qui caractérisent le premier, puis ceux qui caractérisent le second.

Un exemple jouet

Ces deux observations ont pour conséquence une réduction de l'exploration de l'espace de généralisation. Nous les illustrons sur le treillis des descriptions introduit dans le cadre formel de CHARADE. Considérons les exemples suivants :

$$e_1: \quad a \wedge b \wedge c \quad \oplus \qquad\qquad e_2: \quad d \wedge e \wedge f \quad \oplus$$
$$e_3: \quad a \wedge d \wedge g \quad \ominus \qquad\qquad e_4: \quad b \wedge f \quad \oplus$$

a, b, c, d, e, f et g sont des descripteurs du type $\mathsf{unit\acute{e}} = \mathsf{v}$ et $\oplus$ représente le concept «possède la catégorie X», tandis que $\ominus$ correspond au concept «possède la catégorie Y».

Pour trouver les attributs permettant de caractériser la classe $\oplus$, nous cherchons à généraliser les exemples e_1, e_2 et e_4, en respectant l'hypothèse d'absence par défaut. Il apparaît clairement que les exemples e_1 et e_2 ne pourront pas être décrits au moyen d'un langage des instances identique, puisqu'il n'existe pas de règle qui couvre ces deux exemples.

Au lieu d'explorer le treillis engendré par $\{a, b, c, d, e, f, g\}$ pour trouver les attributs caractéristiques de $\oplus$, nous pouvons n'explorer que les treillis engendrés par les descriptions de e_1, e_2 et e_4. Comme nous ne cherchons pas à caractériser e_3 à cette étape, nous n'explorons pas le treillis engendré par sa description.

Ce mécanisme permet d'étudier un nombre restreint de généralisations. Dans le pire des cas, nous explorons 17 généralisations possibles, tandis que le treillis initial comporte 127 nœuds[3]. Une estimation plus réaliste ne comptabilise pas les nœuds du treillis qui ne couvrent pas d'exemples : le nombre de nœuds du treillis initial devient 20 tandis que celui des sous-treillis reste le même. Supposons à présent que la classe qui nous intéresse est $\ominus$. Le nombre d'éléments du treillis engendré par la description de e_3 est alors 7 ; ce chiffre est toujours à comparer avec les 20 nœuds du treillis initial.

Reformulation de la réduction

Ainsi, l'approche inductive que nous proposons pour la réduction du langage des instances prend en compte les objectifs de la tâche à réaliser par l'apprenant, à savoir caractériser l'affectation d'une catégorie et discriminer cette catégorie des autres catégories. Au lieu de parcourir l'espace de généralisation engendré par le langage des instances, elle effectue plusieurs explorations d'espaces plus petits, engendrés par la description de certains exemples. La figure 7.1 illustre cette reformulation.

Nous verrons dans quelle mesure cette reformulation restreint la taille de l'espace à explorer. Notons cependant que l'on envisage plusieurs fois la même généralisation, lors du parcours de deux treillis locaux. En ce sens, SCAR n'est pas optimal. Considérons en effet quelques cas représentés dans la figure 7.1 :

- les treillis $\mathcal{D}_1$ et $\mathcal{D}_2$ sont disjoints. Afin de sélectionner des attributs qui couvrent à la fois e_1 et e_2, il faut explorer ces deux treillis.

3. Le nombre de nœuds d'un treillis construit à partir de n descripteurs est $2^n - 1$.

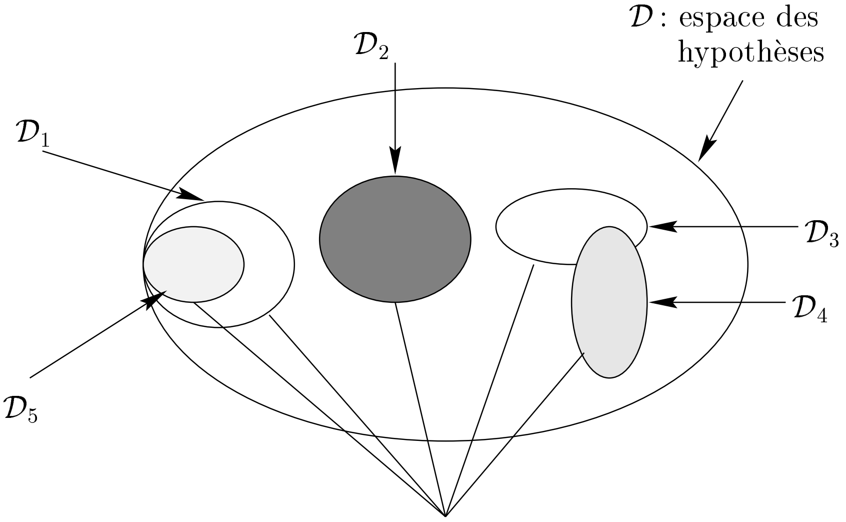


FIG. 7.1 - *Décomposition en sous-problèmes : réduction de l'espace des hypothèses*

– les treillis ne sont pas disjoints, par exemple $\mathcal{D}_5$ et $\mathcal{D}_1$, ou encore $\mathcal{D}_3$ et $\mathcal{D}_4$. Dans ce cas, la partie commune aux deux treillis, $\mathcal{D}_5$ ou $\mathcal{D}_3 \cap \mathcal{D}_4$, peut être explorée deux fois. On rencontre un tel cas lorsque les attributs sélectionnés lors de l'exploration de l'un des deux treillis ne permettent pas de caractériser l'exemple engendrant l'autre treillis, c'est-à-dire lorsque l'hypothèse retenue appartient à la différence des treillis.

Dans le pire des cas, pour chaque catégorie c, le nombre d'hypothèses à étudier pour SCAR est de l'ordre de $m2^{l_{max}}$, où m est le nombre d'exemples positifs de c et l_{max} est le nombre de descripteurs compris dans l'exemple le plus long. Le nombre de nœuds du treillis original est de l'ordre de 2^n, où n est le nombre d'attributs dans le langage des instances. Sur la collection Reuters, le nombre d'unités appartenant à l'index est d'environ 22000, le texte le plus long comporte 500 unités[4] et le nombre d'exemples d'apprentissage est environ de 8000. Le nombre de nœuds explorés par SCAR est alors proche de 2^{516}, (en considérant que $m = 2^{16}$) tandis que la taille originale du treillis est de 2^{22000}.

7.2.2.2 Heuristiques

Le nombre de généralisations que SCAR doit étudier dans le pire des cas reste important. Il est donc nécessaire de restreindre l'exploration des différents treillis. Nous pouvons jouer sur deux facteurs pour diminuer le nombre de généralisations à étudier : le nombre de treillis à parcourir et le nombre de nœuds à envisager dans chaque treillis.

Nous avons observé (cf. figure 7.1) que certains treillis se recoupaient. Lorsque la généralisation retenue se situe à l'intersection de plusieurs treillis, nous trouvons des

4. Le nombre d'unités par texte est en moyenne de 80.

attributs qui permettent de caractériser plusieurs exemples. Il n'est donc pas nécessaire de parcourir tous les treillis. Par exemple, si la généralisation retenue pour le treillis $\mathcal{D}_3$ appartient aussi à $\mathcal{D}_4$, SCAR n'explore pas $\mathcal{D}_4$.

Plutôt que de choisir les exemples de manière aléatoire, nous avons préféré étudier les exemples les plus longs en premier. Si nous nous référons à la figure 7.1, cela peut sembler contradictoire : en effet, toute hypothèse de $\mathcal{D}_5$ est aussi une hypothèse de $\mathcal{D}_1$. Dans une application réelle, un tel cas est rare. De plus, nous avons observé que les textes les plus durs à caractériser étaient les textes les plus courts.

Le deuxième élément est lié au présupposé de l'apprenant. Une des caractéristiques importantes de CHARADE ou de AQ est de considérer la couverture d'une règle comme mesure de son pouvoir de généralisation. Ceci transparaît dans le critère de choix entre généralisations concurrentes. SCAR adopte aussi cette stratégie, qui se répercute sur les attributs sélectionnés : on retient en priorité les attributs fréquents [5]. Ceci permet de ne pas explorer le treillis dans son intégralité. Après différents essais, nous avons rejeté les stratégies d'exploration de CHARADE (en largeur d'abord) et de AQ (recherche en faisceau) : CHARADE parcourt le treillis avec l'objectif de couvrir tous les exemples, ce qui ne nous intéresse pas ici, tandis que nous avons trouvé la recherche en faisceau trop sensible au nombre de généralisations conservées à chaque étape.

Nous adoptons une stratégie de recherche en profondeur d'abord, dont certaines branches sont coupées car elles ne sont pas d'assez bonne qualité (i.e. «branch and bound»). La qualité d'une hypothèse est définie par un couple (couverture, précision). La couverture est le nombre d'exemples que couvre une règle. La précision est la proportion d'exemples négatifs couverts par les attributs apparaissant dans l'hypothèse pris individuellement [6]. Une règle r_1 est meilleure que r_2 si la couverture de r_1 est strictement supérieure à la couverture de r_2, ou si lorsque les couvertures sont égales, la précision de r_1 est supérieure à celle de r_2.

Cette exploration est réalisée après exclusion d'un certain nombre de termes en fonction des critères fréquentiels de la section 7.2.1.3.

7.2.2.3 Paramètres

Nous avons conservé certains des paramètres de CHARADE qui permettent de contrôler l'exploration des treillis, en particulier la redondance et le fractionnement. Rappelons que la redondance conditionne la couverture des exemples. Nous explorons une seule fois le treillis engendré par un exemple ; pour couvrir λ fois cet exemple, nous retenons alors les λ meilleures règles. Dans le corps de l'algorithme, nous choisissons les exemples qui ne sont pas couverts λ fois, ce qui est l'approche de CHARADE pour traiter la redondance.

Les deux autres paramètres limitent l'exploration et sont employés de façon similaire dans CHARADE. Il s'agit de la taille maximale des prémisses (la profondeur du treillis) et

5. Nous ne sommes pas assurée que ces attributs soient les plus fréquents : la conjonction des attributs maximisent la couverture, pas les attributs pris individuellement.

6. Soit $d = d_0 \wedge \ldots \wedge d_n$, $|E|$ le nombre d'exemples d'apprentissage, $E/\gamma(c)$ correspond à l'ensemble des exemples négatifs. La précision de d est alors définie par $\frac{|E|^n}{\prod_{i=0}^{i=n} |(E/\gamma(c)) \cap \gamma(d_i)|}$

Soit E un ensemble d'exemples étiquetés
Soit RÈGLES et ATTRIBUTS, deux ensembles initialement vides
Pour chaque instance positive I de E
  Si I est couvert moins de λ fois
  Alors soit DESCRIPTEURS l'ensemble des descripteurs apparaissant dans I
    Réduire DESCRIPTEURS à l'aide d'heuristiques
    Meilleur_Ensemble_Règles = SCAR(E,k,b,λ)
    RÈGLES = RÈGLES $\cup$ Meilleur_Ensemble_Règles
ATTRIBUTS = $\{ d \mid d \in$ prémisse(R), $\forall$ R $\in$ RÈGLES $\}$
Retour(ATTRIBUTS)

Procédure SCAR*(E,k,b,λ)*

  Soit ENSEMBLE_RÈGLES, un ensemble initialement vide
  Soit PILE une pile contenant la description vide $\top$
  Tant que PILE n'est pas vide
    Dépiler Elt, le sommet de PILE
    Soit E' l'ensemble des instances couvertes par Elt
    Si toutes les instances de E' ont la même étiquette C
    Alors Ajouter la règle Elt $\rightarrow$ C à ENSEMBLE_RÈGLES
      Incrémenter la couverture des instances de E'
    Sinon Spécialiser_nœud(Elt,PILE,E,k,b,λ)
    Si Elt $\in$ DESCRIPTEURS
    Alors Enlever Elt de l'ensemble DESCRIPTEURS
  Retourner les λ meilleures règles de ENSEMBLE_RÈGLES

Procédure Spécialiser_nœud(Elt,PILE,E,k,b,λ)

  Soit QUEUE l'ensemble $\{$ Elt $\wedge$ x $\mid$ x $\in$ DESCRIPTEURS $\}$
  Pour chaque description D de QUEUE
    Si le nombre de descripteurs dans D est plus petit que k et
    D couvre au moins b instances
    Alors Empiler la description D sur PILE

FIG. 7.2 - *Le noyau de l'algorithme* SCAR

du nombre minimal d'exemples couverts pour engendrer une règle (le fractionnement). Dans CHARADE, le fractionnement est donné sous la forme d'un pourcentage. Notre approche est de proposer différentes fonctions du nombre d'exemples positifs de la classe.

7.2.3 SCAR et les algorithmes de sélection en apprentissage

L'originalité de SCAR tient à la notion d'absence par défaut, qui intervient aussi bien dans la représentation des exemples que dans le mécanisme de recherche des attributs à retenir. SCAR est un algorithme qui intervient en amont de l'apprenant. À ce titre, il appartient à la classe des méthodes de sélection par filtrage. Nous avons cependant souhaiter conserver des propriétés du modèle encapsulé.

Des différentes caractéristiques du modèle encapsulé, nous avons retenu le principe, c'est-à-dire prendre en compte le présupposé d'apprentissage, mais nous avons rejeté la mise en œuvre qui fait appel à la validation croisée et à l'apprenant car elle est trop coûteuse dans le contexte de la catégorisation. Le modèle encapsulé est un modèle générique, qui peut s'adapter à n'importe quel apprenant. L'approche que nous proposons perd cette souplesse : elle est spécifique à une famille d'apprenants particuliers, les algorithmes AQ et Charade. Le choix d'attributs effectué par Scar est sensible à la qualité de l'ensemble d'apprentissage. La solution apportée dans le modèle encapsulé est la validation croisée, qui permet, d'après [Koh95], d'obtenir un ensemble d'attributs qui ne soit pas trop spécifique à l'ensemble d'apprentissage. Dans le cadre de la catégorisation, cette solution présente des difficultés de mise en œuvre en raison de la forte disproportion entre les exemples positifs et les exemples négatifs. Nous verrons lors de l'analyse de la réduction effectuée par Scar que diminuer le nombre d'exemples positifs dans l'ensemble d'apprentissage est nuisible pour certaines catégories.

On peut distinguer deux types d'algorithmes dans la sélection par filtrage : ceux qui s'appuient sur des critères statistiques et ceux qui utilisent la cohérence avec les données comme critère d'arrêt. Il est difficile de comparer autrement que de manière empirique Scar avec les approches statistiques telles que Relief [KR92] ou la sélection au moyen de l'information mutuelle.

Scar s'apparente à l'approche de C. Cardie [Car93], dans la mesure où c'est un algorithme inductif qui effectue la réduction du langage des instances. C. Cardie sélectionne les attributs qui apparaissent dans les nœuds d'un arbre de décision : son approche ne permet pas de prendre en compte l'absence par défaut comme présupposé additionnel.

Les algorithmes FOCUS et FOCUS-2 [AD91] implémentent un présupposé d'apprentissage particulier, appelé *MIN-FEATURES bias* par les auteurs, qui correspond à la sélection du plus petit ensemble suffisant d'attributs. Un ensemble d'attributs est suffisant si deux exemples ayant la même description appartiennent à la même classe, ce qui diffère de la notion de cohérence sur laquelle s'appuie Scar : en effet, une règle peut être cohérente sans pour autant engendrer un ensemble suffisant (cf. figure 7.3). Afin de trouver cet ensemble, les auteurs définissent la notion de conflit et explorent l'espace des sous-ensembles d'attributs qui résolvent les conflits engendrés sur l'ensemble d'apprentissage. Un conflit mémorise les attributs qui permettent de distinguer deux instances n'ayant pas la même classe. Scar et FOCUS se distinguent aussi par leur comportement lorsqu'un tel ensemble n'existe pas. Alors que FOCUS ne produit aucun résultat, Scar fournit un ensemble qui décrit une majeure partie de l'ensemble d'apprentissage. Finalement, l'objectif de FOCUS est de trouver le plus petit ensemble ayant cette propriété, alors que Scar cherche à limiter le nombre d'attributs retenus, mais engendre un ensemble non minimal. Même si FOCUS n'inclut pas l'absence par défaut comme présupposé, il est possible de modifier l'algorithme pour prendre en compte ce présupposé, par exemple en contraignant la génération des conflits.

SCAR trouve la règle cohérente suivante :
si gold $= v \wedge$ silver $= v \wedge$ ounce $=v$ *alors gold* $= v$

Pourtant l'ensemble d'attributs pour la catégorie *gold* n'est pas suffisant, comme l'illustrent ces deux textes :

Dépêche	Représentation
TOPICS: *gold silver* **END-TOPICS** CBT METALS/FINANCIALS DELIVERY LOCATIONS Chicago, April 3 - The following deliveries are scheduled for April 6 against Chicago Baord of Trade Futures - Silver - 392 lots at Chicago, Illinois. Kilo Gold - 21 lots. 16 at Chicago, Illinois, 5 at New York, New York. Reuter	defex T1765 { (*silver* == true) (*gold* == true) (gold == true) (silver == true) }
TOPICS: *earn* **END-TOPICS** <KIENA GOLD MINES LTD> 4TH QTR NET TORONTO, March 11 - Shr 17 cts vs 16 cts Net 1,019,000 vs 985,000 Revs 7,997,000 vs 7,492,000 YEAR Shr 1.18 dlrs vs 64 cts Net 6,959,000 vs 3,778,000 Revs 36.5 mln vs 29.8 mln Reuter	defex T8934 { (*earn* == true) (gold == true)}

FIG. 7.3 - *L'ensemble d'attributs engendré par* SCAR *n'est pas suffisant, même si les règles sont cohérentes*

Dans le pire des cas, la complexité de FOCUS est de l'ordre de mpn^p, où m est le nombre d'exemples, n le nombre d'attributs et p le nombre d'attributs pertinents. En supposant connu le nombre d'attributs pertinents, la complexité de SCAR est celle de l'exploration de m treillis de profondeur l_{max}, qui est de l'ordre de $mn^{l_{max}}$, où p et l_{max} sont du même ordre de grandeur.

7.2.4 SCAR et les apprenants symboliques à base de règles

L'algorithme SCAR représente une approche pour la réduction du langage des instances. C'est aussi un apprenant naïf, applicable à la catégorisation de textes.

7.2.4.1 Où se situe la parenté avec CHARADE ?

À l'origine, SCAR était une approche pour la réduction incarnant deux présupposés : le présupposé d'apprentissage de CHARADE et le présupposé d'absence par défaut. Nous montrons ici deux propriétés relatives aux treillis qui permettent de mieux cerner les relations entre SCAR et CHARADE.

Les connections de Galois restent inchangées par projection sur le treillis engendré par la description d'un exemple

Nous introduisons tout d'abord la notation [ ], qui permet de représenter les éléments apparaissant dans une description, avant de montrer l'invariance des connections de Galois par projection :

NOTATION 7.1 *Soit d un élément du treillis $\mathcal{D}$. Nous notons $[d]$ l'ensemble des éléments apparaissant dans d.*

Par exemple, si $d = a \wedge b \wedge d$, alors $[d] = \{a, b, d\}$.

PROPOSITION 1
Soit $\mathcal{E}$ le treillis des exemples, $\mathcal{D}$ le treillis des descriptions et $\mathcal{D}_i$ le treillis des descripteurs engendré par l'ensemble $D_i = [\beta(e_i)]$, la connection de Galois entre $\mathcal{E}$ et $\mathcal{D}$ reste inchangée sur le sous-espace $\mathcal{D}_i$, c'est-à-dire :
soient $\beta_i : \mathcal{E} \to \mathcal{D}_i$, $\gamma_i : \mathcal{D}_i \to \mathcal{E}$, $\beta : \mathcal{E} \to \mathcal{D}$ et $\gamma : \mathcal{D} \to \mathcal{E}$, alors

$$\forall d \in \mathcal{D}_i, \quad \beta_i \circ \gamma_i(d) = \beta \circ \gamma(d)$$

Cette proposition n'est pas vérifiée dans le cas général, c'est-à-dire lorsqu'il n'y a pas de condition sur D_i.

Il est facile de trouver une illustration de la dernière partie de la proposition 1. Considérons l'exemple suivant :

$$e_1 : a \wedge b \wedge c \wedge d$$
$$e_2 : a \wedge c$$
$$e_3 : a \wedge b \wedge d$$

et le sous-ensemble de descripteurs $D_i = \{a, b\} \subset \{a, b, c, d\}$. Dans ce cas, $\beta_i \circ \gamma_i(a \wedge b) \neq \beta \circ \gamma(a \wedge b)$. En effet, $\beta_i \circ \gamma_i(a \wedge b) = a \wedge b$, alors que $\beta \circ \gamma(a \wedge b) = a \wedge b \wedge d$.

Montrons à présent la première partie de la proposition 1. Pour cela, nous avons besoin de rappeler et de montrer différentes propriétés préliminaires.

PROPRIÉTÉ 7.1 *Soient D_i et D_j deux ensembles de descripteurs tels que D_i est inclus dans D_j, alors le treillis induit par D_i est inclus dans le treillis induit par D_j.*

$$\forall D_i \subset D_j, \quad \mathcal{D}_i \subset \mathcal{D}_j$$

La preuve de cette propriété est triviale grâce à l'existence d'un isomorphisme entre le treillis induit et l'ensemble des parties.

Pour tout treillis $\mathcal{D}_i$, engendré à partir de $[\beta(e_i)]$, la propriété suivante est vérifiée.

PROPRIÉTÉ 7.2 *Le treillis $\mathcal{D}_i$ est un sous-treillis de $\mathcal{D}$ dont les éléments sont inférieurs à $\beta(e_i)$*

$$d \in \mathcal{D}_i \quad \Leftrightarrow \quad d \in \mathcal{D} \text{ et } d \leq \beta(e_i)$$

$d \in \mathcal{D}_i \quad \Rightarrow \quad d \in \mathcal{D}$ et $d \leq \beta(e_i)$ est une conséquence de la propriété 7.1 et de la construction de $\mathcal{D}_i$. En effet, le treillis $\mathcal{D}_i$ est inclus dans le treillis des descriptions $\mathcal{D}$. De plus, l'élément le plus général du treillis $\mathcal{D}_i$ est par construction $\beta(e_i)$.

Soit $d \in \mathcal{D}$. Montrons que si $d \notin \mathcal{D}_i$, alors $d \nleq \beta(e_i)$. Considérons trois cas où $d \notin \mathcal{D}_i$:

1. $[d] \cap D_i = \emptyset$

   Montrons la propriété suivante : $\forall d_i \in \mathcal{D}_i, d_i \curlywedge d = \top$, c'est-à-dire que les éléments de $\mathcal{D}_i$ et d ne sont pas comparables.

   Utilisons l'isomorphisme existant entre le treillis des descriptions et l'ensemble des parties de l'ensemble des descripteurs. Cela revient à montrer que

   $$\forall d_i \in \mathcal{D}_i, [d_i] \cap [d] = \emptyset$$

   Supposons que $\forall d_i \in \mathcal{D}_i \exists e \in D | [d] \cap [d_i] = \{e\}$. Alors $e \in [d]$ et $e \in [d_i]$. Ceci entraîne $[d_i] \cap [d] \neq \emptyset$, d'où $[d] \cap D_i \neq \emptyset$ puisque $[d_i] \subseteq D_i$.

   $\beta(e_i)$ vérifie cette propriété. Nous avons donc $d \nleq \beta(e_i)$.

2. $[d] \cap D_i = D_i$

   Dans ce cas, $\beta(e_i) \curlywedge d = \beta(e_i)$. En effet, d peut se décomposer en $\beta(e_i) \curlyvee t$, où $t \in \mathcal{D}$ vérifie $[t] \cap D_i = \emptyset$. Nous en déduisons que $d \geq \beta(e_i)$.

3. $[d] \cap D_i \neq \emptyset$ et $[d] \cap D_i \neq D_i$

   Si $d \leq \beta(e_i)$, alors $d \curlywedge \beta(e_i) = d$. d peut se décomposer de la manière suivante : $\exists x \in \mathcal{D}_i, y \notin \mathcal{D}_i | y \neq \top$ et $d = x \curlyvee y$. Nous avons donc $x \leq d$ et $x \neq d$. Comme le treillis $\mathcal{D}$ est distributif, $\beta(e_i) \curlywedge d = (\beta(e_i) \curlywedge x) \curlyvee (\beta(e_i) \curlywedge y)$. Or $\beta(e_i) \curlywedge x = x$ car $x \in \mathcal{D}_i$ et y vérifie $[y] \cap D_i = \emptyset$. Il s'en suit que $\beta(e_i) \curlywedge d = x \neq d$ par hypothèse. D'où $d \not\leq \beta(e_i)$.

Introduisons à présent la notion de projection sur un sous-espace de descriptions par analogie à la projection sur les ensembles. Soit $\pi_i : \mathcal{D} \to \mathcal{D}_i$ la projection sur $\mathcal{D}_i$, cette application vérifie la propriété suivante:

$$\forall d \in \mathcal{D}_i, \quad \pi_i(d) = d \tag{7.1}$$

Nous pouvons définir les fonctions β_i et γ_i à partir de β, γ et pi_i de la façon suivante :

– γ_i est la restriction de l'application γ à l'ensemble $\mathcal{D}_i$, c'est-à-dire :

$$\forall d \in \mathcal{D}_i, \quad \gamma_i(d) = \gamma(d) \tag{7.2}$$

– β_i est la projection de l'application β sur le treillis $\mathcal{D}_i$:

$$\forall e \in \mathcal{E}, \quad \beta_i(e) = \pi_i \circ \beta(e) \tag{7.3}$$

Par définition des applications γ_i et β_i, nous avons

$$\forall d \in \mathcal{D}_i, \quad \beta_i \circ \gamma_i(d) \in \mathcal{D}_i \tag{7.4}$$

Montrons à présent que

$$\forall d \in \mathcal{D}_i, \quad \beta \circ \gamma(d) \in \mathcal{D}_i \tag{7.5}$$

Rappelons les propriétés suivantes de connexions de Galois de γ et β entre les treillis $\mathcal{E}$ et $\mathcal{D}$ [Gan92]:

$$\begin{array}{rc} \forall e \in \mathcal{E}, & \beta \circ \gamma \circ \beta(e) = \beta(e) \\ \forall d \in \mathcal{D}, d' \in \mathcal{D}, & d \leq d' \Rightarrow \beta \circ \gamma(d) \leq \beta \circ \gamma(d') \end{array}$$

Il en découle que :

$$\begin{array}{rcl} \forall d \in \mathcal{D}_i, \quad \beta \circ \gamma(d) & \leq & \beta \circ \gamma \circ \beta(e_i) \\ & \leq & \beta(e_i) \end{array}$$

D'après la propriété 7.2, comme $\beta \circ \gamma(d) \in \mathcal{D}$ et $\beta \circ \gamma(d) \leq \beta(e_i)$, nous déduisons que $\beta \circ \gamma(d) \in \mathcal{D}_i$.

Complétons la démonstration de la proposition 1 en remarquant que :

$$\begin{array}{rll} \forall d \in \mathcal{D}_i, \quad \beta_i \circ \gamma_i(d) & = \beta_i \circ \gamma(d) & \text{par définition de } \gamma_i \\ & = \pi_i \circ \beta \circ \gamma(d) & \text{par définition de } \beta_i \\ & = \beta \circ \gamma(d) & \text{par définition de } \pi_i \text{ et } (7.5) \end{array}$$

Cette notion de projection sur un treillis engendré par la description d'un exemple permet de décrire Scar et de lier l'espace exploré par Scar à l'espace exploré par Charade. En fait, cette notion de projection caractérise aussi les liens entre AQ et Charade.

Les règles engendrées par SCAR sont les seules à respecter les contraintes d'absence par défaut liées aux langages des instances et des hypothèses

Nous n'avons pas, jusqu'ici, fait intervenir les contraintes liées au présupposé d'apprentissage. Ces contraintes ont la conséquence suivante : les règles engendrées sur les treillis locaux sont les seules à respecter la notion d'absence par défaut.

Par construction, les éléments des treillis locaux $\mathcal{D}_i$ et global $\mathcal{D}$ vérifient les contraintes du langage des hypothèses. En effet, chaque exemple est représenté à l'aide des descripteurs correspondant uniquement à la présence ; ce sont ces descripteurs qui apparaissent dans la première couche du treillis. Tout autre descripteur $a = f$ n'est pas pris en compte dans la couche initiale car $\gamma(a = f) = \emptyset$, c'est-à-dire que ce descripteur ne peut permettre d'engendrer une règle.

Soit d un élément quelconque de $\mathcal{D}$. Nous montrons à présent que si d permet d'engendrer une règle $d \to c$, alors d appartient à un treillis local $\mathcal{D}_i$, c'est-à-dire qu'il existe e_i tel que $d \leq \beta(e_i)$.

Montrons la contraposée : s'il n'existe aucun exemple e_i tel que $d \leq \beta(e_i)$, alors on ne peut engendrer une règle $d \to c$. Remarquons tout d'abord que le fait qu'il n'existe aucun exemple e_i tel que $d \leq \beta(e_i)$ entraîne que $\gamma(d) = \emptyset$. Or, par construction, la règle $d \to c$ est engendrée si $\beta \circ \gamma(d)/d = c$. Par hypothèse, $\beta \circ \gamma(d) = \beta(\emptyset) = \top$. Il n'est donc pas possible d'engendrer la règle $d \to c$, car $\beta \circ \gamma(d)/d = \top/d = \top$.

7.2.4.2 SCAR : une implémentation «creuse» d'une version booléenne de l'algorithme AQ

L'algorithme CHARADE est à l'origine de SCAR. Cependant, les différents choix que nous avons effectués pour implémenter le présupposé d'absence par défaut nous conduisent à considérer SCAR comme proche des algorithmes de la famille AQ.

SCAR possède en effet des caractéristiques communes à cette famille. En particulier, l'exploration de l'espace de généralisation est réalisée à partir des exemples (et non directement, comme le fait CHARADE) de manière explicite par la mise en œuvre d'une stratégie *séparer pour régner*. La seconde caractéristique est la cohérence de la généralisation obtenue à partir de l'exemple graine.

Cependant, SCAR n'appartient pas à proprement parler à la famille de AQ. En effet, une fois l'exemple graine choisi, l'exploration de l'espace de généralisation est différente : alors que AQ utilise une recherche en faisceau pour engendrer une généralisation cohérente, SCAR met en œuvre une stratégie d'exploration par évaluation (*branch and bound*). Cette approche est plus coûteuse car le nombre de généralisations conservées à chaque étape n'est pas constant, mais elle permet d'obtenir des règles qui ont une plus grande couverture [7]. Nous avons par ailleurs restreint le langage de description à un langage booléen, qui nous permet de décrire les textes en fonction de la présence ou de l'absence d'unités d'indexation. Finalement, AQ ne permet pas, à notre connaissance, de contraindre le

7. G. Raškinis a comparé les deux approches [Raš95]. Les résultats nous ont amenée à ne retenir que l'exploration par évaluation comme pertinente pour la réduction.

langage des hypothèses à vérifier la notion d'absence par défaut. Cette notion joue aussi bien sur le langage des instances que sur le langage des hypothèses : ceci nous permet d'envisager une implémentation creuse [8], où les exemples ne sont pas représentés par des vecteurs, mais par des ensembles, où seuls apparaissent les attributs décrivant l'exemple.

7.2.4.3 Autres apprenants symboliques

CHARADE, AQ et SCAR engendrent des hypothèses cohérentes. La cohérence est une contrainte forte face à des données réelles. L'algorithme CN2, un descendant de AQ, a relâché cette contrainte, en permettant la création de règles admettant quelques erreurs.

Les apprenants plus récents, tels que C4.5Rules ou RIPPER, utilisent une approche glouton, similaire à celle employée pour construire les arbres de décision : le littéral choisi pour la spécialisation est sélectionné au moyen d'un critère numérique (par exemple l'information mutuelle) mais l'ensemble des exemples est cependant manipulé au moyen d'une approche *séparer pour régner*, que l'on retrouve dans la majorité des apprenants de règles. Ils explorent donc moins d'hypothèses que ne le fait SCAR. De plus un élagage a posteriori de la base de règles permet de minimiser l'erreur de prédiction sur une base de validation (un sous-ensemble de la base d'apprentissage, mis de côté dans ce but). Cette phase a pour objectif d'éviter la sur-spécialisation de la base de règles, lorsque les données sont imparfaites. Ces aspects ne sont pas du tout pris en compte par SCAR.

7.3 RELIEF+ et IM : deux filtres numériques incarnant un présupposé différent

Par construction, SCAR implémente deux présupposés distincts : un présupposé relatif à l'apprenant, ici CHARADE, et un présupposé relatif au langage des hypothèses, l'absence par défaut.

Afin de délimiter l'influence de ces deux présupposés, nous comparons SCAR à deux approches pour la réduction, qui ont pour caractéristique commune d'ordonner les attributs de façon linéaire et de sélectionner ceux qui obtiennent le meilleur score.

La première est une version modifiée de RELIEF [KR92], qui incorpore la notion d'absence par défaut. La deuxième approche s'appuie sur l'information mutuelle comme critère de comparaison lors du classement des différents attributs.

7.3.1 RELIEF+

L'algorithme RELIEF [KR92] détecte les attributs qui sont statistiquement significatifs pour décrire un concept. Dans cette méthode, on considère qu'un attribut est statistiquement significatif si le poids qui lui est affecté est supérieur à un seuil τ. La partie importante de l'approche est le calcul des poids des attributs. L'algorithme s'articule

8. Nous employons le terme creux par analogie avec la notion de matrice creuse

autour des points suivants :

1. Choisir m exemples de façon aléatoire dans l'ensemble d'apprentissage
2. Pour chaque exemple i, trouver
   - son plus proche voisin ayant la même classe, noté e^+
   - son plus proche voisin n'ayant pas la même classe, e^-
3. Pour chaque attribut a, modifier son poids.
   - $w_a = w_a + \delta^2(a_i, a_{e^-}) - \delta^2(a_i, a_{e^+})$ où w_a est le poids de l'attribut a, a_x la valeur de l'attribut a pour l'exemple x. $\delta(a, b) = 0$ si $a = b$, 1 sinon.
4. Normaliser le vecteur de poids $w = \frac{w}{m}$ et sélectionner les attributs dont le poids est supérieur à un seuil $\tau > 0$

Relief+ [Raš95, MRG96] est une version simplifiée et modifiée de Relief. Relief+ correspond à une version simplifiée car nous ne nous intéressons qu'à des attributs booléens, alors que Relief envisage d'autres types d'attributs. Les modifications apportées sont relatives à l'absence par défaut et à la disproportion entre exemples positifs et exemples négatifs que l'on observe dans la majorité des problèmes de catégorisation. Elles interviennent lors du choix des m exemples et pour la mise à jour des poids :

1. Pour chaque exemple positif i^+, trouver
   - son plus proche voisin ayant la même classe, noté e^+
   - son plus proche voisin n'ayant pas la même classe, e^-
2. Pour tout attribut a, tel que $a_{i^+} = v$, modifier son poids
   - $w_a = w_a + \alpha(a_i, a_{e^-}) - \alpha(a_i, a_{e^+})$, avec les mêmes conventions que précédemment. $\alpha(a, b) = 1$ si $a \neq b$, -1 sinon.
3. Normaliser le vecteur de poids, ordonner les attributs par poids décroissant et sélectionner les θ' premiers attributs, vérifiant $\theta' \leq \theta$, $w_a > \tau$ et $\sum_{i=1}^{i=\theta'} w_{a_i} < \sigma$.

Les modifications apportées dans le calcul des poids consistent à ne pas calculer explicitement le poids des unités absentes. Nous utilisons cependant une partie de l'information sur l'absence d'une unité lors du calcul des poids, dans un souci de discrimination : pour distinguer deux textes de classes opposées, le poids d'une unité croît lorsqu'elle est présente dans l'un et absente de l'autre.

Par ailleurs, nous avons remarqué que la seule utilisation du seuil τ ne suffisait pas à obtenir un langage des instances réduit, de taille suffisamment petite pour l'application d'apprenants symboliques. Nous avons donc introduit un second seuil θ relatif au nombre d'attributs maximal conservés lors de la réduction. Finalement, le seuil σ traduit la qualité de l'ensemble sélectionné. Si une qualité suffisante est atteinte, il n'est pas nécessaire de sélectionner plus d'attributs.

U / C	$U = v$	$U = f$	
$C = v$	a	c	$a + c$
$C = f$	b	d	$b + d$
	$a + b$	$c + d$	$n = a + b + c + d$

TAB. 7.1 - *Table de contingence entre la catégorie C et l'unité U. a est le nombre de textes où U et C apparaissent ensemble.*

7.3.2 IM : filtrage utilisant l'information mutuelle comme critère

L'information mutuelle est souvent employée en catégorisation de textes comme critère d'évaluation pour réduire le langage des instances dans différents travaux [Lew92, MG95, Wie95]. En effet, dans le contexte de la théorie de l'information, la catégorisation peut être vue comme un processus de communication où les symboles en entrée sont les unités d'indexation et les symboles en sortie les catégories. On peut alors utiliser l'information mutuelle entre les symboles en entrée et les symboles en sortie pour réduire la taille du vocabulaire d'entrée.

Pour deux variables booléennes, l'information mutuelle est de la forme :

$$I(U,C) = \sum_{u=v,f} \sum_{c=v,f} p(U=u, C=c) \log_2 \frac{p(U=u, C=c)}{p(U=u) \times p(C=c)}$$

Il existe différentes applications de l'information mutuelle comme critère de sélection. En apprentissage, c'est l'un des principaux critères pour les explorations de l'espace de généralisation de type glouton, par exemple pour la construction d'arbres de décision. En statistique linguistique, l'information mutuelle, telle qu'elle est proposée par [CH90] pour déterminer des collocations, n'a pas la même définition que notre critère. On retrouve cependant notre définition de l'information mutuelle (*généralisée* [Gau95]) dans différentes approches pour déterminer les collocations sous la forme du logarithme du ratio de vraisemblance [DAS91, Dun93, Dai94].

Nous décrivons à présent comment se déroule la sélection des attributs. Il faut dans un premier temps évaluer les diverses probabilités pour chaque catégorie et chaque unité d'indexation. Ces probabilités sont estimées en fonction des fréquences de ces éléments dans l'ensemble d'apprentissage. Si l'on résume ces fréquences dans un tableau de contingence (cf. table 7.1), cela revient à : $p(U=v, C=v) = \frac{a}{n}$, $p(U=f, C=f) = \frac{d}{n}$, $p(U=v) = \frac{a+b}{n}$, $p(U=f) = \frac{c+d}{n}$, $p(C=v) = \frac{a+c}{n}$, $p(C=f) = \frac{b+d}{n}$. Puis, pour chaque catégorie, on sélectionne les τ unités ayant l'information mutuelle la plus forte. Le nombre τ est un paramètre fixé par l'utilisateur.

Le modèle pour représenter les textes est booléen : nous avons choisi de ne conserver que les coocurrences entre unité et catégorie et non pas d'utiliser le nombre de fois où une unité apparaît dans un texte ; ainsi, le nombre n est figé pour tout le corpus.

Il est important de remarquer que l'information mutuelle n'est pas compatible avec la notion d'absence par défaut. En effet, du fait de la symétrie entre les éléments a et d

tableau de contingence, l'absence d'une unité ou d'une catégorie intervient explicitement et avec le même poids que leur présence.

7.4 Analyse a priori de la réduction effectuée par les 3 approches

Nous cherchons, ici, à qualifier la réduction réalisé par chacune des approches envisagées. Ceci nous amène à évaluer les propriétés des attributs retenus : sont-ils discriminants, sont-ils fréquents dans le corpus, apparaissent-ils fréquemment avec la classe ?

Nous avons choisi d'illustrer les qualités des trois approches sur quatre catégories : *veg-oil* (huile végétale), *acq*, *gold* (or) et *silver* (argent). Les figures 7.4, 7.5, 7.6 et 7.7 montrent les unités sélectionnées par chaque approche, ainsi que les unités maximisant l'estimation de $p(U = v|C = v)$ sur la base d'apprentissage, c'est-à-dire les unités fréquentes pour une catégorie donnée. Par souci de clarté, nous avons fixé le nombre d'attributs à 50 pour IM, RELIEF+ et la fréquence. SCAR détermine dynamiquement le nombre d'unités retenues.

7.4.1 veg-oil illustre les particularités de SCAR et RELIEF+

La catégorie *veg-oil* représente une catégorie de fréquence moyenne dans la collection ; elle figure dans 87 dépêches de la base d'apprentissage. Sa caractéristique principale réside dans le fait qu'elle est rarement l'unique étiquette affectée à une dépêche : elle apparaît conjointement avec une autre catégorie dans 67 dépêches.

Nous pouvons observer figure 7.4 que le groupe formé des premières unités est similaire pour chaque approche. Il est intéressant cependant de noter que le terme palm est classé en premier par SCAR alors qu'il n'apparaît qu'en troisième et cinquième rang pour IM et RELIEF+.

Si nous considérons les tables de contingence reliant palm et vegetable à *veg-oil* (cf. table 7.2), le choix effectué par SCAR semble contre-intuitif : vegetable est un attribut plus discriminant que palm. Ce classement révèle une première différence entre SCAR et les deux autres approches. La fonction d'évaluation de la qualité d'un attribut dans IM ou RELIEF+ ne fait intervenir que cet attribut, alors que SCAR, qui construit des règles de production, sélectionne des attributs qui sont discriminants en conjonction avec d'autres.

Examinons à présent le comportement de RELIEF+. Pour un seuil $\tau = 0$, nous observons que le terme tonnes n'est jamais sélectionné, quelque soit le nombre θ d'attributs retenus. Quelle différence existe-t-il entre les unités tonnes et tax pour que RELIEF+ ne les évalue pas de façon semblable ? Sur les 41 dépêches possédant la catégorie *veg-oil* et l'unité tonnes, 35 dépêches possèdent à la fois un plus proche voisin positif et un plus proche voisin négatif ayant l'unité tonnes, tandis que seules 4 dépêches (sur 26) possédant l'unité tax ont cette particularité. Lorsqu'une unité apparaît ainsi à la fois dans l'exemple positif le plus proche et dans l'exemple négatif le plus proche, son score n'est pas modifié, puisque le gain obtenu pour le voisin positif est compensée par la perte liée au voisin négatif. Le terme tax apparaît dans un plus proche voisin positif 20 fois, tandis que tonnes

Im :

vegetable oils palm fats oil soybean tonnes community ec rbd veg european olein agriculture refined indonesia soyoil import cts bleached deodorised cotton edible cpo commodity tax vs tonne malaysian proposed india traders kernel olive indian corp cif ministers shr bennett jakarta meal opposed net imported soybeans crude coconut company gatt

Relief+ :

vegetable oils tax oil fats palm import community traders crude meal opposed soybean cif imported indonesian london ministers tonne commission corporation delegations ended european increases indian proposal proposed result states cargoes coconut corn diplomats ec estimated imports level meetings offset production proposals refined rejected south stc three vote wayne confident

Scar (43) :

palm oil vegetable oils tonnes fats community traders proposed soybean ec import indonesia ministers commodity european tender increases approved crude tax veg cotton indian rbd india indonesian refined wheat shortage kernel reporting soyoil meal olein tonne agriculture cpo corn oilseeds association calendar soybeans

Fréquence :

reuter said march oil vegetable oils tonnes 's palm u.s dlrs agriculture european trade community ec mln tax fats washington year april soybean new london exports import proposed told production added export traders crude government u.s. commodity pct products tonne two yesterday association expected indonesia prices states tender world countries

Fig. 7.4 - *Les meilleures unités pour la catégorie* veg-oil*, ordonnées par intérêt décroissant. Les nombres indiqués entrent parenthèses représentent le nombre d'unités sélectionnées lorsque celui n'est pas fixé a priori.*

tonnes		cotton		corn		wheat	
41	46	11	76	8	79	11	76
506	7196	43	7659	154	7548	232	7470

ec		palm		vegetable		tax	
26	61	33	54	49	38	26	61
176	7526	18	7526	11	7691	489	7213

TAB. 7.2 - *Tableaux de contingence reliant les termes* tonnes, cotton, corn, wheat, ec, palm, vegetable *et* tax *à la catégorie veg-oil. Les colonnes correspondent à la catégorie et les lignes aux unités.*

vs		shr		revs	
0	2037	0	1480	0	957
1651	4101	1651	4658	1651	5181

TAB. 7.3 - *Tableaux de contingence entre les unités* vs, shr, revs *et la catégorie acq. Les colonnes correspondent à la catégorie, les lignes aux unités.*

apparaît dans 2 plus proches voisins positifs et 4 plus proches voisins négatifs. Le score associé à tax est 20/87, tandis que celui associé à tonnes est -2/87.

Finalement, la réduction effectuée par IM, pour la catégorie *veg-oil*, mais aussi de manière générale, offre un compromis entre unités fréquentes (cooccurrentes) et unités discriminantes.

7.4.2 Réflexions engendrées par la réduction locale à acq

Nous avons choisi ensuite la catégorie *acq* car nous verrons qu'elle met en avant la difficulté à représenter les textes comme exemples d'apprentissage, et par là même, elle met en cause l'adéquation de la généralisation dans un tel contexte. La catégorie *acq* est la deuxième plus fréquente sur la base d'apprentissage, avec 1651 textes, après la catégorie *earn* (bénéfices des entreprises). Elle n'est affectée que 54 fois en même temps qu'une autre catégorie.

7.4.2.1 IM ne vérifie pas la notion d'absence par défaut

Parmi les unités retenues par IM, vs, shr et revs ne sont jamais sélectionnées par SCAR et RELIEF+. Si nous observons leurs tables de contingence (cf. table 7.3), il apparaît clairement qu'IM sélectionne ces unités parce que leur présence est un prédicteur de l'absence de la catégorie *acq*: on pourrait par exemple induire la règle logique *Si* vs $=$ *v* *alors acq* $=$ *f*. Ceci illustre bien le fait que la réduction effectuée au moyen de l'information mutuelle entre une unité et une catégorie ne vérifie pas la notion d'absence par défaut, puisqu'une unité est explicitement choisie pour son absence.

IM :

vs shares acquisition cts acquire stake shr company said qtr merger net corp year offer common revs acquired buy transaction unit group purchase stock outstanding sell terms note undisclosed bid shareholders 4th takeover record cash completed subsidiary loss completes investor disclosed div subject securities approval agreed tonnes filing management share

RELIEF+ :

stake group acquisition corp offer march commission acquire transaction outstanding merger bid exchange securities washington investor buy company sale unit york management previously owned takeover agreement filing shareholders subject bought shareholder holdings tender approval common purchase sell comment completed firm undisclosed buyout companies deal terms usair letter owns pacific seeking

SCAR (100 sur 293) :

pct company corp shares offer stake group acquisition acquire new merger stock share agreement common exchange outstanding commission bid buy securities washington agreed sell unit investor transaction approval york tender shareholders cash investment buyout terms comment undisclosed sale owned acquired board buys firm completes subject proposed letter takeover spokesman assets purchase completed disclosed sells deal management definitive industries sold international control owns tendered intent bought subsidiary acquires offered total led principle received held holdings plc systems price talks shareholder makes airlines gencorp merge capital division made proposal blah holds financial interest part allegheny directors plans seek chairman business preferred twa filing

Fréquence

said reuter march dlrs company 's corp mln pct shares new share stock group acquisition offer common unit stake york buy april agreement acquire cash agreed shareholders exchange terms sell board sale merger securities outstanding two purchase investment subsidiary acquired price year commission international total bid assets completed management

FIG. 7.5 - *Les meilleures unités pour la catégorie* acq. SCAR *sélectionne 293 unités. Nous ne montrons ici que les 100 premières.*

7.4.2.2 De l'adéquation de la généralisation : les enseignements de Scar

Les cent unités que nous avons présentées en résultat de la réduction effectuée par Scar pour la catégorie *acq* ne correspondent qu'à une partie du langage des instances réduit. Quelles propriétés des dépêches étiquetées par *acq* conduisent Scar à retenir autant d'unités ? Le nombre de dépêches possédant la catégorie ne semble pas être le motif. En effet, nous n'observons pas un aussi grand nombre d'attributs lors de la réduction du langage des instances localement à la catégorie la plus fréquente, *earn*.

Scar réduit le langage des instances en employant la généralisation comme moteur de recherche. Les attributs apparaissant dans la généralisation d'un exemple sont conservés pour composer le langage réduit. Scar cherche une généralisation pour chaque exemple qui n'est pas déjà couvert par une autre, dans l'espace de généralisation engendré par l'exemple.

Un vocabulaire commun pour décrire les textes est un prérequis à la généralisation. En comparant le nombre d'unités distinctes constituant le vocabulaire des dépêches étiquetés par les catégories *acq* et *earn* (cf. table 7.4), nous observons que le vocabulaire employé pour décrire les dépêches relevant de *acq* est plus varié : malgré un nombre inférieur de dépêches, le nombre d'unités apparaissant dans les dépêches étiquetées par *acq* est plus élevé. Il est donc a priori plus difficile de décrire de façon succincte le concept «possède la catégorie *acq*» que le concept «possède la catégorie *earn*». Ceci est confirmé par le nombre d'exemples étudiés par les généralisations locales successives : alors que Scar étudie 490 exemples (28%) pour la catégorie *acq*, il n'en étudie que 255 (7%) pour la catégorie *earn*. La catégorie *acq* n'est pas un cas isolé : on retrouve, par exemple, un comportement similaire pour les catégories *dlr* (dollar), money-fx (cours monétaires), interest (taux d'intérêts), *crude* (pétrole brut) ou *trade* (commerce). Les tables 7.4 et 7.5 résument, pour quelques catégories, le nombre d'exemples positifs, la taille du vocabulaire original, le nombre d'unités retenues après réduction et le nombre d'exemples étudiés par Scar lors de la recherche d'une généralisation.

La diversité du vocabulaire employé pour décrire les textes appartenant à une catégorie intervient lors de la réduction du langage des instances. Elle apparaît aussi, à moindre échelle, lors de l'apprentissage. En effet, le langage des instances réduit est aussi diversifié, puisque les sélections effectuées par Im et Relief+ favorisent aussi les termes apparaissant dans peu de dépêches.

7.4.3 Comment la réduction se comporte-t-elle lorsque deux catégories se recouvrent ?

L'apparition simultanée de deux catégories est chose fréquente sur le corpus Reuters. Nous avons choisi deux catégories, *gold* et *silver*, pour conclure notre analyse. La catégorie *gold* est affectée à 94 dépêches, tandis que la catégorie *silver*, moins fréquente, apparaît dans 21 dépêches, dont 13 possèdent aussi *gold*.

Du fait de la binarisation de l'apprentissage et de la réduction (c'est-à-dire une réduction locale), un tel cas ne pose pas de problème. Nous remarquons figures 7.6 et 7.7 que les

Catégorie	Nombre d'exemples	Nombre d'unités initiales	Nombre d'unités retenues	Nombre d'exemples étudiés	Longueur moyenne des dépêches
earn	**2877**	**8824**	**160**	**215**	**33.6**
dlr	131	2479	81	48	70.5
money-fx	538	5379	182	177	72.7
rubber	37	1541	14	12	94.0
interest	347	3857	140	116	60.6
gnp	103	2885	43	30	105.1
palm-oil	29	882	21	12	58.0
oilseed	122	2668	67	46	67.4
veg-oil	87	2055	36	30	65.5
acq	**1651**	**9393**	**243**	**432**	**51.1**
cocoa	56	1896	12	11	91.0
ship	198	3797	81	65	71.1
iron-steel	40	1464	29	19	79.9
wheat	212	3466	81	62	65.2
grain	433	5166	93	108	65.7
coffee	113	2888	34	28	92.3
sugar	126	3051	33	27	76.3
gold	94	1874	27	24	62.2
silver	21	748	15	10	63.9

TAB. 7.4 - *Tableau récapitulatif du nombre d'exemples, de la taille du vocabulaire initial et de la taille du vocabulaire réduit (première partie)*

meal-feed	30	936	23	15	62.2
gas	37	1214	20	10	65.1
bop	75	1837	40	25	75.4
barley	36	708	22	13	43.6
corn	176	3171	54	48	66.6
cpi	69	1659	26	21	62.3
livestock	73	2115	41	25	69.6
soybean	78	2227	44	35	78.6
copper	47	1307	25	19	57.7
cotton	39	1449	19	14	73.0
sorghum	23	871	15	10	72.3
yen	45	1623	39	17	90.8
carcass	50	1686	31	21	
crude	388	5706	120	117	82.7
nat-gas	75	2349	52	34	84.0
trade	369	5627	164	145	95.1
rice	35	1331	22	14	74.0
jobs	46	1227	23	13	67.9
retail	23	945	26	14	78.7
ipi	42	1141	21	15	66.5
alum	35	1090	14	13	60.9
zinc	21	688	14	9	54.5
reserves	54	1204	33	19	60.3
money-supply	139	1764	44	35	48.4
pet-chem	20	856	20	12	66.1

TAB. 7.5 - *Tableau récapitulatif du nombre d'exemples, de la taille du vocabulaire initial et de la taille du vocabulaire réduit (deuxième partie)*

IM :

gold ounces ounce mine ton ore silver grading mining mines tons short underground bullion feasibility depth assays property vancouver feet deposit coin coins drilling zone vein proven quebec grams reserves containing said assayed mineworkers holes resources drill cts production grade mined vs platinum hole mill net assay intersected intersections width

RELIEF+ (30) :

gold mine ounces ounce ore ton short silver grading mining tons south mines deposit feasibility price study coins drilling underground zone owned proven three bullion coin containing project quebec assays

SCAR (36) :

gold ounces ore mine ton short reserves owned ounce silver underground grade deposit coin south bullion average mining found mineworkers zone british containing assays estimated potential production north troy mined depth positive physical deposits begin operation

Fréquence :

gold reuter said march ounces 's mine pct company ton ounce year mln new ore dlrs mining production short silver corp april reserves mines two three price resources tons expected estimated property south average deposit years feet grading toronto british drilling owned vancouver added results start tonnes depth study underground

FIG. 7.6 - *Les meilleures unités pour la catégorie gold.*

IM :

silver gold ounces mine ore zinc tons ounce ton short gold/silver mill grading troy underground copper mining assaying carbonate drill-indicated higher,it muskeg outpace vein precious production idaho 30th admiralty coeur construct d'alene gold-silver-lead-zinc greens hecla hl feasibility drilling amselco firmness minted northwestern patricia supply/demand lead benefiting adjacent montagu mines study creek infrastructure

RELIEF+ (31) :

silver ounces mine short estimated gold necessary tons corp exploration five ore ounce preliminary production results south start strike ton vein zinc completion consolidation construct containing contain continue continuing covers d'alene

SCAR (23) :

silver ore ounces mine ton zinc gold troy drilling gold/silver east minimum mining minted anniversary peru 30th increasing underground copper columbia coin mines

Fréquence :

reuter silver said march gold mine ounces mln 's april dlrs ore production short year company pct tons ounce ton zinc expected mining rate annual continue copper estimated five lead mill new period british corp dlr drilling east grading likely mines necessary reserves study troy underground added additional agreed buy

FIG. 7.7 - *Les meilleures unités pour la catégorie silver.*

trois approches sélectionnent en premier les mêmes unités pour *silver* et *gold*. Du point de vue de la réduction locale à *silver*, ceci n'est pas étonnant puisque la majorité des dépêches ont les deux catégories. C'est plus surprenant pour la catégorie *gold*. Ceci suppose en effet que les dépêches possédant uniquement la catégorie *gold* emploient des termes semblables aux dépêches qui possèdent les deux catégories. Les langages des instances obtenus par Relief+ et Scar insistent plus sur le caractère discriminant des unités retenues.

Le problème lié aux classes à recouvrement apparaît alors lors de l'étiquetage d'une nouvelle dépêche. Le langage des instances décrivant *gold* permet difficilement de distinguer les dépêches étiquetées par le couple *gold* et *silver* des dépêches étiquetées uniquement par *gold*.

7.4.4 Conclusion

L'étude de la réduction effectuée pour ces quatre catégories nous a permis de mettre en évidence quelques propriétés des trois approches envisagées. Utiliser l'information mutuelle comme mesure d'évaluation permet d'obtenir un compromis entre fréquence et discrimination. La particularité de l'information mutuelle est de choisir les unités aussi bien en fonction de leur présence ou de leur absence, pour caractériser la présence ou l'absence d'une catégorie.

Relief+ favorise la discrimination par rapport à la fréquence, comme l'illustre la comparaison entre les unités tonnes et tax pour la catégorie *veg-oil*, ce qui est dû au présupposé d'absence par défaut.

Scar se distingue des deux approches précédentes par son aptitude à sélectionner des termes qui ne sont pas discriminants en eux-mêmes, mais qui le deviennent en conjonction avec d'autres. Par ailleurs, l'étude de la catégorie *acq* a montré les limites de la généralisation dans un cadre particulier, où les textes ne sont décrits que comme un ensemble d'unités présentes.

Le prochain chapitre est consacré à l'évaluation a posteriori de ces méthodes de réduction. Nous regarderons en particulier si les propriétés mises à jour sont confirmées lorsque la méthode de réduction est intégrée à un système de catégorisation.

Chapitre 8

Évaluation des approches de réduction

Nous avons décrit au chapitre précédent trois approches, Scar, Relief+ et Im, pour réduire le langage des instances. Nous avons de plus montré leurs propriétés en étudiant quelques catégories. Nous nous consacrons à présent à l'évaluation de la réduction dans le contexte général d'un système de catégorisation.

Rappelons que les méthodes Scar et Relief+ s'appuient sur l'hypothèse suivante : l'absence d'un mot est une donnée implicite dans la description des exemples et dans le langage des hypothèses. Ainsi, les règles de décision apprises ne contiennent que des descripteurs positifs, c'est-à-dire les descripteurs $\mathsf{X} = \mathrm{v}$, traduisant «le mot X est présent dans le document». L'apprenant utilisé en aval des trois algorithmes vérifie également cette contrainte.

L'objectif des expérimentations présentées dans ce chapitre est triple. Nous chercherons d'abord à évaluer les idées du modèle encapsulé lorsque l'on réduit le langage des instances dans un système de catégorisation. Une telle évaluation est rendue possible grâce aux différents présupposés qu'emploient les trois approches : forme des règles et cohérence des généralisations pour Scar, forme des règles uniquement pour Relief+ et aucune contrainte pour Im.

Au chapitre 7, nous avons montré comment ces différences conduisaient à des propriétés distinctes pour les trois approches. Nous utiliserons les résultats du système de catégorisation construit à l'aide d'un même apprenant pour confirmer ou infirmer ces propriétés. Ceci nous permettra de plus de mesurer l'influence de la réduction sur le comportement de l'apprenant.

Finalement nous mettrons en évidence l'intérêt d'une évaluation locale à chaque catégorie pour compléter l'évaluation globale. L'étude de certaines catégories nous conduira à proposer un regroupement des celles-ci en fonction de leur comportement commun relativement à l'apprenant et aux approches de réduction.

8.1 Cadre expérimental : variables et invariants

Les expériences que nous avons réalisées s'inscrivent dans un cadre commun, qui correspond à une instanciation du cadre proposé au chapitre 1. Nous avons figé différents composants de ce modèle, tandis que d'autres varient, comme l'illustre la figure 8.1.

La base de données textuelles que nous étudions est la collection Reuters-22 173. La génération des bases d'apprentissage et de test a été décrite à la section 1.3.

Pour affecter plusieurs catégories à un même texte, nous avons adopté l'approche

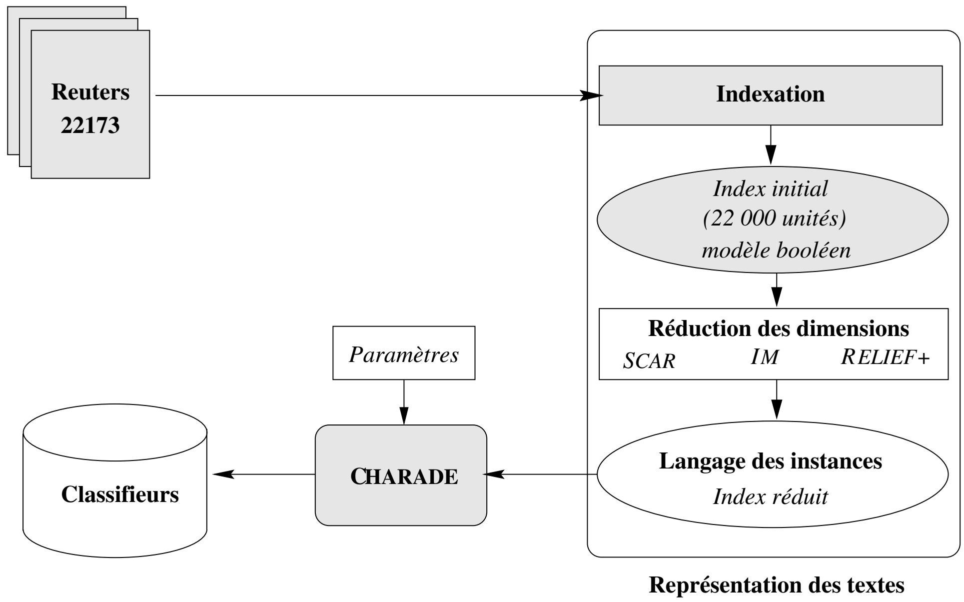


FIG. 8.1 - *Une instanciation particulière du cadre de la figure 1.1. Les parties grisées représentent les invariants dans les expériences décrites. L'index réduit varie en fonction de l'algorithme adopté pour la réduction.*

classique en catégorisation : on transforme le problème qui consiste à affecter n catégories en même temps en n sous-problèmes qui visent à affecter une unique catégorie. Cette transformation prend effet dès la représentation des textes (voir la section 7.2.2.1). Chaque concept «possède la catégorie X» est appris en opposant ce concept à tous les autres concepts «possède la catégorie Y». On cherche donc uniquement à caractériser les documents qui possèdent la catégorie X.

La transformation des textes en exemples d'apprentissage se décompose en deux étapes : l'indexation et la réduction du langage des instances. Nous avons choisi d'employer l'index initial construit par D. Lewis, disponible avec les données brutes (voir la section 1.3.3). La réduction du langage des instances représente la principale variable des expérimentations rapportées dans ce chapitre. Nous avons étudié les trois approches décrites au chapitre précédent, SCAR, RELIEF+ et IM, en faisant varier leurs différents paramètres, dont le nombre d'attributs.

L'apprenant est un invariant dans nos expérimentations. Il s'agit de CHARADEP, une version de CHARADE contrainte à engendrer des règles en employant uniquement des descripteurs positifs. Pour évaluer les propriétés des trois approches de réduction, nous avons fait varier les paramètres de CHARADEP, en particulier ceux contrôlant la redondance de l'apprentissage et la couverture minimale des règles engendrées.

Dans le but de comprendre le comportement des approches de réduction et d'apprentissage pour certaines catégories, nous introduisons COMP, une règle de classification

Réduction [1]	Paramètres	F_1	Gain (%)
IM–10	$\lambda_C = 1,\ f = 5\%$	0.481	-11,7
IM–10	$\lambda_C = 3,\ f = 0\%$	0.514	-5,7
IM–75	$\lambda_C = 2,\ f = 0\%$	0.732	+34,3
RELIEF+–10	$\lambda_C = 1,\ f = 5\%$	0.485	-11,0
RELIEF+–10	$\lambda_C = 3,\ f = 0\%$	0.547	+0,4
RELIEF+–100	$\lambda_C = 2,\ f = 0\%$	0.726	+33,2
SCAR–1–1	$\lambda_C = 1,\ f = 2\%$	0.417	-22,0
SCAR–1–1	$\lambda_C = 2,\ f = 0\%$	0.429	-21,3
SCAR–1–3	$\lambda_C = 2,\ f = 1\%$	0.747	+37,1
COMP	–	0.545	0

TAB. 8.1 - *Résumé des plus mauvais et des meilleurs résultats, évalués à l'aide du critère F_1 micro-moyenné, pour les trois approches de réduction.*

simple. Soit Y l'unité sélectionnée par RELIEF+ pour la catégorie X, COMP correspond à la règle *Si* $\mathsf{Y} = v$ *Alors* $X = v$. Cependant, l'unité Y peut conduire à un taux d'erreurs très fort pour les catégories peu fréquentes. Dans ce cas, la règle COMP devient $X = f$, c'est-à-dire que nous n'affectons jamais la catégorie X.

8.2 Résultats sur l'ensemble de la collection

Les résultats que nous présentons dans cette section ont pour objectif de mettre en évidence les facteurs importants lors de la réduction et d'estimer si les propriétés a priori données au chapitre 7 sont confirmées par le comportement du système construit.

Dans cette optique, nous nous intéressons tout d'abord aux performances des classifieurs appris. Nous étudierons ensuite la complexité expérimentale de l'apprentissage.

8.2.1 Des performances comparables pour les trois approches

Les résultats donnés à la table 8.1 montrent les plus mauvaises et les meilleures performances obtenues. La colonne Réduction indique l'approche de réduction et les paramètres retenus. La colonne Paramètres se rapporte aux paramètres de CHARADEP. Le gain est défini comme différence relative entre la performance observée et la performance réalisée par COMP. Ces résultats nous permettent de distinguer l'influence du langage des hypothèses et du langage des instances. En modifiant le critère d'évaluation, nous mesurons le poids relatif des différentes catégories dans le résultat global.

Lorsque la méthode de réduction ne sélectionne pas assez d'attributs, le langage des instances ne suffit pas à décrire les exemples et devient de plus inadéquat (cf. la section 4.2.1), car il ne permet pas de distinguer les exemples positifs des exemples négatifs.

1. Lorsque l'approche de réduction est RELIEF+ ou IM, nous indiquons le nombre d'attributs retenus lors de la sélection. Pour SCAR, nous mentionnons en premier paramètre la redondance, et en second la profondeur des treillis.

Réduction	Paramètres	mF_1	mf_1
IM–10	$\lambda_C = 1,\ f = 5\%$	0.443	0.213
IM–75 $*$	$\lambda_C = 2,\ f = 0\%$	0.525	0.332
RELIEF+–10	$\lambda_C = 1,\ f = 5\%$	0.427	0.212
RELIEF+–100	$\lambda_C = 2,\ f = 0\%$	0.500	0.308
SCAR–1–1	$\lambda_C = 1,\ f = 2\%$	0.349	0.097
SCAR–1–3	$\lambda_C = 2,\ f = 1\%$	0.575	0.331
COMP	–	0.573	0.320

TAB. 8.2 - *La mesure F_1 macro-moyennée traduit l'influence des catégories peu fréquentes et varie selon la valeur donnée a priori au coefficient indéfini 0/0.* SCAR*–1–3 est équivalent à* COMP*.*

Comme CHARADEP cherche à engendrer des règles cohérentes, la majorité des exemples n'est pas décrite par les règles apprises. La réduction effectuée par SCAR–1–1 est de loin la plus mauvaise ; c'est une conséquence des contraintes liées à la méthode. Il est difficile, en effet, de trouver des généralisations cohérentes, qui comprennent un seul descripteur [2]. Ainsi, pour 59 catégories, SCAR–1–1 ne retient aucun attribut, tandis que 102 catégories sont décrites par moins de 5 unités.

En utilisant le critère F_1 micro-moyenné, nous observons une amélioration franche des résultats par rapport à COMP, lorsque le nombre d'attributs croît [3]. Les trois approches ont des performances semblables, même si SCAR montre des résultats légèrement supérieurs.

Un changement du critère d'évaluation nous permet de mieux comprendre les différences entre les trois approches de réduction et les différents jeux de paramètres. La micro-moyenne favorise les catégories fréquentes, alors que la macro-moyenne donne un poids égal à toutes les catégories. Nous employons à présent le critère F_1 macro-moyenné, dont les valeurs varient selon la valeur donnée au rapport indéfini 0/0. À la table 8.2, nous notons le critère macro-moyenné mf_1 lorsque 0/0 vaut 0, et mF_1 lorsque 0/0 vaut 1. La différence entre mf_1 et mF_1 s'explique par la vingtaine de classes n'ayant pas d'exemples de test ou d'apprentissage. Les résultats de la table 8.2 sont en faveur de SCAR–1–3. Tout d'abord, par rapport à SCAR–1–1, cette approche conduit à une meilleure affectation de toutes les catégories. En comparaison avec RELIEF+–100 et IM–75, cette approche affecte mieux les catégories fréquentes, avec une performance similaire pour les catégories moins fréquentes, ce que montrent la faible différence entre les valeurs de mf_1 et une différence plus marquée pour celles de mF_1. En utilisant la macro-moyenne, COMP reste supérieur aux trois approches de réduction. Nous détaillerons cette caractéristique à la

2. Si de telles généralisations existaient, la catégorisation de textes serait une application ordinaire de la classification.

3. Notons que le nombre d'attributs augmente de façon explicite avec IM et RELIEF+, puisque c'est un paramètre de chacune de ces approches. Il augmente aussi avec SCAR lorsque la profondeur du treillis augmente, mais de façon non constante : par exemple, il y a plus d'attributs retenus pour une profondeur de 2 que pour une profondeur de 6, à redondance constante.

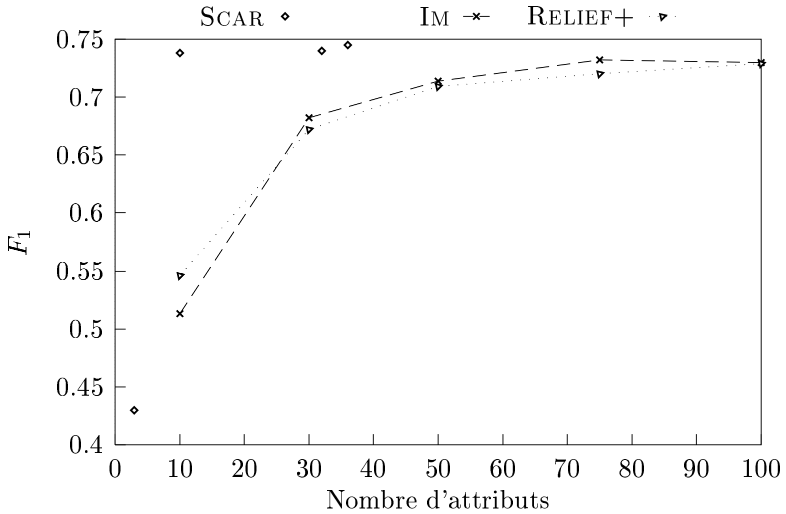


FIG. 8.2 - *Influence du nombre d'attributs sur les performances du système de catégorisation.*

section 8.3.

8.2.1.1 Influence des paramètres de réduction

L'un des points marquants est l'influence du langage des instances. Nous étudions à présent cet aspect de manière plus approfondie. La figure 8.2 montre l'évolution de la mesure F_1 en fonction du nombre d'attributs pour les trois approches. Le nombre d'attributs est un paramètre de IM et RELIEF+ [4], qui peut ne pas être atteint avec RELIEF+ en raison des deux autres paramètres de la méthode. Il n'existe pas de lien direct entre la réduction effectuée par SCAR et le nombre d'attributs sélectionnés. À titre indicatif, nous indiquons pour SCAR le nombre d'attributs correspondant à la moyenne, pour toutes les catégories, du nombre d'attributs par catégorie.

Comme nous l'avons observé à la table 8.1, l'augmentation du nombre d'attributs entraîne une amélioration des performances. La figure 8.2 apporte une information supplémentaire. En effet, à partir d'un certain nombre d'attributs, on atteint un seuil puis la mesure F_1 décroît. Pour IM, cette limite est située entre 75 et 100 attributs, tandis que pour RELIEF+, F_1 redescend à partir de 150 attributs. Tant que cette limite n'est pas atteinte, une augmentation d'attributs entraîne une hausse du rappel, mais une baisse de la précision. Au delà de cette limite, cette tendance diminue, puis s'inverse.

De manière surprenante, les paramètres de SCAR n'ont qu'une influence très limitée sur les performances du système. Nous observons en effet à la figure 8.2 qu'un seul point se distingue des autres lors de la réduction avec SCAR. Il s'agit de SCAR–1–1, cas déjà distingué précédemment et qui entraîne la création d'un langage des instances non perti-

4. Le nombre d'attributs maximal est toujours atteint pour IM, mais peut ne pas l'être pour RELIEF+ à cause des autres paramètres de la méthodes.

nent. Lorsque le paramètre de redondance [5] augmente, le rappel augmente également au détriment de la précision. Ces variations restent proportionnelles, dans la mesure où F_1 reste stable.

Différents travaux en catégorisation ont cherché à déterminer le nombre optimal d'attributs pour un apprenant et une collection donnés. Par exemple, dans [LR94], ce nombre est de 90 pour construire des arbres de décision et de 4 pour un apprenant Bayésien, tandis que [Wie95] évalue à 20 le nombre d'attributs optimal pour un réseau de neurones. Dans ces approches, la réduction est effectuée à l'aide de l'information mutuelle. Le nombre optimal que nous observons avec IM est plus élevé, conséquence de la représentation des hypothèses que nous avons choisie.

8.2.1.2 Confirmation des propriétés a priori des trois approches

Nous concluons notre discussion sur les performances comparées des trois approches de réduction par l'évaluation de leurs propriétés en fonction de l'apprenant. Nous avons donné au chapitre 7 des propriétés a priori de ces approches en étudiant les unités retenues. Nous utilisons à présent l'apprenant et le système de catégorisation pour estimer la validité de ces propriétés.

La figure 8.3 résume l'influence des paramètres de CHARADEP pour chacune des approches de réduction. Certaines caractéristiques de ces courbes sont relatives à l'apprenant, comme par exemple la diminution de F_1 lorsque le fractionnement [6] f augmente, ou encore l'augmentation de la redondance, qui entraîne une baisse de la précision.

La première observation que nous faisons est la similitude entre les courbes représentant SCAR et RELIEF+, et leur différence par rapport aux courbes caractérisant IM. Cette similitude est la conséquence d'une réduction effectuée uniquement en utilisant des informations sur la présence des unités. Nous avons en effet mené des expériences complémentaire en utilisant le rang défini dans [WPW95], à la place de l'information mutuelle. Ce critère, comme l'information mutuelle, prend en compte l'absence des unités dans les textes lors de la réduction. Les courbes obtenues sont semblables à celles obtenues pour l'information mutuelle.

La combinaison des trois courbes permet de confirmer les propriétés a priori des approches de réduction. Par exemple, SCAR entraîne un rappel plus élevé et une précision en moyenne équivalente. Les unités sélectionnées par SCAR permettent d'étiqueter correctement plus de documents, en ayant un taux d'erreurs en moyenne équivalent. Ces unités caractérisent bien les catégories mais entraînent plus d'erreurs que lorsque les unités sont sélectionnées par IM. Le comportement de RELIEF+ est similaire, tandis que IM entraîne un taux d'erreurs plus faible, mais permet de retrouver moins de documents, ce qui confirme que les unités retenues sont plus discriminantes et moins caractéristiques que celles retenues par les deux autres approches.

5. Rappelons que la redondance lors de la sélection a pour objectif de décrire un même exemple à l'aide de deux expressions différentes, afin de varier le vocabulaire sélectionné.

6. Le fractionnement représente la proportion minimale d'exemples d'une même classe pour entraîner la création d'une règle.

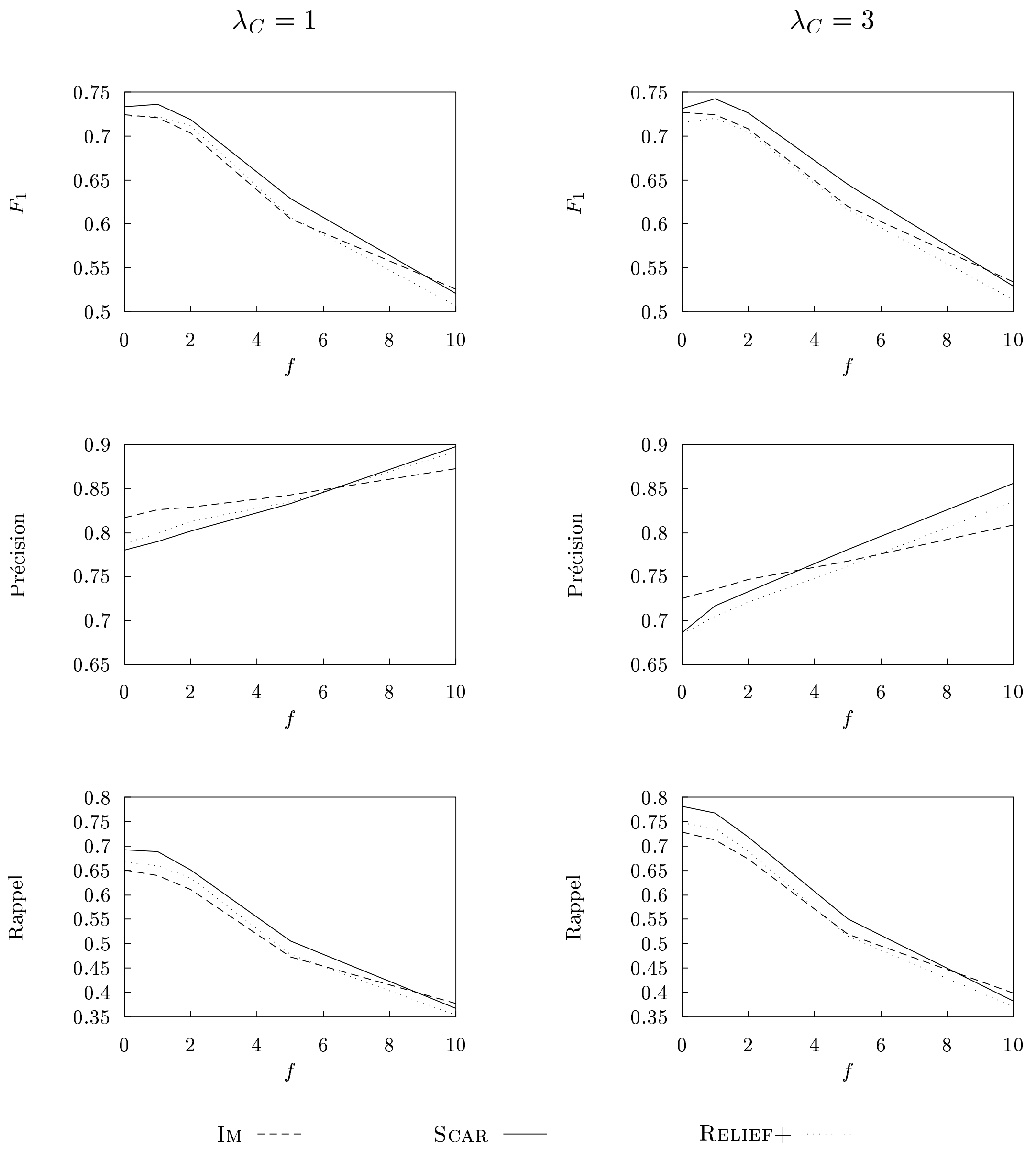


Fig. 8.3 - *Poids des paramètres λ_C et f pour chaque approche de réduction*

8.2.1.3 Discussion

Nous avons essayé de reproduire dans des approches par filtrage les idées sous-jacentes au modèle encapsulé proposé par Kohavi [JKP94]. SCAR s'appuie sur deux informations : la forme des généralisations recherchées, exprimées en termes de descripteurs positifs uniquement, et la cohérence des généralisations . RELIEF+ prend en compte uniquement la première information, tandis que IM les ignore.

On a montré la supériorité du modèle encapsulé pour des applications classiques en apprentissage [JKP94, AB95]. Les résultats que nous obtenons vont à l'encontre de cette supériorité lorsque l'ensemble d'attributs est au départ de taille importante : en effet IM obtient des performances comparables aux deux autres approches. Notons cependant que ces approches ne correspondent pas au modèle encapsulé mais essaient d'en capturer les caractéristiques.

8.2.2 Diminution de la complexité lorsque la réduction ignore l'absence des unités

En théorie, et dans le pire des cas, l'apprentissage avec CHARADE (et CHARADEP) est exponentiel[7] en fonction du nombre d'attributs. Pourtant la complexité expérimentale est moindre. Nous nous intéressons à présent à la complexité calculatoire, qui permet d'évaluer l'exploration effective de l'espace de généralisation. Nous étudions ensuite la qualité des classifieurs induits.

8.2.2.1 Complexité calculatoire

Nous définissons la complexité calculatoire comme le nombre de nœuds explorés lors du parcours de l'espace de généralisation. Nous avons choisi de ne pas prendre en compte le temps nécessaire à l'apprentissage car nous n'avons pas pu conserver des conditions expérimentales constantes.

La complexité expérimentale des trois approches est donnée à la table 8.3. La différence entre SCAR et RELIEF+ d'une part, et IM d'autre part est frappante. Ainsi, il apparaît clairement que réduire le langage des instances en prenant en compte la forme des généralisations recherchées, c'est-à-dire respectant l'absence par défaut, a une influence non négligeable sur la complexité calculatoire.

Le présupposé relatif à l'apprenant, *i.e.* la cohérence des hypothèses, qui est inclus dans SCAR, ne simplifie pas l'exploration de l'espace de généralisation. L'explication se trouve dans la différence principale entre SCAR et RELIEF+, présentée à la section 7.4. Alors que RELIEF+ favorise chaque unité isolément, SCAR les retient en groupe, lorsqu'elles forment une règle cohérente dans l'espace d'origine. En conséquence, le treillis engendré par le langage des instances sélectionné par RELIEF+ possède de nombreux nœuds auxquels aucun exemple n'est associé. De plus, le nombre d'attributs retenus joue

7. Si on limite le nombre de descripeurs à k par prémisse, la complexité réduit et devient polynomiale en n^{k+1}. Dans la pratique, nous avons choisi k grand (20) pour limiter le moins possible l'exploration.

Réduction	λ_C	Complexité	Réduction	λ_C	Complexité
IM–30	1	222 948	IM–75	1	446 242
–	3	247 995	–	3	493 689
RELIEF+–30	1	34 343	RELIEF+–75	1	94 267
–	3	43 651	–	3	119 323
SCAR–1–2	1	47 196	SCAR–1–3	1	236 404
–	3	64 761	–	3	277 860

TAB. 8.3 - *Complexité globale réelle de l'exploration par* CHARADEP *suivant les approches de réduction*

un rôle important. Si SCAR retient moins d'unités en moyenne, il retient plus de 80 attributs pour 9 catégories (voir les tables 7.4 et 7.5). C'est en partie à ces catégories qu'est due l'augmentation de la complexité.

8.2.2.2 Complexité et qualité des bases de règles

Cette propriété de RELIEF+, sélectionner les unités individuellement, est visible également pour la complexité des classifieurs, qui sont en l'occurrence des bases de règles. Nous mesurons la complexité d'une base de règles en fonction du nombre de règles induites. Cette complexité dépend du langage des instance, mais aussi des paramètres de CHARADEP, en particulier la redondance. Rappelons en effet que l'objectif de la redondance est de proposer plusieurs généralisations possibles pour un même exemple.

La complexité relative aux trois approches de réduction est donnée à la table 8.4, en fonction de la redondance autorisée lors de l'apprentissage. Observons tout d'abord que, comme on pouvait le prévoir, la redondance lors de l'apprentissage entraîne une augmentation de la complexité. L'exception que représente SCAR–1–1 a pour origine la même explication que ci-dessus. Le langage des instances réduit ne permet pas de créer des règles cohérentes, car il est à la fois de taille restreinte et inadéquat. Les classifieurs engendrés après réduction avec chacune des méthodes ont des complexités comparables. Deux points attirent cependant l'attention. Tout d'abord, lorsque le nombre d'attributs augmente, le nombre de règles engendrées lorsque la réduction est réalisée par RELIEF+ est inférieur à ceux obtenus avec les autres approches de réduction. C'est une conséquence de la sélection individuelle des unités. Lorsque peu d'attributs sont retenus, le fait qu'IM engendre le moins de règles est une conséquence de la prise en compte de l'absence lors de la réduction et non lors de l'apprentissage.

Le nombre de règles engendrées reflète-t-il la qualité de la base de règles ? Pour évaluer la qualité d'une base de règles, nous considérons le taux d'utilisation de ces règles sur l'ensemble de test. Une évaluation globale de la qualité des règles est limitative. C'est pourquoi nous distinguons différentes catégories. En devançant la section 8.3, la table 8.5 montre, pour quelques catégories, le nombre de règles et le taux d'utilisation associé, relativement à chaque approche de réduction. Remarquons tout d'abord que la qualité des classifieurs engendrés par le couple SCAR – CHARADEP est supérieure à celle des autres

Réduction	λ_C	Nb. règles	Réduction	λ_C	Nb. règles
IM–30	1	1180	IM–75	1	1872
IM–30	3	2954	IM–75	3	5335
RELIEF+–30	1	1225	RELIEF+–75	1	1884
RELIEF+–30	3	3095	RELIEF+–75	3	5254
SCAR–1–1	1	211	SCAR–1–3	1	2096
SCAR–1–1	3	214	SCAR–1–3	3	5542

TAB. 8.4 - *Taille de la base de règles totale pour les trois approches, selon la redondance de l'apprentissage (λ_C)*

classifieurs, que ce soit globalement (la ligne Total) ou par catégorie. La supériorité par rapport à IM – CHARADEP s'explique par la qualité des classifieurs induits pour les catégories les plus fréquentes (*earn* (bénéfices des entreprises/rapports prévisionnels), *acq* (fusions/achats d'entreprises), *money-fx* (cours monétaires) et *grain* (céréales). Ces classifieurs jouent un rôle déterminant dans la base de règles globale, puisqu'ils représentent presqu'un tiers des règles. Notre seconde observation est liée au rapport entre règles induites et exemples d'apprentissage. À l'exception de la catégorie *earn*, le rapport entre règles induites et nombre d'exemples d'apprentissage est supérieur à un quart. C'est en partie dû à la redondance lors de l'apprentissage, mais aussi au présupposé de l'apprenant. En effet, lorsqu'une règle possède un contre-exemple, CHARADEP spécialise cette règle, diminuant ainsi sa couverture.

8.2.2.3 Discussion

Nous concluons cette étude de la complexité en rappelant les points marquants que nous avons mis en évidence. La complexité calculatoire donne l'avantage aux deux approches qui, lors de la réduction, prennent en compte la forme des hypothèses engendrées par l'apprenant. Les classifieurs engendrés ont une complexité comparable dans tous les cas. Ce point tend à confirmer les idées du modèle encapsulé, puisque l'emploi de connaissances sur l'apprenant permet de réduire la complexité de l'apprentissage à performances égales.

Le deuxième point marquant repose sur la qualité des classifieurs et conduit à deux remarques. Tout d'abord, les résultats obtenus confirment également les idées sous-jacentes au modèle encapsulé. En effet, les classifieurs sont en moyenne de meilleure qualité lorsque la réduction est réalisée par SCAR, qui prend en compte les caractéristiques de l'apprenant. La seconde remarque concerne CHARADEP. Cet algorithme éprouve des difficultés à généraliser les exemples en créant des bases de règles compactes, comme l'ont montré les résultats relatifs aux catégories de la table 8.5, à l'exception de *earn*. Ainsi, le comportement de CHARADEP est plus proche d'une mémorisation de descriptions d'exemples que d'une véritable généralisation, caractéristique renforcée par la redondance.

Réduction	Catégorie (Nb. exemples)	Nb. règles	Taux réel (%)	Taux correct(%)
SCAR–1–3	*earn* (2877)	198	97,5	96,5
	acq (1651)	411	98,8	96,1
	money-fx (538)	233	95,3	87,1
	grain (433)	150	96,7	89,3
	crude(388)	179	93,8	82,1
	dlr(131)	79	67,1	32,9
	interest (347)	163	77,5	54,6
	silver (21)	12	83,3	41,7
	Total	3151	79,2	62,2
IM–75	*earn*	302	79,8	56,0
	acq	573	92,0	88,6
	money-fx	194	89,1	70,6
	grain	156	92,9	80,1
	crude	148	93,9	85,1
	dlr	81	71,6	45,7
	interest	132	75,8	59,8
	silver	18	38,9	11,1
	Total	3613	74,0	54,6
RELIEF+–100	*earn*	177	57,1	56,5
	acq	336	99,1	97,0
	money-fx	217	82,5	75,6
	grain	152	94,1	85,5
	crude	175	85,1	76,0
	dlr	79	56,9	35,4
	interest	137	65,7	50,4
	silver	14	28,6	7,1
	Total	3163	67,8	52,4

TAB. 8.5 - *Comparaison entre le taux d'utilisation et le taux d'utilisation correcte pour quelques catégories. Le nombre d'exemples indiqué correspond à la base d'apprentissage. Dans tous les cas, la redondance est fixée à 2.*

8.2.3 Quel est le coût inhérent à la réduction?

Nous avons montré à la figure 7.3 un exemple illustrant une ambiguïté introduite par la réduction. Nous cherchons à présent à évaluer le coût de la réduction. En termes de complexité, peu d'apprenants symboliques, à l'exception de l'algorithme RIPPER [CS96], sont capables de traiter des exemples décrits à l'aide de milliers d'attributs. En conséquent, la réduction diminue nécessairement la complexité de l'apprentissage, le rendant tout simplement possible.

Les résultats présentés dans [CS96] auraient pu nous permettre d'estimer le coût de la réduction. Cependant, nous n'avons pas retenu cette évaluation pour deux raisons : tout d'abord parce que les expériences décrites par W. Cohen n'ont pas été réalisées sur la même base d'apprentissage, ensuite parce que RIPPER est un apprenant plus sophistiqué que CHARADEP et repose sur une exploration de type glouton et un élagage de la base de règles.

Pour effectuer cette évaluation, nous avons préféré SCAR. En effet, si nous conservons les règles cohérentes qui servent à retenir des unités, nous pouvons considérer que ces règles forment des classifieurs. Nous n'avons pas cherché à optimiser le résultat obtenu par SCAR. Le meilleur paramétrage,SCAR–2–6, permet d'obtenir un F_1 égal à 76,5%. Pour SCAR–1–3, le critère vaut 75%, ce qui est très proche du résultat obtenu par CHARADEP. De plus, dans la majorité des cas, le couple SCAR – CHARADEP obtient une performance comparable à l'emploi de SCAR seul.

Pourquoi coupler SCAR avec CHARADEP, si les résultats obtenus par SCAR seul sont équivalents? SCAR est un algorithme simple, qui ne peut engendrer qu'un type de généralisation, de la forme *Si* $\mathsf{U} = v$ *Alors* $C = v$. Si nous voulons faire évoluer le langage des hypothèses, SCAR seul n'effectuera pas la tâche qui lui est dévolue. C'est pourquoi nous avons estimé le coût de la réduction dans ce cadre simple.

En somme, les résultats obtenus montrent que la réduction a un impact assez faible sur les performances du système de catégorisation et favorise l'application d'algorithmes d'apprentissage classiques. Des expériences supplémentaires avec RIPPER, avec et sans sélection, permettraient de vérifier la généralité de ces résultats.

8.3 Un point de vue local : émergence de deux groupes

Nous nous intéressons à présent aux classifieurs relatifs à chaque catégorie, dans l'optique de faire apparaître des propriétés communes à certaines catégories. Ces propriétés peuvent être relatives aux méthodes de réduction ou à l'apprenant. Pour cela, nous introduisons dans notre comparaison la ligne de comparaison, notée COMP. C'est un classifieur simple, réduit à une règle possédant une seule prémisse. Holte a montré que de tels classifieurs se comportaient bien sur la majorité des bases d'évaluation classiques en apprentissage [Hol93].

Notre étude nous a permis de mettre en évidence deux groupes particuliers de catégories et de souligner une caractéristique de la catégorisation pour l'apprentissage. Ce sont ces aspects que nous exposons à présent.

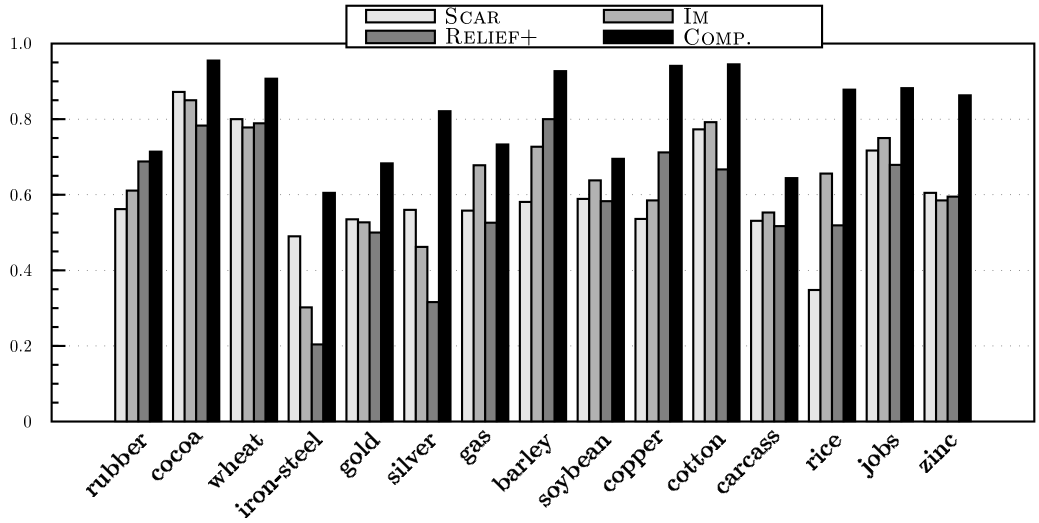


FIG. 8.4 - *Un classifieur simple (*COMP*) obtient de meilleurs résultats qu'un classifieur plus complexe.*

PATTERN-ID 7429
TOPICS: *cocoa acq* **END-TOPICS**
W.R. GRACE <GRA>, BERISFORD PLAN COCOA VENTURE
NEW YORK, June 2 - W.R. Grace and Co said it has agreed to combine its $\boxed{\text{cocoa}}$ processing businesses with those of S. and W. Berisford PLC.
It said the joint venture, to be 68.4 pct owned by Grace and 31.6 pct by Berisford, would have annual sales in 1987 of over 700 mln dlrs.
Grace said the transaction involves the combination of its $\boxed{\text{cocoa}}$ products division and two Berisford $\boxed{\text{cocoa}}$ processing units, which would be operated under Grace management.
The company said Berisford would contribute its Dutch and West German cocoa subsidiaries and issue new ordinary shares to Grace in connection with the transaction. It said closing is expected by early fall, subject to regulatory approvals.
Reuter

FIG. 8.5 - *La catégorie* cocoa *n'est pas retrouvée pour cette dépêche*

8.3.1 Un classifieur simple est plus performant sur des domaines restreints

Pour un certain nombre de catégories, nous observons que COMP est de loin le classifieur le plus performant (voir la figure 8.4). Pourquoi en est-il ainsi ? Ces catégories possèdent-elles des caractéristiques communes ?

Prenons l'exemple de la dépêche 7429 (cf. figure 8.5). L'affectation de la catégorie *cocoa* (cacao) semble évidente. Pourtant aucune des trois approches utilisant CHARADEP n'est capable de la trouver, alors que la simple règle *Si* cocoa = *v Alors cocoa* = *v*, qui créée par COMP, le permet. CHARADEP ne parvient pas à engendrer cette règle car elle n'est pas cohérente avec les exemples d'apprentissage. En fait, la majorité des règles apprises par CHARADEP spécialisent la règle *Si* cocoa = *v Alors cocoa* = *v*, comme le montre la table 8.6.

(defrule I48-R6 "8"	(defrule I48-R7 "19"
(cocoa == true)	(cocoa == true)
(quality == true)	(tonnes == true)
=>	=>
(assert (*cocoa* == true)))	(assert (*cocoa* == true)))
(defrule I48-R8 "15"	(defrule I48-R9 "32"
(cocoa == true)	(cocoa == true)
(member == true)	(rules == true)
=>	=>
(assert (*cocoa* == true)))	(assert (*cocoa* == true)))

TAB. 8.6 - *Règles apprises pour affecter la catégorie* cocoa*. Le nombre "X" indique le nombre d'exemples couverts.*

	cocoa	
cocoa	56	6
	2	–

	gold	
gold	94	79
	0	–

	silver	
silver	20	29
	1	–

	wheat	
wheat	205	38
	7	–

TAB. 8.7 - *Les unités sont plus caractéristiques qu'elles ne sont discriminantes, comme le montrent les tables de contingence entre catégorie (les colonnes) et unité (les lignes).*

Cette incapacité à engendrer une règle simple, en raison de l'hypothèse de cohérence, se retrouve pour les autres catégories de la figure 8.4. En effet l'unité u, sélectionnée par RELIEF+ pour la ligne de comparaison, est caractéristique de la catégorie *X*, c'est-à-dire qu'il existe peu de dépêches possèdant la catégorie *X* sans possèder l'unité u. Mais cette unité n'est pas forcément discriminante, puisqu'elle est aussi présente dans des dépêches non étiquetées par *X* (voir la table 8.7). Notons que l'unité retenue correspond en général à la catégorie, par exemple le terme cocoa pour la catégorie *cocoa*, ou bien l'unité gasoline pour la catégorie *gas* (essence).

Pour la catégorie *rice* (riz), il est nécessaire de prendre en compte un autre facteur. Pour réduire le langage des instances, SCAR recherche des règles cohérentes, comme le fait CHARADEP pour apprendre. Lorsqu'il existe peu de généralisations cohérentes pour les exemples d'une catégorie, la pertinence des attributs retenus diminue. Ainsi, même si le terme rice est retenu, les autres unités sélectionnées ne sont pas assez générales pour permettre une bonne reconnaissance de la catégorie.

Nous terminons l'étude de ce groupe par les catégories à recouvrement *silver* (argent) et *gold* (or), introduites au chapitre précédent. Alors que les simples termes gold et silver suffisent à étiqueter correctement un grand nombre de documents, CHARADEP n'y parvient pas. En particulier, l'apprentissage sur la catégorie *silver* est problématique puisque l'on engendre presqu'autant de règles qu'il y a d'exemples. Pourtant, ces règles sont non seulement inutiles sur la base de test, mais de plus la source d'erreurs d'étiquetage. La catégorie *gold* est mieux reconnue par CHARADEP. Les erreurs commises pour la catégorie *gold* le sont sur des catégories sémantiquement proches, par exemple *silver* ou *reserves*

TOPICS: *rubber* **END-TOPICS**
TOKYO EXCHANGE ALLOWS MORE GOLD AND RUBBER BROKERS
TOKYO, June 1 - The Tokyo Commodity Exchange for Industry (TOCOM) said it will allow five more members to become precious metal brokers, taking the total to 54, and four more members to broke rubber, for a total of 39.
An exchange official told Reuters the Ministry of International Trade and Industry is expected to approve the the new brokers by mid-June.
The move has been under study since early May in response to requests by non-broker members of the exchange.
REUTER

FIG. 8.6 - *Erreur humaine ou erreur d'apprentissage ?*

(réserves). Ces confusions montrent deux aspect du mot gold, un métal précieux, dans le premier cas, et une monnaie d'échange dans le second. Notons cependant une exception, la dépêche de la figure 8.6, où *gold* est confondue avec *rubber* (caoutchouc).

Nous pouvons résumer les propriétés communes à ces catégories de la manière suivante : ce sont des catégories de fréquence moyenne, qui traitent d'un domaine restreint en des termes précis. La connaissance de quelques termes caractéristiques permet d'affecter sans trop d'erreurs ces catégories, puisque leur vocabulaire est spécifique. De plus, dans ce contexte, engendrer un classifieur qui crée des généralisations cohérentes mène en général à un sur-apprentissage.

8.3.2 Apport de l'apprentissage lorsque le sujet est abstrait

La figure 8.7 montre un deuxième groupe de catégories, dont le comportement est inversé par rapport au groupe de la section précédente. En effet, COMP obtient des résultats médiocres par rapport aux classifieurs induits par CHARADEP, et ce quelle que soit la méthode de réduction employée.

COMP obtient des performances médiocres pour deux raisons, non exclusives : le classifieur entraîne de nombreuses erreurs, ou bien il ne permet pas de retrouver les documents étiquetés. Prenons l'exemple de la catégorie *dlr* (dollar). L'unité rates retenue permet de ne retrouver que 31 dépêches sur 96, mais est à l'origine de 233 erreurs d'étiquetage.

L'apprentissage permet de reconnaître ces catégories de manière plus efficace, indépendamment de la réduction semble-t-il. Nous observons en effet le même comportement, qui se résume à trois actions :

- diminuer les oublis tout en diminuant les erreurs,
- diminuer les oublis, même si les erreurs augmentent simultanément,
- faire moins d'erreurs au détriment du nombre de textes retrouvés.

Le premier cas est illustré par la catégorie *dlr*. En effet, les trois approches de réduction, couplées à CHARADEP, permettent de diminuer le nombre d'erreurs, environ 50, tout en retrouvant une dizaine de documents en plus. Observons que cette catégorie est affectée

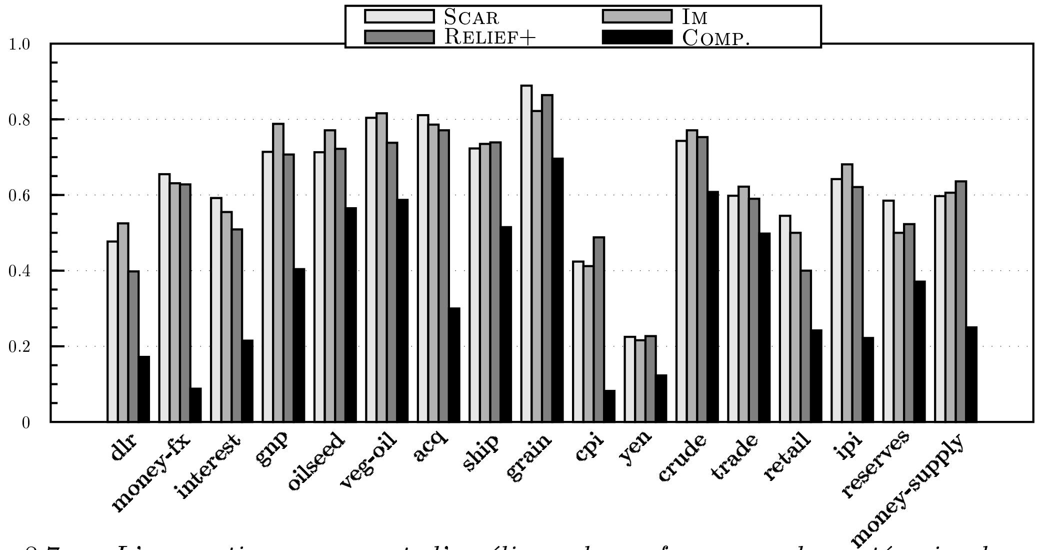


FIG. 8.7 - *L'apprentissage permet d'améliorer la performance des catégories les plus abstraites par rapport à* COMP.

à des textes de longueur et de forme variables, et qu'elle est rarement affectée comme étiquette unique.

Les catégories *trade* (commerce) et *acq* sont des exemples du deuxième cas. La différence entre COMP et les autres approches est moins frappante pour la catégorie *trade*, ce qui est la conséquence d'une précision assez faible : si l'on reconnaît mieux cette catégorie, c'est au détriment des erreurs effectuées. En particulier, lorsque les dépêches sont longues, les classifieurs affectent la catégorie *trade* à la place ou en complément de *dlr* ou de *money-fx*, qui sont des catégories sémantiquement proches. La catégorie *acq* se distingue de ce groupe par les erreurs commises. En effet, les erreurs sont en général faites sur des textes traitant d'un sujet proche, alors que pour *acq*, on commet peu d'erreurs mais dans des domaines très variés (cf. la table 8.8).

Le dernier cas est plus rare et correspond à des catégories moins fréquentes. On retrouve moins de textes, mais la différence avec COMP est inférieure à 10 textes. Par contre, la diminution des erreurs est beaucoup plus impressionnante. Illustrons notre propos avec la catégorie *yen* (yen) : le terme japanese permet de retrouver 13 textes sur 34 mais engendre 164 erreurs, en majorité commises sur des textes traitant de *dlr*, *money-fx*, *interest* (taux d'intérêts) et *trade*. La confusion entre *yen* et *trade* n'est pas étonnante, lorsqu'on observe que *trade* étiquette de nombreux exemples traitant de relations commerciales entre États-Unis et Japon. Les classifieurs induits avec CHARADEP pour la catégorie *yen* engendrent 50 erreurs, avec un nombre de déductions constant.

Pour conclure, remarquons une propriété commune à la plupart des catégories de ce groupe. Elles font partie des catégories pour lesquelles SCAR a du mal à réduire le langage des instances (voir la table 7.4). Cette difficulté est liée à la taille du vocabulaire, qui dépend lui-même de l'abstraction des sujets traités dans les documents. Dans ce cas, la cohérence des hypothèses ne semble pas être un handicap à la construction de classifieurs, puisque, de toute manière, chaque règle ne décrit qu'un petit nombre d'exemples.

acq	*earn* (33) *crude* (14) *money-fx* (12) *nat-gas* (12) *gold* (6) *copper* (6) *trade* (6) *iron-steel* (5) *grain* (4) *livestock* (4) *dlr* (3) *alum* (3) *interest* (2) *wheat* (2) *coffee* (2) *corn* (2) *yen* (2) *carcass* (2) *rubber* (1) *gnp* (1) *palm-oil* (1) *oilseed* (1) *veg-oil* (1) *tin* (1) *sugar* (1) *gas* (1) *soybean* (1) *stg* (1) *strategic-metal* (1) *platinum* (1) *zinc* (1)
earn	*acq* (50) *crude* (11) *trade* (10) *nat-gas* (6) *grain* (5) *copper* (5) *gas* (4) *corn* (4) *gnp* (3) *oilseed* (3) *meal-feed* (3) *soybean* (3) *sorghum* (3) *rice* (3) *money-supply* (3) *soy-meal* (3) *money-fx* (2) *veg-oil* (2) *iron-steel* (2) *coffee* (2) *gold* (2) *barley* (2) *cpi* (2) *livestock* (2) *lead* (2) *cotton* (2) *pet-chem* (2) *soy-oil* (2) *oat* (2) *dlr* (1) *ship* (1) *wheat* (1) *groundnut* (1) *carcass* (1) *retail* (1) *ipi* (1) *alum* (1) *nickel* (1) *cotton-oil* (1)

TAB. 8.8 - *Liste des catégories étiquetant une dépêche attribuée à tort aux catégories acq et earn par les classifieurs induits par* CHARADEP, *après réduction avec* SCAR. *Les nombres indiqués entre parenthèses représentent le nombre de fois où acq ou earn ont été affectées à tort.*

8.3.3 Un groupe non homogène

Nous regroupons à présent toutes les catégories qui n'entrent pas dans les deux groupes des sections précédentes, mais qui ont une fréquence moyenne dans la collection. Les propriétés de ces catégories sont très variées. Certaines se comportent comme les catégories du premier groupe, où une règle simple suffit à obtenir une catégorisation efficace. Ce sont par exemple les catégories *coffee* (café) et *sugar* (sucre).

Nous nous intéressons ici à la catégorie *earn*, car elle est la seule à représenter un cas d'apprentissage classique, puisqu'un tiers de la base d'apprentissage possède cette catégorie. Les résultats obtenus par les différentes approches sont comparables et se caractérisent par une valeur de F_1 comprise entre 90% et 95%. Cependant, de manière analogue à la catégorie *acq*, les erreurs commises lors de l'affectation de cette catégorie sont relatives à des documents traitant de domaines variés (voir la table 8.8). Nous proposons la raison suivante pour expliquer cette similarité. Ces deux catégories étiquettent des documents généraux, qui s'intéressent aux entreprises. Une partie importante du vocabulaire relatif à ces deux catégories est constituée de noms propres, et les termes employés par ailleurs sont généraux, contrairement aux autres catégories qui possèdent un vocabulaire assez spécifique. Cette explication ne nous semble que partielle, et doit être mise en relation avec le fait que *earn* ou *acq* sont rarement affectées en conjonction avec d'autres catégories.

8.3.4 Forte variation pour les catégories peu fréquentes

Les catégories *earn* et *acq* sont très fréquentes et possèdent un poids important dans l'évaluation globale. Ce sont cependant des cas marginaux de la collection Reuters, où les catégories sont plutôt caractérisées par une fréquence faible. Si nous observons à présent les catégories ayant peu d'exemples d'apprentissage ou de test[8], nous ne pouvons que

8. Nous considérons arbitrairement que ces catégories ont moins de 20 exemples d'apprentissage.

Méthode	*dmk*			*fuel*			*sunseed*		
	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
SCAR–1–3	0	0	5	1	7	15	2	3	5
RELIEF+–75	0	0	5	2	5	14	2	3	5
IM–75	0	4	5	0	8	16	3	0	4
COMP	0	0	5	13	25	3	4	45	3

TAB. 8.9 - *Déductions effectuées par les quatre approches pour les catégories dmk, fuel et sunseed. La colonne (a) représente le nombre de déductions correctes, la colonne (b) le nombre d'erreurs et la colonne (c) les oublis.*

constater l'absence d'une même tendance parmi ces catégories.

Certaines favorisent COMP tandis que d'autres avantagent l'une des approches de réduction. Toutefois, il est nécessaire de noter la forte variabilité du critère F_1. Une différence de quelques textes entre deux approches peut se traduire par une variation de plus de 20% pour ce critère.

Prenons l'exemple des trois catégories *dmk* (mark allemand), *fuel* (combustible/carburant) et *sunseed* (graines de tournesol). Les résultats sont présentés à la table 8.9. Considérons tout d'abord la catégorie *dmk*. Seule la réduction effectuée par IM se démarque. En fait, IM retient 75 unités alors que RELIEF+ et SCAR n'en retiennent qu'une dizaine. Pourtant, les unités retenues par IM ne permettent pas de retrouver les textes[9] mais entraînent des erreurs (principalement une confusion avec *dlr* et *money-fx*).

Étudions à présent la catégorie *fuel*. SCAR et RELIEF+ ne diffèrent que pour trois documents (deux erreurs et un oubli), pourtant le critère F_1 est doublé pour RELIEF+ (17,4% par rapport à 8,3%). Malgré les nombreuses erreurs commises par COMP, le classifieur appris, *Si* fuel $=$ *v Alors fuel* $=$ *v*, est celui qui obtient le meilleur résultat. Finalement, considérons le cas de IM : CHARADEP engendre une dizaine de règles qui sont plus spécifiques que la règle précédente, car le terme fuel ne permet pas d'engendrer une règle cohérente. On retrouve ici la caractéristique du premier groupe, où la cohérence mène à un sur-apprentissage.

La catégorie *sunseed* est intéressante car elle n'est jamais affectée seule à un document. En particulier, sur l'ensemble d'apprentissage, cette catégorie apparaît conjointement avec la catégorie *soybean* (graines de soja) 9 fois sur 11. Comme les classifieurs sont appris de manière indépendante, il est difficile de distinguer *sunseed* de *soybean*, en particulier lorsqu'une des règles apprises est *Si* hectares $=$ *v et* soybean $=$ *v alors sunseed* $=$ *v*.

La forte variation du critère F_1 pour ces différentes approches ne reflète pas les problèmes rencontrés par ces catégories, qui sont similaires à ceux des groupes précédents.

8.3.5 Discussion

En somme, nous avons mis en évidence deux groupes où les propriétés de l'apprenant CHARADEP étaient importantes. Ces groupes représentent un nombre restreint de caté-

9. COMP, SCAR et RELIEF+ ne permettent pas non plus de retrouver ces textes.

	Matières premières		Catégories abstraites	
Sélection	F_1	mF_1	F_1	mF_1
SCAR	0.638	0.604	0.711	0.632
RELIEF+–100	0.625	0.592	0.688	0.614
IM–75	0.654	0.633	0.703	0.632
COMP	0.812	0.813	0.329	0.349

TAB. 8.10 - *Performances relatives des méthodes de sélection et* CHARADEP *pour les groupes des matières premières et les catégories abstraites*

gories, un peu plus d'une trentaine. Pourtant, ils sont plus intéressants que la majorité des catégories de la collection où le manque d'exemples ne permet pas d'évaluer l'apport véritable de l'apprentissage et les différences entre les méthodes de réduction. Les résultats obtenus sont résumés à la table 8.10. Cette étude locale à quelques catégories nous a montré l'influence de la cohérence des règles. Dans certains cas, c'est une contrainte qui n'est pas adaptée à la catégorisation, en particulier lorsque le vocabulaire est limité.

8.4 Conclusion

Dans ce chapitre, nous avons montré que les idées sous-jacentes au modèle encapsulé n'influaient pas sur les performances d'un système de catégorisation, mais avaient un impact non négligeable sur la complexité de l'apprentissage et la qualité des classifieurs induits. Cette stabilité des performances peut s'expliquer de deux manières. D'une part, elle est liée à l'ambiguïté introduite lors de la réduction, dont nous avons évalué le coût. D'autre part, la sélection des attributs par l'apprenant, qui est implicite, compense la réduction explicite réalisée en amont dès qu'un certain nombre d'attributs descriptifs est retenu.

Les résultats obtenus ont confirmé les propriétés a priori, que nous avons montrées au chapitre précédent, pour les trois approches de réductions. En particulier, nous avons montré que les unités retenues par SCAR et RELIEF+ étaient en général caractéristiques mais pouvaient entraîner des erreurs d'étiquetage. Les unités sélectionnées par IM offrent un compromis entre rappel et précision, mais ne sont pas toutes utiles lors de l'apprentissage avec CHARADEP.

Finalement, une analyse locale nous a permis de faire émerger deux groupes de catégories. Le premier groupe est caractérisé par un vocabulaire restreint. Dans un tel cadre, un classifieur simple permet d'obtenir des performances respectables. Le second groupe regroupe des catégorie plus abstraites, décrites par un large vocabulaire. Dans ce cas, un classifieur plus complexe permet de fragmenter la représentation de la généralisation induite.

Nous avons mis en évidence ces propriétés à l'aide de CHARADEP et des approches de réduction. Dans le chapitre suivant, nous étudierons si elles se généralisent à d'autres schémas d'apprentissage. En particulier, nous modifierons la forme des classifieurs, puis la manière de construire ceux-ci.

Chapitre 9

Évaluation de différents schémas d'apprentissage

Nous concluons nos travaux par l'évaluation de différents schémas d'apprentissage appliqués à la catégorisation de textes. En fait, nous avons indirectement commencé cette évaluation au chapitre précédent lorsque nous avons mis en relation COMP et CHARADEP, puisque les méthodes de réduction ne sont pas apparues foncièrement différentes lors de l'étude de quelques catégories représentatives.

Les travaux présentés aux chapitres 7 et 8 reposent sur une hypothèse simple : seule la présence d'une unité convoie une information intéressante pour la catégorisation. Cependant, la réduction avec la méthode IM a montré que certaines unités pouvaient être sélectionnées en raison de leur absence. La catégorie *acq* (fusions, achats d'entreprises), étudiée à la section 7.4, est un exemple où l'unité vs est retenue en raison de son absence. Il est donc possible que la classification tire parti de ce type d'informations. Par ailleurs, nous avons fait abstraction du genre des collections documentaires manipulées. Dans une application de la catégorisation à l'indexation avec un vocabulaire contrôlé, la collection documentaire traite d'un domaine restreint, circonscrit par le langage documentaire. Travailler sur un tel domaine permet d'utiliser des relations entre les différents concepts pour construire les classifieurs individuels. Par exemple, les catégories *gold* (or) et *silver* (argent) décrivent toutes deux des métaux précieux. Si l'on parvient à déterminer qu'un document traite d'un métal précieux, la conjugaison de la présence du mot silver et de l'absence du mot gold est un indicateur fort pour préférer la catégorie *silver*.

Dans une première partie, nous évaluons si l'absence a une influence sur la qualité des classifieurs induits. Dans cette optique, nous employons à nouveau CHARADE en modifiant les contraintes sur la forme des règles engendrées. La manière de construire les classifieurs est la même qu'au chapitre précédent mais nous autorisons à présent les règles à faire référence explicitement à l'absence d'une unité ou à prédire le fait qu'un document ne possède pas une catégorie.

Dans une seconde partie, nous étudions le comportement des autres apprenants présentés au chapitre 6 et qui s'appuient sur une représentation vectorielle des exemples, c'est-à-dire qu'ils utilisent à la fois la présence et l'absence des unités et des catégories. Non seulement la forme des classifieurs varie (arbre de décision, fonction numérique), mais aussi la façon de les construire. Nous avons choisi des algorithmes classiques en apprentissage, issus de la bibliothèque de programmes MLC++ [KJL$^+$94].

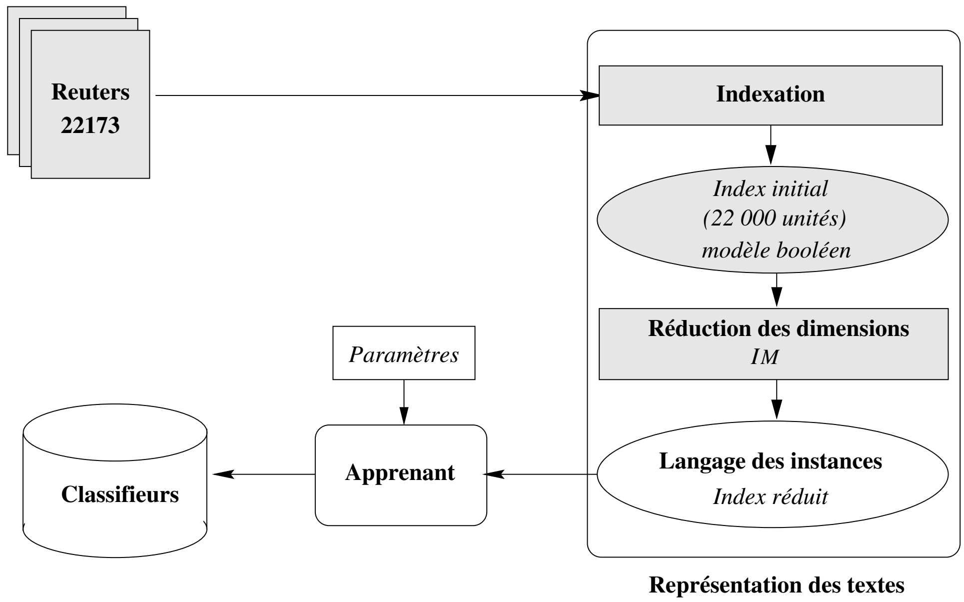


FIG. 9.1 - *Nouvelle instanciation du cadre proposé au chapitre 1. Les parties grisées correspondent aux composants invariants des expérimentations. L'index réduit varie selon le nombre d'attributs retenus.*

9.1 Cadre expérimental

Comme dans le chapitre précédent, nous pouvons figer certains composants du cadre proposé au chapitre 1 (voir figure 9.1). Nous utilisons à nouveau la collection Reuters-22 173 et l'index original construit par D. Lewis. Cet index est ensuite réduit par la méthode IM, mais le nombre d'attributs retenus peut varier, pour prendre en compte les particularités de chaque apprenant. Nous avons choisi IM car c'est la seule approche parmi les trois proposées jusqu'ici à prendre en compte l'absence et la présence des unités et catégories. Le langage des instances reste donc inchangé, mais les exemples d'apprentissage sont à présent décrits à l'aide de descripteurs positifs $\mathsf{X} = v$ (l'unité X est présente) et de descripteurs négatifs $Y = f$ (la catégorie Y n'est pas affectée à l'exemple).

Dans la première partie, nous utilisons deux versions de CHARADE. La première, notée CHARADEN, est contrainte à employer présence et absence en prémisses des règles pour décider d'affecter une catégorie. Ainsi l'absence d'une catégorie reste un défaut, c'est-à-dire que les règles ne concluent pas sur des descripteurs négatifs. La deuxième version est en fait la version courante de CHARADE, c'est-à-dire qu'il n'y a plus de contraintes sur la forme des généralisations. Cependant, il est nécessaire de fixer une stratégie pour résoudre les conflits lors du déclenchement de règles contradictoires sur un même document. Nous avons choisi d'ordonner les règles sous la forme d'une liste de décision. Ainsi, nous utilisons la conclusion de la première règle qui est déclenchée. Ceci nous a amenée à ajouter une

Si sunflower $=$ *v et* hectares $=$ *v Alors* *sunseed* $=$ *v*

Sinon si sunflower $=$ *f et* hectares $=$ *v Alors* *sunseed* $=$ *f*

Sinon si sunflower $=$ *v et* soybean $=$ *v et* maize $=$ *f Alors* *sunseed* $=$ *v*

Sinon *sunseed* $=$ *f* (règle par défaut)

FIG. 9.2 - *Un exemple de liste de décision pour la catégorie* sunseed *(graines de tournesol)*

règle par défaut, qui prédit toujours de ne pas affecter la catégorie étudiée. Nous appelons cette version CHARADEDL. Nous donnons par ailleurs un exemple de liste de décisions à la figure 9.2.

Dans la seconde partie, nous utilisons CHARADEDL comme représentant des apprenants de règles. Nous employons ID3 pour construire des arbres de décision. ID3 ne met pas en œuvre de processus d'élagage. Pourtant les généralisations construites ne sont pas forcément cohérentes puisque le critère d'arrêt est double : un nombre d'exemples minimal par feuille ou la cohérence de la feuille. Les apprenants numériques sont issus, comme ID3, d'une bibliothèque de programmes pour l'apprentissage. NaiveBayes, le classifieur Bayésien, est différent des classifieurs probabilistes utilisés habituellement en recherche documentaire (cf. [Fuh89, Lew92] par exemple). En particulier, NaiveBayes s'appuie sur une hypothèse d'indépendance entre les attributs (hypothèse remise en cause par [Coo91] pour la recherche documentaire). De plus l'évaluation de la probabilité d'un attribut de fréquence nulle est faite de manière naïve. Finalement, les expériences que nous avons réalisées avec IB, qui implémente la méthode des plus proches voisins, sont limitées en raison du coût de calcul lié à l'implémentation. À titre indicatif, il nous faut environ 20 heures réelles pour construire un classifieur et le tester pour une seule catégorie.

9.2 Évolution du langage des hypothèses avec CHARADE

Les expériences présentées dans cette section vont nous permettre d'évaluer le poids du langage des hypothèses dans le cycle de la catégorisation. Nous partons des résultats obtenus au chapitre précédent où les règles n'utilisent que des descripteurs positifs (X $=$ *v*) en prémisses et en conclusion. Nous autorisons progressivement l'utilisation de descripteurs négatifs (X $=$ *f*), tout d'abord en prémisses puis en conclusion.

Nous montrons que l'ajout de ces descripteurs négatifs permet d'obtenir de meilleurs classifieurs. La qualité des classifieurs est étudiée de manière globale et se mesure par la balance entre rappel et précision. Nous nous intéressons ensuite aux deux groupes de catégories qui ont émergé et confirmons cette tendance lorsque le langage des hypothèses évolue.

9.2.1 Résultats sur l'ensemble de la collection

Lorsque l'on enrichit le langage des instances ou que l'on modifie le langage des hypothèses, l'apprentissage peut être facilité comme il peut être rendu beaucoup plus difficile.

Apprenant	F_1	Gain / COMP	Gain / CHARADEP	Gain / CHARADEN
CHARADEN–10	0,584	7,1	13,8	–
CHARADEN–30	0,753	38,2	10,4	–
CHARADEN–50	0,751	37,8	5,2	–
CHARADEN–75	0,750	37,6	2,5	–
CHARADEDL–10	0,587	7,7	14,4	0,5
CHARADEDL–30	0,763	40,0	11,9	1,3
CHARADEDL–50	0,775	42,2	8,5	3,2
CHARADEDL–75	0,783	43,6	7,0	4,4

TAB. 9.1 - *Performance pour les deux langages des hypothèses différents, en fonction du nombre d'attributs. Les trois dernières colonnes indiquent le gain obtenu pour chaque approche.*

Nous montrons que CHARADE appliqué à la catégorisation de textes induit des classifieurs de meilleure qualité lorsque nous ajoutons des descripteurs relatifs à l'absence, mais c'est au détriment de la complexité du processus d'apprentissage.

9.2.1.1 Une amélioration de la performance

La performance moyenne, exprimée à l'aide de la mesure F_1, est rapportée à la table 9.1. Les résultats présentés sont obtenus avec le même jeu de paramètres lors de l'apprentissage. Tout d'abord, observons que le gain par rapport à COMP est notable pour toutes les approches, les performances de CHARADEDL dominant légèrement les autres configurations.

Il existe des différences entre CHARADEP, CHARADEN et CHARADEDL mais elles sont moins franches. Nous constatons que l'écart entre CHARADEP et les deux autres apprenants se réduit lorsque le nombre d'attributs augmente. À l'inverse, CHARADEN et CHARADEDL ne se démarquent que lorsque le nombre d'unités croît. En particulier, CHARADEN atteint un plafond très tôt, vers 30 attributs, alors que CHARADEDL peut encore tirer parti des informations contenues dans les unités supplémentaires.

Cette différence du nombre d'attributs optimal selon l'apprenant se justifie par l'observation suivante : CHARADEP et CHARADEN, qui n'autorisent pas de descripteur négatif en conclusion d'une règle (c'est-à-dire de prédire qu'un document n'appartient pas à une classe), emploient un nombre de descripteurs semblable, puisque CHARADEP s'appuie sur la présence d'environ 75 unités, et CHARADEN sur la présence et l'absence de 30 unités. CHARADEDL s'attache de plus à décrire les documents qui n'appartiennent pas à une catégorie donnée. Il semble donc naturel que cette approche nécessite un nombre de descripteurs supérieur (double).

La mesure F_1, qui correspond à une fonction bi-variable, ne permet pas de déterminer simplement la qualité des classifieurs induits, en fonction des erreurs et des oublis commis. La figure 9.3 résume le comportement du rappel et de la précision pour chaque approche par rapport au nombre d'attributs. L'allure similaire de certaines courbes est remarquable. En particulier, le rappel et la précision de CHARADEP et de CHARADEN varient dans des proportions semblables. Par contre CHARADEDL se distingue par une précision sujette à des écarts moins importants. C'est principalement l'utilisation de descripteurs négatifs en conclusion qui est à l'origine de ce comportement. La structuration de la base de règles en liste de décisions [1] permet de prédire tôt qu'on ne doit pas affecter la catégorie, limitant ainsi les erreurs.

En somme, utiliser des descripteurs négatifs en prémisses des règles (c'est-à-dire des informations relatives à l'absence d'une unité) favorise le rappel, car cela permet d'effectuer plus de déductions. Par contre, il est nécessaire d'introduire des descripteurs négatifs en conclusion des règles pour éviter de commettre trop d'erreurs lors de ces déductions.

9.2.1.2 Une complexité croissante

Nous distinguons à nouveau la complexité calculatoire et la complexité des classifieurs obtenus. Nous montrons que le coût de calcul principal est relié à l'introduction de descripteurs négatifs dans les prémisses, tandis que prédire de ne pas affecter une catégorie se répercute dans la complexité et la qualité des bases de règles apprises.

D'un point de vue théorique, l'espace de généralisation augmente lorsque le nombre de descripteurs augmente. Si l'on évalue la complexité calculatoire en nombre de nœuds explorés dans le treillis, la borne maximale passe de 2^n lorsque l'apprentissage est effectué avec les présences seules, à 3^n lorsque l'absence est prise en compte [2]. n est le nombre d'attributs booléens qui compose le langage des instances.

La complexité calculatoire réelle croît de manière significative lorsque l'on introduit l'absence des unités dans les prémisses des règles (cf. la table 9.2). Par contre, modifier le langage des hypothèses pour introduire des descripteurs négatifs en conclusion entraîne une augmentation plus nuancée du nombre de nœuds explorés. Cette différence s'explique simplement. Quand on autorise à utiliser l'absence d'une unité dans les prémisses de règles, on double le nombre de descripteurs à la base du treillis des descriptions. Par contre, ajouter l'absence dans la conclusion des règles consiste à n'ajouter qu'un seul descripteur. Le passage de CHARADEN à CHARADEDL devrait être négligeable devant celui de CHARADEP à CHARADEN. Mais le manque d'exemples étiquetés par une classe restreint fortement la croissance du treillis pour CHARADEN, alors que CHARADEDL développe des nœuds couvrant les exemples négatifs, plus nombreux.

Le nombre de nœuds explorés ne reflète pas la complexité de la base de règles engendrée, comme le montre la table 9.3. En effet, alors que CHARADEN explore 3 fois plus de nœuds que CHARADEP, le nombre de règles induites reste similaire, ainsi que le taux d'utilisation. Par contre, les classifieurs construits par CHARADEDL comportent de nombreuses règles, concluant principalement sur des descripteurs négatifs. En effet,

1. Lors de cette structuration, en effet, les règles sont ordonnées en fonction de leur poids, quelle que soit leur conclusion.

2. En pratique, on limite le nombre de termes dans les prémisses, ce qui permet d'avoir une complexité polynomiale et non plus exponentielle.

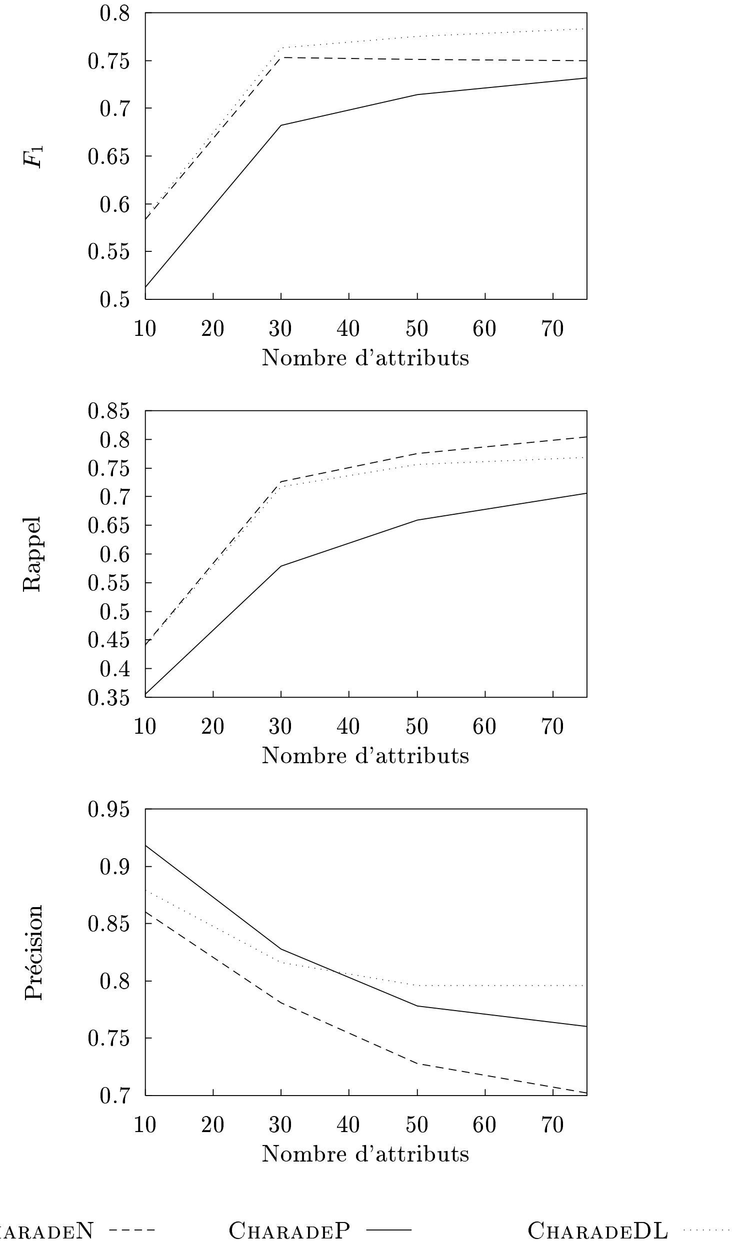


FIG. 9.3 - *Influence du nombre d'attributs sur les systèmes de catégorisation induits*

Apprenant	Nombre de nœuds
CHARADEP–75	446 242
CHARADEN–75	1 252 591
CHARADEDL–75	1 684 113

TAB. 9.2 - *Complexité calculatoire pour* CHARADE, *selon le langage des hypothèses choisi.*

Apprenant	Nombre de règles	Règles utiles (%)	Premières règles (%)
CHARADEP–75	3613	2673 (74,0)	1558 (43,1)
CHARADEN–75	3745	2481 (66,2)	1654 (44,2)
CHARADEDL–75	17277	5508 (31,9)	–

TAB. 9.3 - *Augmentation de la taille de la base de règles globale, qui résulte de l'ajout de l'absence*

CHARADEDL engendre près de 14 000 règles concluant sur l'absence d'une catégorie en plus des règles engendrées par CHARADEN qui permettent d'affecter une catégorie.

Nous avons suggéré d'employer le taux d'utilisation pour traduire la qualité des bases de règles. D'après ce critère, le classifieur induit par CHARADEDL est médiocre. C'est en fait une conséquence de la structuration en liste de décisions, puisque pour chaque exemple (et chaque catégorie) une seule règle peut se déclencher, à l'inverse des autres approches. Pour évaluer cette contrainte, nous introduisons le taux d'utilisation limité aux premières règles déclenchées, ce qui apparaît dans la dernière colonne de la table 9.3. Les taux obtenus pour CHARADEP et CHARADEN diminuent alors considérablement. Le taux obtenu par CHARADEDL reste inférieur, ce qui est principalement du aux règles concluant sur l'absence d'une catégorie. Les exemples négatifs sont en effet nombreux et ne partagent que rarement des unités communes. La cohérence des généralisations entraîne alors une sur-apprentissage et fragmente le concept «ne possède pas la catégorie X».

9.2.2 Étude par catégorie : confirmation des résultats précédents

Lors de l'étude des catégories effectuée au chapitre précédent, nous avons vu émerger deux groupes de catégories selon leur comportement. Le premier groupe rassemble des catégories qui traitent d'un domaine relativement contraint, comme les matières premières (chocolat, argent, or) et pour lesquelles la présence de quelques mots est très importante. Le deuxième groupe réunit des catégories qui traitent de domaines plus abstraits, tels que les cours monétaires ou les indicateurs économiques, et pour lesquelles les documents emploient un vocabulaire varié. Toutes les catégories de la collection Reuters n'entrent pas dans ces deux groupes. En particulier, il est difficile d'obtenir des résultats fiables pour les catégories de faible fréquence.

Avant de raffiner ces regroupements, observons la conséquence, pour l'étude locale aux catégories, des relations entre les classifieurs induits par CHARADEN et ceux induits par CHARADEDL. Les règles concluant sur l'affectation d'une catégorie sont les mêmes dans les deux cas. À ces règles, CHARADEDL ajoute des règles décidant de ne pas affecter une

	Matières premières		Catégories abstraites	
Classifieur	F_1	mF_1	F_1	mF_1
CHARADEP	0.699	0.663	0.778	0.637
CHARADEN	0.732	0.698	0.790	0.648
CHARADEDL	0.726	0.695	0.828	0.681
COMP	0.792	0.794	0.518	0.387

TAB. 9.4 - *Performances comparées des variantes de* CHARADE *pour les groupes des matières premières et les catégories abstraites*

```
(defrule learn-R57 "57"
   (declare (salience 4))
   (agriculture == true)
   (wheat == false)
   (tonnes == false)
   (u.s == false)
  =>
   (assert ( rice == false )))

(defrule learn-R61 "81"
   (declare (salience 5))
   (farmers == false)
   (said == true)
   (farm == true)
  =>
   (assert ( rice == false )))

(defrule learn-R82 "2744"
   (declare (salience 177))
   (crops == false)
   (said == false)
  =>
   (assert ( rice == false )))

(defrule learn-default
   (declare (salience 0))
  =>
   (assert ( rice == false )))
```

FIG. 9.4 - *Les règles de la liste de décision décrivant rice qui augmente l'oubli.*

catégorie et utilise une structuration en liste de décisions pour ordonner les règles en fonction de leur généralité. En conséquence, le nombre de fois où les classifieurs construits avec CHARADEDL décident d'affecter une catégorie à des documents sera toujours inférieur ou égal au nombre de fois où ceux induits avec CHARADEN le décident. CHARADEDL diminue donc le nombre d'erreurs mais aussi le nombre de décisions correctes prises par le système de catégorisation. La table 9.4 résume les performances des classifieurs en fonction du langage des hypothèses pour les deux groupes que nous détaillons.

9.2.2.1 Le groupe des matières premières

Nous reprenons les catégories étudiées au chapitre 8 pour ce groupe. Commençons par la catégorie *rice* (riz), qui illustre notre remarque précédente sur le comportement de CHARADEDL. En effet, la performance du classifieur induit avec CHARADEDL est inférieure à celles des autres approches, car celui-ci ne retrouve pas des textes que les autres récupèrent. Ces oublis sont dus aux quatre règles de la figure 9.4. Le nombre d'erreurs reste constant.

```
(defrule learn-R2 "1"                (defrule learn-R5 "1"
   (silver == true)                     (gold == false)
   (gold == false)                      (ounce == true)
   (minerals == true)                 =>
  =>                                    (assert ( silver == true )))
   (assert ( silver == true )))

(defrule learn-R10 "2"               (defrule learn-R18 "4"
   (silver == true)                     (gold == false)
   (gold == false)                      (ounces == true)
   (copper == true)                   =>
  =>                                    (assert ( silver == true )))
   (assert ( silver == true )))
```

FIG. 9.5 - *Extrait de la base de règles apprises par* CHARADEN *pour décrire la catégorie silver*

La catégorie *cocoa* (cacao) représente un cas particulier dans la mesure où CHARADEDL et CHARADEN engendrent des classifieurs au comportement identique. La raison en est simple : toutes les règles concluant sur l'affectation de *cocoa* sont classées avant les autres dans le classifieur induit par CHARADEDL. Remarquons par ailleurs que le nombre d'attributs retenus a une influence sur la reconnaissance de cette catégorie. COMP s'appuie sur un unique mot, cocoa, mais obtient des résultats supérieurs aux approches plus complexes. Pourtant, il est nécessaire d'utiliser de nombreux attributs pour obtenir de bonnes performances, les attributs ayant la plus forte information mutuelle avec la classe ne permettant pas de différencier les documents possédant la catégorie *cocoa* des autres documents.

Nous avons introduit l'absence des unités dans les prémisses des règles dans le but de différencier deux catégories proches, comme *gold* et *silver*. Or nous observons que la reconnaissance de la catégorie *silver* est détériorée. Conformément à notre attente, certaines règles font usage de la négation sur le terme gold (voir la figure 9.5). Cependant, certaines sont inutilisées lors du test tandis que d'autres sont source d'erreurs : sur 8 erreurs commises sur la base de test, 5 le sont par les règles *R5* et *R18*, faisant intervenir l'absence de *gold*.

Pour conclure, remarquons que certaines catégories, par exemple *oilseed* (graines oléagineuses) ou *veg-oil* (huile végétale), étaient classées dans le groupe des catégories abstraites. L'ajout de l'absence a dans leur cas un effet négatif qui les rapproche du groupe plus naturel des matières premières. En effet, que ce soit CHARADEN ou CHARADEDL, les classifieurs font de nombreuses erreurs, en général sur des documents dont la catégorie est proche de *oilseed*, comme *veg-oil* ou *soy-oil* (huile de soja). De plus les classifieurs relatifs à d'autres catégories comme *coffee* (café) ou *sugar* (sucre), que nous avions regroupé sous l'étiquette «groupe non homogène», montrent à présent des propriétés semblables aux catégories de ce groupe, dans la mesure où le rappel augmente pour une précision constante.

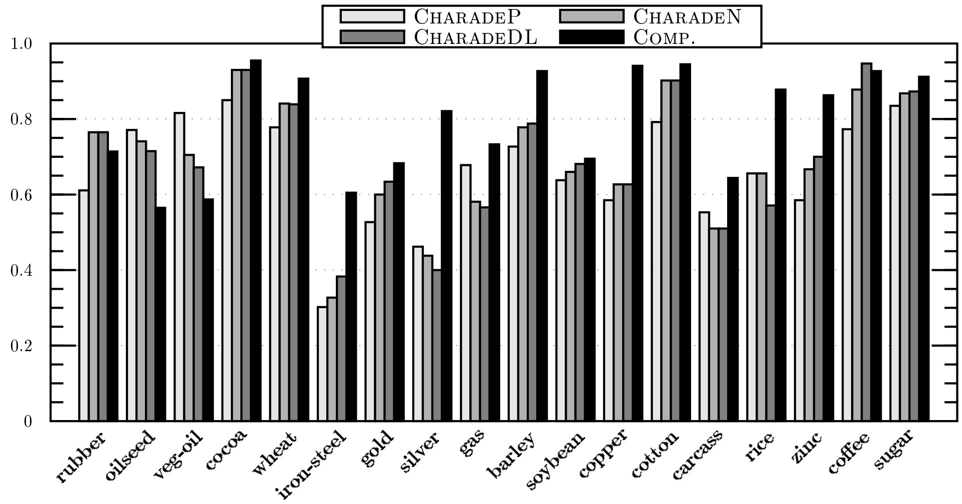


FIG. 9.6 - *Comparaison entre les approches pour le groupe des matières premières*

Catégorie	CharadeP		CharadeN		CharadeDL	
	R.(%)	P.(%)	R.(%)	P.(%)	R.(%)	P.(%)
dlr	54,2	51,0	86,5	40,0	74,0	56,3
acq	69,6	90,3	87,1	84,8	83,7	92,0
trade	65,5	59,3	76,0	53,0	75,0	62,0
yen	23,5	20,0	35,3	21,4	32,3	26,8

TAB. 9.5 - *Rappel et précision des catégories dlr, acq, trade et yen.*

9.2.2.2 Des catégories plus abstraites

La catégorie *jobs* (chômage), qui est un indicateur économique, était pourtant classée dans le groupe précédent. C'est en fait une catégorie qui traite d'un domaine circonscrit. L'unique mot unemployment permet de reconnaître un grand nombre des documents sans commettre trop d'erreurs.

Dans la majorité des cas, autoriser l'emploi de l'absence des unités permet d'étiqueter plus de documents. Le nombre de déductions faites par les classifieurs engendrés par CharadeN est élevé, mais conduit aussi à plus d'erreurs. CharadeDL conserve en général ce nombre de déductions correctes mais diminue notablement le nombre d'erreurs. Considérons les catégories *dlr* (dollar), *acq*, *trade* (commerce) et *yen* (yen) étudiées au chapitre précédent. Nous observons une augmentation du nombre de documents correctement étiquetés pour un nombre de documents erronés constant. La table 9.5 montre les propriétés des trois apprenants en donnant le rappel et la précision pour les quatre catégories précédentes. CharadeN se distingue par un rappel plus fort que les autres approches, mais c'est au détriment de la précision. Schématiquement, les règles apprises se déclenchent aussi souvent à tord qu'à raison. L'ajout de l'absence en conclusion permet de restreindre ce nombre d'erreurs tout en n'aggravant pas l'oubli.

Finalement la majorité des catégories abstraites font peu appel à la sélection d'attri-

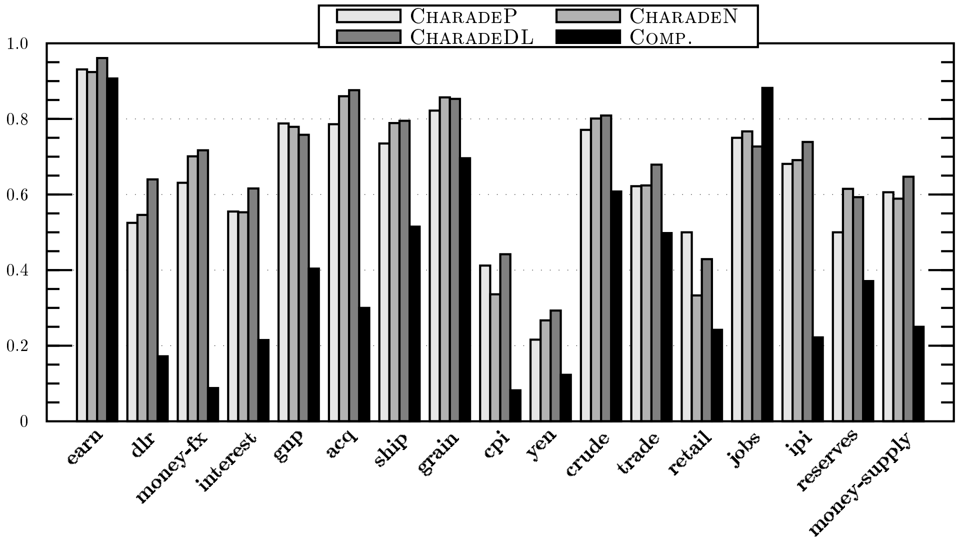


FIG. 9.7 - *Comparaison entre les approches pour le groupe des classes abstraites*

buts implicite à l'exploration de l'espace de généralisation. Nous expliquons ce comportement en observant que les documents étiquetés par ces catégories utilisent un vocabulaire très varié, et ont donc peu d'unités en commun. Ainsi CHARADE a besoin la plupart des unités sélectionnées pour engendrer les classifieurs décrivant chaque catégorie, en particulier lorsque l'on cherche aussi à prédire l'absence d'une catégorie. Nous développerons cette remarque lors de la comparaison avec ID3.

9.2.3 Conclusion

Utiliser les descripteurs correspondant à l'absence d'unités en prémisses permet de diminuer l'oubli d'un système de catégorisation. Si l'on souhaite favoriser la précision, il devient alors nécessaire d'introduire aussi des descripteurs négatifs en conclusion des règles. En particulier, la structuration en liste de décisions assure la diminution du nombre d'erreurs. La complexité des calculs et du résultat est encore une fois amplifiée par la recherche de généralisations cohérentes.

Cette caractéristique se retrouve dans l'étude des catégories relatives aux matières premières, composant le premier groupe. En effet, COMP montre toujours des résultats meilleurs qu'un apprentissage avec CHARADE. Par contre, l'emploi de l'absence des unités comme des catégories permet de mieux reconnaître les catégories plus abstraites.

9.3 Modification de l'exploration de l'espace de généralisation : ID3 et CHARADEDL

Après avoir montré l'influence de la réduction, puis du langage des hypothèses sur un système de catégorisation, nous étudions si l'exploration de l'espace de généralisation joue un rôle important, surtout au niveau du regroupement de catégories.

Nombre d'attributs	CHARADEDL ($\lambda = 2$)		CHARADEDL ($\lambda = 1$)		ID3		ID3/10	
	F_1	mF_1	F_1	mF_1	F_1	mF_1	F_1	mF_1
10	0,586	0,289	0,577	0,262	0,740	0,329	0,752	0,297
30	0,763	0,365	0,747	0,340	0,768	0,355	0,770	0,284
50	0,775	0,362	0,754	0,323	0,774	0,350	0,778	0,282
75	0,783	0,347	0,761	0,316	0,776	0,344	0,782	0,284

TAB. 9.6 - *Performances comparées entre* CHARADEDL *et ID3.*

Avant d'exposer nos expérimentations, rappelons les différentes qui existent entre CHARADEDL et ID3 :

- le langage des hypothèses. CHARADE engendre des systèmes de règles alors qu'ID3 crée un arbre de décision. Notons cependant que tout arbre peut être traduit en système de règles.
- la stratégie d'exploration. Alors que CHARADE parcourt un treillis de manière quasi-exhaustive en utilisant une stratégie *séparer pour régner*, ID3 construit récursivement l'arbre en appliquant une stratégie *diviser pour régner*.

Nous nous proposons à présent de mesurer l'influence de la stratégie d'exploration sur le système de catégorisation engendré.

9.3.1 Résultats sur l'ensemble de la collection

Nous employons ID3 de deux manières, en faisant varier le critère d'arrêt de la construction. Nous appelons ID3 la version qui s'arrête quand un nœud représente une généralisation cohérente, et ID3/10 celle qui s'arrête dès qu'un nœud couvre dix exemples. ID3 est directement comparable à CHARADEDL, tandis que ID3/10 nous donne un premier aperçu de l'influence de la cohérence.

9.3.1.1 Des performances comparables

La table 9.6 synthétise les performances moyennes obtenues par ID3 et ID3/10. Nous rappelons également les performances de CHARADEDL, en faisant varier la redondance (λ). Nous donnons la micro-moyenne à la colonne F_1 et la macro-moyenne à la colonne mF_1 [3].

Ces résultats suscitent plusieurs commentaires. Lorsque nous comparons CHARADEDL–($\lambda = 1$) et ID3, nous observons dans tous les cas un léger avantage pour ID3. La différence entre ces apprenants est la stratégie d'exploration mise en œuvre, car tous deux engendrent des généralisations cohérentes avec les données. La stratégie *diviser pour régner* semble donc être mieux adaptée, dans cette configuration simple, à la construction de systèmes de catégorisation. Par ailleurs, l'intérêt des différents paramètres de chaque

3. Nous avons figé la valeur du rapport 0/0 à 0 ce qui correspond à une évaluation pessimiste de mF_1.

Nombre d'attributs	ID3	ID3/10
10	30 339	11 583
30	142 791	49 607
50	235 071	86 257
75	313 062	119 330

TAB. 9.7 - *Complexité de la construction d'un système de catégorisation à partir d'arbres de décision pour la collection Reuters*

apprenant ressort aussi des résultats présentés. En effet, la redondance pour CHARADEDL et le critère d'arrêt pour ID3/10 permettent d'obtenir des résultats semblables pour la mesure F_1. Pourtant, il existe une différence majeure entre les classifieurs si nous nous référons à la mesure mF_1. Ainsi, ID3/10 subit les effets de la condition d'arrêt : en effet le seuil de dix exemples conduit certains classifieurs à ne jamais affecter les catégories peu fréquentes, qui possèdent un nombre de documents d'apprentissage inférieur à dix.

La figure 9.8 récapitule les tendances à l'oubli ou à l'erreur de CHARADEDL, avec $\lambda = 2$, d'ID3 et d'ID3/10. Comme le suggérait les variations entre les mesures F_1 et mF_1, nous observons que ID3/10 favorise la précision au dépens du rappel, puisque dans de nombreux cas, les arbres construits par ID3/10 préfèrent ne pas affecter la catégorie.

Cette figure montre aussi que les performances d'ID3 et d'ID3/10 ne semblent pas influencées par le nombre d'attributs. Dans les deux cas, le système parvient à de bons résultats même avec dix attributs, ce qui n'est pas le cas du système induit par CHARADEDL. En fait, même avec quatre attributs, les performances des systèmes appris avec ID3 restent honorables. Par exemple, la mesure F_1 vaut 73,1% pour ID3/10.

9.3.1.2 Une complexité plus faible pour ID3

Les performances obtenues par CHARADEDL et ID3 sont comparables. Mais les deux apprenants n'ont pas la même complexité. La complexité de CHARADEDL se mesure en nombre de nœuds du treillis explorés. Nous proposons de calculer la complexité calculatoire d'un arbre de décision de la manière suivante : à chaque nœud, il faut tester si l'on arrête la construction. Si la construction se poursuit, il faut choisir le meilleur attribut. La complexité est donc de $N \times C_S + (N + f) \times C_A$, où N est le nombre de nœuds, f le nombre de feuilles, C_A le coût du test d'arrêt et C_S celui de la sélection. Nous fixons C_A à 1 puisqu'il s'agit d'un seul test. Pour choisir une variable, il faut calculer l'information mutuelle entre chaque attribut et la catégorie. Nous estimons donc le coût C_S à n, où n est le nombre d'attributs. Lorsque ce nombre n augmente, le facteur $(N + f) \times C_A$ représente un coût faible par rapport à $N \times C_S$.

La table 9.7 résume la complexité calculatoire d'ID3 et d'ID3/10, suivant le nombre d'attributs. Si nous comparons cette complexité avec celle rapportée pour CHARADEDL à la table 9.2, nous observons un avantage certain pour ID3. De plus le coût de construction d'un arbre est du même ordre de grandeur que le coût lié à l'ajout d'un descripteur à la base du treillis.

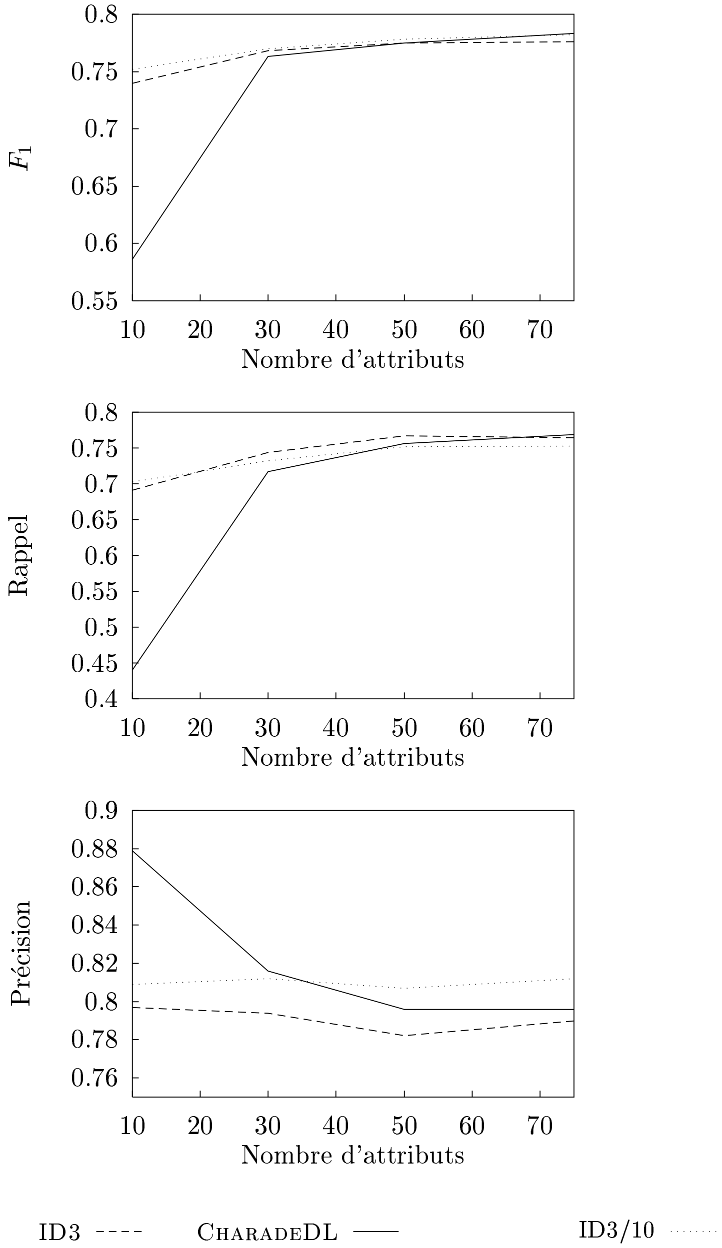


FIG. 9.8 - *Influence du nombre d'attributs sur la catégorisation*

Nb. feuilles/règles	CharadeDL ($\lambda = 2$)	CharadeDL ($\lambda = 1$)	ID3	ID3/10
Total	17277	8086	4198	1682
Présence	3745	1957	1787	444
Absence	13532	6129	2411	1238

Tab. 9.8 - *Complexité des classifieurs appris en nombre de règles. Le nombre d'attributs pour chaque catégorie est 75. La ligne Présence correspond aux règles ou feuilles permettant d'affecter la catégorie.*

De même que l'espace de généralisation parcouru par CharadeDL est plus important, le nombre de règles engendrées l'est également. En particulier, nous observons à la table 9.8 que, sans redondance, CharadeDL engendre de nombreuses règles visant à ne pas affecter une catégorie, alors que ce nombre est limité avec ID3, mais induit un nombre comparable de règles permettant d'affecter une catégorie. Ce comportement est une conséquence de la stratégie d'exploration adoptée par chaque apprenant. En effet, ID3 donne la priorité aux exemples ; ainsi, dès qu'un nœud ne contient que des exemples d'une même classe, la construction du sous-arbre s'arrête. À l'opposé, CharadeDL s'appuie sur le treillis des descriptions et permet de généraliser plusieurs fois le même exemple.

Notons pour conclure que l'abandon de la cohérence des généralisations lors de la construction des arbres a une influence non négligeable. Les complexité du processus et du résultat diminuent sans entraîner une détérioration des performances moyennes du système. Rappelons cependant que c'est au détriment des catégories de fréquence faible.

9.3.1.3 Conclusion

L'étude du comportement des classifieurs sur l'ensemble de la collection nous a permis de montrer que le présupposé d'apprentissage relatif à l'exploration de l'espace de généralisation a une influence limitée sur les performances du système de catégorisation construit, dès que les contraintes pour l'arrêt de l'exploration sont similaires.

La stratégie d'exploration *diviser pour régner* adoptée par ID3 entraîne une complexité plus faible que la stratégie *séparer pour régner* sous-jacente à CharadeDL. En particulier, ID3 n'a pas le travers de CharadeDL qui fragmente la généralisation du concept «ne possède pas la catégorie X».

Les résultats présentés ont permis de confirmer que la cohérence des classifieurs entraîne un sur-apprentissage. En effet, les classifieurs appris par CharadeDL ou ID3 sont plus complexes que ceux créés par ID3/10 et obtiennent des performances comparables. Autoriser des classifieurs non cohérents pourrait être une solution à la réduction de la complexité des classifieurs. De plus amples expérimentations sont nécessaires pour avoir une idée précise sur l'influence du comportement des apprenants qui tolèrent des généralisations non cohérentes.

Finalement, nous étudions l'impact de la méthode de réduction sur l'apprentissage avec ID3. En effet, Im et ID3 s'appuient tous deux sur l'information mutuelle pour choisir les attributs à retenir dans le premier cas, et le test à chaque nœud dans le second.

Pour évaluer si ceci a une influence quelconque sur les performances et la complexité de l'apprentissage, nous avons décrit les exemples en termes de présence et d'absence à l'aide des unités retenues par SCAR–1–3. Les résultats obtenus confirment le rôle restreint de la réduction dès lors que des attributs suffisamment caractéristiques sont retenus, comme nous l'avons mentionné au chapitre 8. En effet, la performance du système appris par ID3/10 est comparable à celle donnée à la table 9.6 : F_1 vaut 77,6% et mF_1 28,1%. La complexité des classifieurs est du même ordre de grandeur, 462 feuilles pour affecter une catégorie sur 1701.

9.3.2 Un point de vue local

Le test d'arrêt de la construction d'un arbre a une influence plus marquée pour certaines catégories. Nous étudions à présent le regroupement des catégories précédent pour évaluer le lien entre stratégie d'exploration et propriétés des données textuelles.

9.3.2.1 Quand l'arbre démarre bien ...

Certaines catégories, en majorité celles pour lesquelles COMP obtient des performances très honorables, se singularisent par l'existence d'unités très caractéristiques et assez discriminantes. Ces unités sont en général sélectionnées par IM, avec une bon score. Dans de telles conditions, ID3 choisit l'une de ces unités comme racine lors de la construction de l'arbre. Par exemple, l'arbre décrivant la catégorie *cocoa* commence par le terme cocoa ; nous observons la même propriété pour *rice*, *gold* et *silver*, ou la majorité des catégories traitant de matières premières.

À la figure 9.10, nous voyons que la catégorie *rubber* (caoutchouc) est mal reconnue. Pourtant il existe une unité caractéristique et discriminante (rubber) pour cette catégorie. En fait, pour les catégories de fréquence faible ou moyenne, le test d'arrêt à dix exemples entraîne un arbre de taille réduite, qui ne permet pas de classer certains exemples qui le sont avec l'arbre induit avec ID3.

Alors qu'ID3/10 permet de mieux décrire la catégorie *gold* que les autres approches, le classifieur induit pour la catégorie *silver* est plus mauvais : il n'affecte correctement *silver* qu'à deux documents, tandis que la base de règles induite par CHARADEP permet d'étiquettes correctement plus de textes, mais au prix d'un nombre d'erreurs supérieur. Cette différence s'explique par le nombre de règles ou de feuilles permettant de décrire l'affectation de la catégorie (cf. table 9.9). En effet, le classifieur induit par ID3/10 ne possède qu'une feuille qui conclut sur l'affectation de *silver*, tandis que CHARADEDL–2 possède vingt règles. C'est l'un des rares cas où le sur-apprentissage semble être utile.

9.3.2.2 Des arbres assez complexes pour les catégories abstraites

La table 9.9 permet de mesurer la complexité des arbres en comparant le nombre de feuilles et le nombre de nœuds. Ce dernier peut être déduit du premier en remarquant que l'arbre induit est binaire. En effet si f est le nombre de feuilles et N le nombre de

Catégorie	ID3/10	ID3	CharadeDL–1	CharadeDL–2
cocoa	2/13	5/38	7/59	15/107
rice	3/8	11/17	15/61	28/112
silver	1/5	11/17	11/64	20/121
gold	7/13	18/34	17/38	34/186
rubber	2/6	9/16	8/56	17/109
corn	15/21	25/46	41/210	77/442
dlr	7/32	47/87	39/168	75/346
trade	24/73	98/229	106/409	199/813
yen	3/15	22/43	20/84	34/153
acq	82/180	269/596	291/654	534/1233
earn	51/169	160/461	174/463	338/876

TAB. 9.9 - *Nombre de règles engendrées par ID3 et* CharadeDL. *Le premier nombre correspond au nombre de règles ou de feuilles concluant sur l'affectation de la catégorie, tandis que le second représente le nombre de règles ou de feuilles total.*

nœuds, nous avons $N = f - 1$[4]. Les catégories que nous avons qualifiées d'abstraites donnent lieu à des arbres plus complexes. Comparons par exemple le nombre de nœuds des arbres engendrés pour les catégories *corn* (maïs) et *dlr* qui ont à peu près le même nombre d'exemples d'apprentissage : l'arbre obtenu pour *dlr* est presque deux fois plus gros que celui obtenu pour *corn*. Cette complexité reste cependant inférieure à celle des classifieurs induits par CharadeDL.

Comme nous l'avons remarqué lors de l'étude sur l'ensemble de la collection, ID3 et plus particulièrement ID3/10 a tendance à favoriser la précision du classifieur au détriment du rappel. Cette tendance se retrouve au niveau des catégories. Par exemple, CharadeDL permet d'affecter correctement les catégories *dlr* et *yen* plus souvent, mais engendre plus d'erreurs. La catégorie *trade* représente une exception dans la mesure où toutes les approches ont le même comportement.

9.3.2.3 Les cas particuliers de acq, money-fx et earn

Les catégories *acq* et *money-fx* (cours monétaires), que l'on peut toujours regrouper avec les catégories abstraites, ont pour ID3 des propriétés similaires à la catégorie *earn* (bénéfices des entreprises). En effet, les classifieurs induits par ID3 commettent plus d'erreurs que ceux engendrés par CharadeDL, sans pour autant étiqueter correctement plus de documents. Notons que ce sont les catégories les plus fréquentes de la collection. Nous ne pouvons cependant pas approfondir le lien entre d'une part la fréquence et la qualité abstraite des catégories et l'apprentissage avec ID3 de l'autre, car les catégories fréquentes sont des matières premières qui vérifient le comportement du premier groupe.

4. Cette propriété se montre par récurrence sur la profondeur de l'arbre.

Catégorie	ID3/10	ID3	CHARADEDL–1	CHARADEDL–2
cocoa	11	34	54	58
rice	6	14	49	58
silver	4	14	55	60
gold	12	22	37	67
rubber	4	13	50	56
dlr	21	43	70	73
trade	39	57	73	74
yen	13	32	62	65
acq	50	70	71	73
earn	51	65	70	73

TAB. 9.10 - *Nombre d'attributs utilisés par ID3 et* CHARADEDL *parmi les 75 possibles*

9.3.3 Discussion

Nous avons mentionné précédemment que la sélection des attributs implicite à CHARADEDL n'avait qu'un effet très limité. Nous comparons à présent le rôle de la sélection implicite aux apprenants. La table 9.10 nous permet d'approfondir cette remarque. En effet, CHARADEDL utilise la majorité des attributs mis à disposition, tandis qu'ID3 effectue une sélection lors de la construction des arbres. Les catégories *acq* et *earn* représentent des exceptions pour ID3, où de nombreux attributs sont retenus, ce qui conduit à la remarque suivante : la sélection des attributs implicite à ID3 est fortement liée au nombre d'exemples possédant la catégorie.

Nous concluons cette section en rappelant les résultats obtenus (cf. table 9.13). ID3 a tendance à mieux reconnaître les catégories concrètes où quelques termes sont caractéristiques. Cependant, le choix d'un bon attribut dès le départ peut être anéanti par la condition d'arrêt lorsque celle-ci ne correspond pas à la distribution de la catégorie (*i.e.* lorsqu'il y a peu exemples positifs). Lorsque les catégories sont plus abstraites, les arbres sont complexes ; par exemple, dans ce contexte, le nombre de feuilles des arbres construits par ID3 est proche du nombre de règles engendrées par CHARADEDL–1. À la différence de CHARADEDL, ID3 favorise la précision au détriment du rappel. De plus les performances favorisent les classifieurs simples aux généralisations plus complexes, ce qui est l'opposé du comportement de CHARADEDL.

9.4 Apprentissage numérique : un aperçu

L'emploi de techniques de classification numériques est usuel en recherche documentaire. Certaines approches, tel qu'un apprenant Bayésien [LR94] ou un algorithme implémentant la méthode des k plus proches voisins [Yan94, LC96], ont été appliquées à la catégorisation de textes. Les méthodes proposées sont en général des adaptations de méthodes originales pour les particularités des données textuelles. Nous avons choisi de travailler avec une implémentation utilisée en apprentissage de ces deux méthodes. Nous

Apprenant	F_1	mF_1
NaiveBayes–2	0,686	0,339
NaiveBayes–6	0,710	0,341
NaiveBayes–10	0,703	0,324
NaiveBayes–30	0,671	0,308
NaiveBayes–50	0,647	0,287
NaiveBayes–75	0,629	0,279
ID3/10–75	0,782	0,284
CHARADEDL–75	0,783	0,347

TAB. 9.11 - *Performances de NaiveBayes sur l'ensemble de la collection Reuters*

considérons notre étude comme un aperçu dans la mesure où nous ne nous intéressons qu'à la performance des apprenants numériques et non pas à leur complexité, plus difficile à évaluer.

9.4.1 NaiveBayes : un comportement moyen

Les informations élémentaires manipulées par NaiveBayes sont des probabilités conjointes et conditionnelles. Par nature, la réponse des classifieurs induits par NaiveBayes reflète un degré de confiance pour affecter les catégories. Ce degré est transformé en décision binaire en fixant un seuil. Nous avons adopté le seuil par défaut de la bibliothèque MLC++, à savoir 0,5.

La table 9.11 résume les performances sur l'ensemble de la collection pour NaiveBayes et rappelle des résultats obtenus précédemment. La colonne F_1 fait référence à la micro-moyenne de la mesure F_1, tandis que la colonne mF_1 correspond à la macro-moyenne. La mesure F_1 est sensiblement inférieure aux valeurs obtenues par ID3 ou CHARADE. Par contre la mesure mF_1 est comparable voire supérieure aux approches précédentes. Ceci traduit le fait que NaiveBayes obtient de mauvaises performances pour les catégories fréquentes, étant donné que ce sont elles qui prédominent dans le calcul de F_1.

En fait, le comportement de NaiveBayes ne peut être expliqué qu'en prenant en compte le nombre d'attributs. Avec peu d'attributs, NaiveBayes commet globalement de nombreuses erreurs, et ne parvient pas de plus retrouver un nombre important de documents. Par contre, lorsque le nombre d'attributs augmente, le nombre de déductions globales augmente, entraînant une augmentation des erreurs mais une diminution du nombre de textes non retrouvés. La figure 9.9 résume l'influence du nombre d'attributs sur le rappel et la précision. L'allure des courbes n'est pas surprenante ; en particulier, la précision baisse plus rapidement que n'augmente le rappel.

Le fort taux d'erreur, même avec peu d'attributs, est une conséquence de la méthode mise en œuvre pour le calcul des probabilités. Les modèles proposés en recherche d'informations [Fuh89] et adaptés à la catégorisation de textes [Lew92] prennent en compte deux éléments qui ne sont pas considérés par NaiveBayes. Tout d'abord, l'évaluation des probabilités est problématique lorsque la fréquence des termes est nulle. L'attitude adoptée

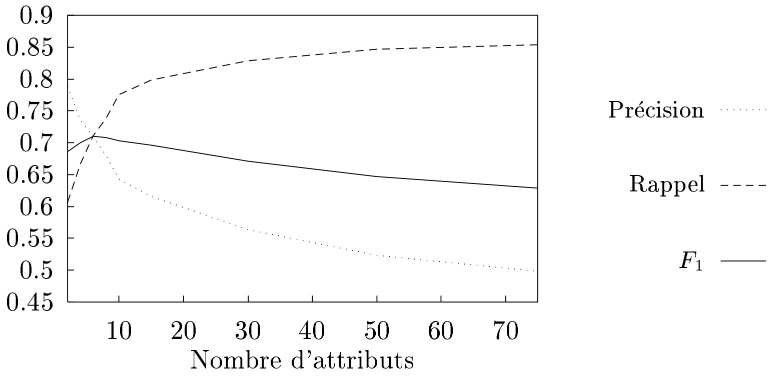


FIG. 9.9 - *Influence du nombre d'attributs pour NaiveBayes*

par NaiveBayes ne prend pas en compte le fait que c'est un cas fréquent dans une application telle que la catégorisation. Ensuite, NaiveBayes s'appuie sur l'indépendance entre les attributs, ce qui s'applique difficilement aux données textuelles (cf. [Coo91, Lew92]).

Nous concluons cette étude par une discussion des performances de NaiveBayes sur certaines catégories. Les figures 9.10 et 9.11 résument les performances comparées de NaiveBayes, IB, ID3/10 et CHARADEDL. Puisque COMP obtient de bonnes performances pour les catégories relatives à des matières premières, nous nous attendions à ce que NaiveBayes montre de bonnes performances sur ces catégories. Or comme l'illustre la figure 9.10, ce n'est pas le cas. Quelle en est la raison ? Lorsque NaiveBayes permet de retrouver un taux plus élevé de documents, c'est au détriment de la précision. Prenons trois exemples parmi les matières premières. La catégorie *silver* est mieux reconnue par NaiveBayes que par les autres approches étudiées jusqu'ici. Mais c'est au détriment d'un nombre d'erreurs élevé. Par exemple, avec six attributs, NaiveBayes retrouve 15 textes sur 16 mais commet 20 erreurs, alors que ID3/10 ne retrouve que 2 textes mais commet seulement 2 erreurs. La catégorie *cocoa* est un exemple où six attributs conduisent à une performance plus mauvaise que pour deux attributs. Ceci tend à confirmer les observations que nous avons fait pour COMP où un unique mot entraîne une meilleure reconnaissance. Les résultats mitigés obtenus pour ces catégories sont dus à deux facteurs : le nombre d'exemples faible pour évaluer les probabilités et le fort taux de recouvrement entre catégories, qui est la source de nombreuses erreurs.

Parmi les catégories plus abstraites, NaiveBayes est plus performant sur la catégorie *jobs*. Nous avons mentionné précédemment cette catégorie comme partageant une caractéristique du groupe des matières premières. Le seul terme unemployment permet de retrouver sans trop d'erreurs la majorité des textes étiquetés par *jobs*. Cette tendance est confirmée par NaiveBayes. Conformément à notre attente, NaiveBayes a un comportement médiocre sur les autres catégories abstraites. Nous concluons cette discussion par l'étude de la catégorie *yen*, qui se caractérise par un taux d'erreur excessivement élevé quel que soit le nombre d'attributs employés.

En somme, NaiveBayes, tel qu'il est disponible dans la bibliothèque de programmes

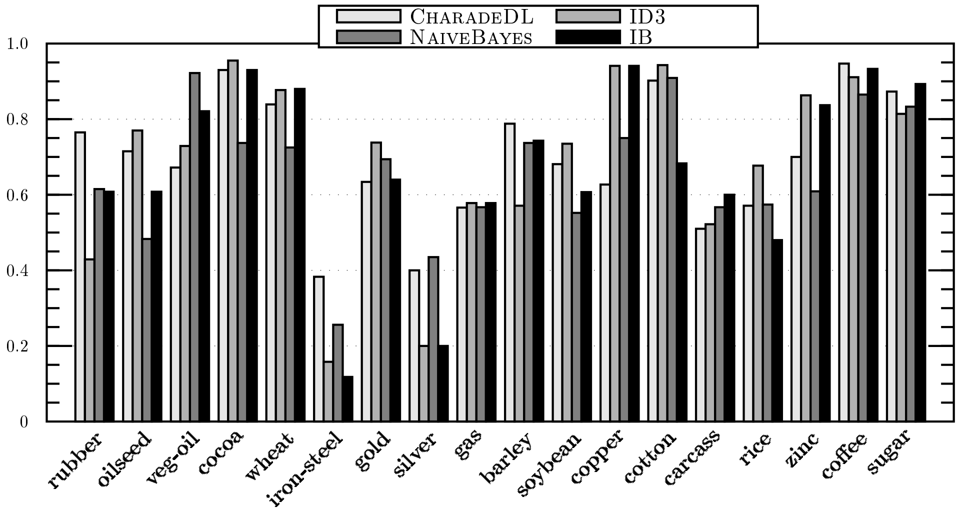


FIG. 9.10 - *Performances en termes de F_1 pour les catégories relatives aux matières premières*

MLC++, s'adapte difficilement à la catégorisation de textes. Pourtant des approches ont montré que des classifieurs Bayésiens obtenaient des performances équivalentes à celles obtenues par des arbres de décision [LR94]. Nous supposons en conséquence qu'un tel classifieur doit prendre en compte la disparité dans les données et les faibles fréquences des attributs pour parvenir à des résultats intéressants. Finalement, l'étude de la division entre les deux groupes, concret et abstrait, doit être approfondie à la lumière de la fréquence des unités et des catégories.

9.4.2 IB : une première étude

Nous nous sommes intéressée à un algorithme implémentant la méthode des plus proches voisins lorsque nous avons observé que les règles engendrées par CHARADE généralisaient un nombre d'exemples réduits. Les expérimentations menées avec IB correspondent à une première étude dans la mesure où nous avons choisi une représentation qui n'est pas forcément la meilleure pour IB et où les expériences ont été limitées en raison du coût de calcul d'IB.

Dès que l'on s'intéresse à la méthode des k plus proches voisins, le choix de k se pose. Dans le cas de la catégorisation de textes, ce choix est critique, parce que les catégories possèdent peu d'exemples positifs et de nombreux exemples négatifs. Les expériences rapportées par [Wie95], qui font varier k entre 3 et 12, suggèrent 12 comme valeur optimale sur un sous-ensemble de la collection Reuters. En partant de l'hypothèse que k est le même pour toutes les catégories, nous devons prendre en compte le fait qu'un certain nombre de catégories ne possède que peu d'exemples (moins d'une dizaine), tandis que la catégorie *earn* représente un tiers du corpus. Choisir k supérieur à 10 revient à accepter que les catégories de fréquence inférieure à 10 ne soient jamais affectées. De plus la forte proportion d'exemples de la classe *earn* peut conduire à un taux d'erreur important pour

Apprenant	F_1
IB–4	0,736
IB–10	0,749
IB–75	0,740
ID3/10–75	0,794
CHARADEDL–75	0,794
NaiveBayes	0,717

TAB. 9.12 - *Performances d'IB et les approches précédentes*

cette classe. Nous avons donc commencé par fixer k à 1.

Les expériences ont été réalisées sur un sous-ensemble de la collection Reuters, comprenant en particulier toutes les catégories étudiées précédemment. Les résultats, résumés à la table 9.12, montrent qu'IB obtient des résultats encourageants, étant donné que la représentation booléenne n'est pas forcément adaptée à ce type d'apprentissage. De manière générale, nous observons qu'IB engendre peu d'erreurs, le nombre d'erreurs diminuant quand le nombre d'attributs augmente. Le nombre d'oublis cumulés ne possède pas cette régularité : il commence par diminuer puis augmente à nouveau.

Si nous étudions les figures 9.10 et 9.11, nous observons que la performance moyenne, légèrement inférieure, provient du fait que jamais IB ne montre un avantage notable par rapport aux autres apprenants.

À l'exception de quelques catégories, IB engendre des classifieurs qui ont une performance moyenne. Dans le groupe des matières premières, remarquons la bonne performance pour la catégorie *cocoa* et la médiocre performance pour *silver* ou *rice*. Ces deux dernières catégories se caractérisent par un nombre d'erreurs très faible et par un nombre d'oublis élevés. La majorité de catégories du groupe des matières premières partage cette propriété. De plus, nous observons que lorsque le nombre d'attributs augmente, la précision augmente et le rappel diminue.

On ne retrouve pas une telle uniformité de comportement pour le groupe des catégories abstraites. Tout comme pour NaiveBayes, la fréquence de la catégorie semble rentrer en jeu. Par rapport à ID3/10 ou CHARADEDL , IB tend à retrouver moins de textes et à faire plus d'erreurs sur les catégories fréquentes.

Finalement, nous avons estimé l'impact du choix de k en passant de 1 à 5 pour quelques catégories. Les résultats sont peu concluant dans la mesure où la catégorie *earn* a une performance très mauvaise (il y a plus d'un millier d'erreurs, ce qui correspond également au nombre de déductions correctes). Ceci est lié à la remarque que nous avons faite sur le choix de k. Les autres catégories se singularisent par un nombre d'erreurs faible, tandis que le classifieur ne retrouve qu'une petite partie des documents.

Les résultats que nous obtenons avec IB sont comparables à ceux obtenus par [Wie95], mais semblent inférieurs à ceux de [Yan96b]. L'une des différences principales avec l'approche de Y. Yang est la transformation en n problèmes binaires. Yang cherche les voisins les plus proches et affecte au document étudié toutes les catégories de ses voisins, à partir d'un certain seuil. Cette approche se retrouve aussi chez [LC96].

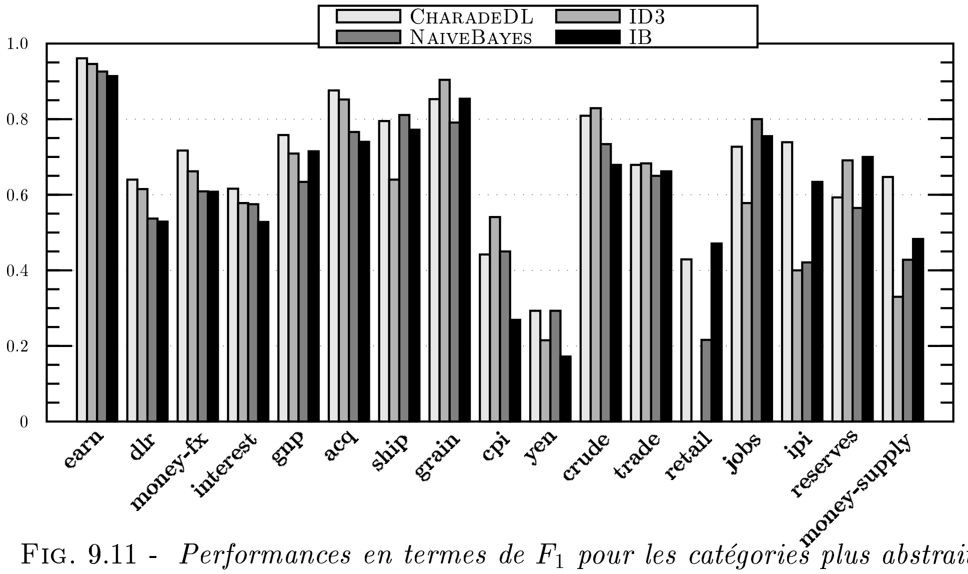


FIG. 9.11 - *Performances en termes de F_1 pour les catégories plus abstraites*

	Matières premières		Catégories abstraites	
Classifieur	F_1	mF_1	F_1	mF_1
CHARADEDL	0.726	0.695	0.828	0.681
ID3/10	0.761	0.690	0.813	0.598
NaiveBayes	0.694	0.692	0.737	0.620
IB	0.701	0.672	0.761	0.617

TAB. 9.13 - *Performances comparées des apprenants étudiés sur les deux groupes de catégories*

9.4.3 Discussion

La table 9.13 résume les performances obtenues par chaque apprenant (symbolique ou numérique) sur les deux groupes de catégories que nous avons extraits. Les résultats et conclusions relatifs aux apprenants numériques sont préliminaires. L'étude de deux apprenants numériques classiques en apprentissage nous a montré que de tels apprenants doivent être modifiés pour prendre en compte les caractéristiques des données textuelles. En particulier, l'apprenant Bayésien peut faire abstraction de l'hypothèse d'indépendance entre les attributs ou proposer une alternative au calcul élémentaire des probabilités à partir des fréquences. La méthode des plus proches voisins semble difficile à adapter en bloc. En effet, nous pensons que le nombre d'attributs et le nombre de voisins dépendent de la catégorie étudiée de manière plus frappante que pour ID3 ou CHARADEDL. De plus la transformation en n sous-problèmes semble être la source d'un rappel assez faible.

9.5 Comparaison avec d'autres travaux

9.5.1 Performances

Les expériences que nous venons de décrire correspondent à un cadre plus courant en catégorisation et nous permettent de comparer nos résultats avec les résultats obtenus par les différents travaux portant sur la collection Reuters-22173.

Cependant, nous avons souligné lors de la présentation des données qu'il était difficile de comparer les différents travaux effectués sur cette collection. En effet, chacun semble avoir adopté une transformation en bases d'apprentissage distincte. En particulier, les catégories étudiées et les documents conservés varient d'une approche à l'autre. Par conséquent, les résultats présentés ici ne sont qu'approximatifs et visent plus à donner un ordre de grandeur, qu'une comparaison exhaustive.

Parmi les différentes travaux, seule l'approche décrite dans [CS96] s'appuie explicitement sur le présupposé d'absence par défaut et est donc directement comparable aux résultats présentés au chapitre 7. Les performances de RIPPER semblent meilleures : F_1 vaut $79,1\%$, alors que le meilleur résultat obtenu avec CHARADEP est de $74,7\%$. Contrairement aux résultats que nous obtenons, l'ajout de l'absence dans les prémisses des règles n'améliore pas vraiment les performances de RIPPER ($F_1 = 79,1\%$) tandis que la différence entre CHARADEP et CHARADEDL est plus marquée.

Apté *et al.* [ADW94b] utilisent un apprenant de règles Swap-1 qui, comme RIPPER, engendre des règles qui ne sont pas contraintes à être cohérentes avec les données. À partir d'une représentation des textes sous forme de vecteurs booléens, les auteurs obtiennent un point moyen de $77,5\%$.

Dans [Wie95], E. Wiener étudie le comportement de différents apprenants, en particulier des réseaux de neurones, un algorithme implémentant la méthode des plus proches voisins et un classifieur par prototype. La performance obtenue par les réseaux de neurones, avec une architecture à plat, sont de l'ordre de $F_1 = 78\%$ tandis que l'apprenant s'appuyant sur la méthode des plus proches voisins ne dépasse pas 75% et le classifieur par prototype 67%. [Yan96a] obtient des résultats similaires avec un autre algorithme implémentant la méthode des plus proches voisins, à partir d'un modèle non booléen.

En résumé, nous observons que CHARADEDL permet d'obtenir des performances similaires aux différentes approches rencontrées. Ceci ne peut être considéré que comme une indication puisque toutes les évaluations ne portent pas sur les mêmes bases d'apprentissage et d'évaluation.

9.5.2 Discussion

Différents travaux sur la catégorisation de textes se sont appuyés sur des modèles plus complexes, en particulier en adoptant une description non pas booléenne des textes, mais numériques [WPW95, ADW94b, Yan94]. D'autres ont mis en œuvre des techniques plus poussées que celles que nous avons étudiés. Par exemple, dans [CS96], l'apprenant employé utilise une technique d'élagage qui s'appuie sur un calcul d'erreurs rétribuant différemment les exemples positifs et les exemples négatifs.

L'originalité de nos travaux ne se situe pas dans la proposition d'une méthode supérieure aux autres, mais repose sur une évaluation de différents apprenants en fonction des propriétés des différentes catégories. Peu de travaux, à notre connaissance, ne se sont attaqués à ce problème. Dans [Wie95], Wiener expose les résultats de différents schémas d'apprentissage et de réduction sur les catégories de la collection Reuters, mais ne cherche pas à déterminer les propriétés des catégorie qui donnent lieu à une différence de comportement. Par contre, Thorstens [Tho96b] étudie différents types d'applications de la catégorisation et en particulier deux catégories de la collection Reuters, dans un cadre probabiliste.

9.6 Conclusion

Nous avons dans un premier temps montré comment l'évolution du langage des hypothèses, lorsque la recherche dans l'espace de généralisation reste fixe, influe sur les performances d'un système de catégorisation. Ces résultats sont bien sur intrinsèquement liés à la nature de l'apprenant utilisé. En particulier, l'apport de littéraux négatifs avec RIPPER n'entraîne pas d'augmentation des performances franche [CS96]. À la différence de CharadeDL, RIPPER n'emploie pas de méthode de réduction en aval de l'apprentissage. Il dispose donc de plus d'informations dès le départ, alors que la modification du langage des hypothèses et des instances apporte des informations supplémentaires. Nous avons effectué d'autres expériences sur l'enrichissement des descriptions des exemples, que nous décrivons brièvement à l'annexe D.

Nous avons ensuite comparé deux apprenants symboliques, CharadeDL et ID3, en insistant sur leur capacité à manipuler un type de catégories plutôt qu'un autre. Les expériences avec ID3 nous ont permis d'estimer les limites de la cohérence. Nous avons mis à jour que la cohérence des systèmes de règles et des arbres de décision favorisaient le rappel tout en conservant la précision à un niveau sensiblement constant. Lorsque le langage des instances comporte un attribut caractéristique, ID3 se trouve avantagé par rapport à CharadeDL. Nous avons de plus remarqué une différence entre la stratégie *diviser pour régner* et la stratégie *séparer pour régner* : CharadeDL, qui utilise cette dernière engendre de nombreuses règles décidant de ne pas affecter une catégorie, alors qu'ID3 engendre un nombre de feuilles nettement plus faible.

Finalement, nous ne pouvons pas tirer de réelles conclusions des expériences menées avec les classifieurs NaiveBayes et IB, en fonction de leur performance pour les différentes catégories. Le seul point que nous pouvons assurer est la nécessité d'adapter ces méthodes pour les données textuelles, comme c'est en général effectué [Coo91, Yan94].

Conclusion

Dans ce mémoire, nous nous sommes intéressée à la catégorisation de textes, en partant de l'idée qu'elle était, comme la recherche documentaire, modélisée comme un problème de classification [Lew92]. La catégorisation de textes reste un domaine actif [HH96], car elle met en œuvre des techniques applicables à des domaines variés tels que l'archivage de courriers électroniques [Coh96a] ou le filtrage de documents [Lew96].

Il existe de nombreuses techniques de classification, que ce soit en IA ou en statistiques. Nous avons accordé notre attention aux approches inductives, qui construisent des classifieurs de manière automatique à partir d'exemples. Ces approches inductives se divisent en apprentissage numérique et apprentissage symbolique. Dans tous les cas, la catégorisation représente un défi en raison des dimensions qui la caractérisent, plusieurs dizaines de milliers de documents décrits par plusieurs dizaines de milliers de mots.

Une approche pour aborder ce problème de dimensions est de faire appel à des techniques de réduction de l'ensemble des mots employés pour décrire les textes. C'est une stratégie couramment employée dans les travaux sur la catégorisation qui s'inscrit dans le cadre des travaux d'apprentissage sur la sélection d'attributs.

Nous pouvons résumer les objectifs que nous nous sommes fixés en trois points. Le premier concerne l'aptitude des apprenants symboliques à s'adapter à des données de grandes dimensions, comme la catégorisation de textes. Cette question est en fait sous-jacente tout au long de nos travaux. Le second est relatif à la réduction du vocabulaire. Il s'agit de déterminer ce qu'il est important de donner comme informations à la méthode de réduction pour faciliter l'apprentissage. Le dernier point concerne le lien entre la nature des données textuelles et le comportement des apprenants. Dans leur ensemble, les travaux sur la classification de textes se sont intéressés aux résultats cumulés sans chercher à distinguer les données propices à l'apprentissage des autres.

Notre contribution

Nous avons tout d'abord proposé un cadre assez général pour prendre en compte la majorité des approches de catégorisation de textes existantes. Ce cadre nous a permis de distinguer les deux étapes importantes dans la construction automatique d'un système de catégorisation, la représentation des textes, qui inclut la réduction de l'index original, et l'apprentissage inductif.

Réduction des dimensions

Nous avons établi que la réduction de l'index était un cas particulier de la sélection d'attributs pertinents en apprentissage. Une des distinctions réside dans l'objectif de la sélection. En catégorisation, la sélection est nécessaire pour pouvoir apprendre à l'aide d'algorithmes classiques, tandis qu'en apprentissage, on sélectionne un sous-ensemble d'attributs pour optimiser les classifieurs induits. À la lumière de cette différence, nous avons analysé les différentes approches pour la sélection d'attributs, filtrage et modèle encaspulé, en insistant sur leurs intérêts et limites pour les données textuelles.

Nous avons proposé SCAR un algorithme de réduction symbolique qui agit comme un filtre en amont de l'apprentissage. Cet algorithme se caractérise par la prise en compte de deux types d'informations : une représentation des textes en utilisant uniquement la présence des mots et une partie du présupposé de l'apprenant CHARADE. Nous avons de plus décrit deux autres algorithmes de réduction, RELIEF+ et IM. Le premier est une modification d'un algorithme proposé en apprentissage [KR92], dans le but de prendre en compte la représentation à partir de la présence des mots uniquement et la répartition entre les classes. La deuxième approche, IM, est classique en catégorisation de textes : elle s'appuie sur la présence et l'absence des unités. Nous avons effectué une comparaison a priori de ces méthodes de sélection qui a montré que les méthodes s'appuyant sur une description par l'absence étaient plus caractéristiques que discriminantes, en particulier RELIEF+, tandis que IM mitigeaient les deux aspects.

Le but de la réduction est de permettre l'application de techniques classiques d'apprentissage, tout en cherchant à définir le langage des instances qui optimisent la représentation des exemples au regard de l'algorithme d'apprentissage. Nous avons donc naturellement étudié l'apprentissage réalisé avec CHARADE en fonction des trois méthodes de sélections. Du point de vue des performances des classifieurs appris, la sélection réalisée par SCAR se montre légèrement supérieure. Ainsi, les hypothèses sur les trois algorithmes n'ont pas d'influence frappante sur la performance du système de catégorisation appris. Cependant, l'emploi de la même hypothèse, représentation par la présence, que l'apprentissage entraine une amélioration non négligeable de la complexité réelle de l'apprentissage. Finalement, l'étude de quelques catégories nous a conduit à distinguer l'influence des approches de sélection de manière locale. Les propriétés a priori sur la nature de la sélection effectuée ont été confirmées par le couplage apprentissage et sélection.

Apprentissage et données textuelles

Le deuxième aspect de nos travaux s'est intéressé aux propriétés des techniques d'apprentissage pour la catégorisation. Il existe évidemment un certain nombre de travaux qui comparent différentes approches ou l'approche proposée aux travaux décrits précédemment. Il est cependant très difficile d'évaluer le composant apprentissage de ces travaux. étant donné que la représentation des documents varie selon les approches. Nous nous sommes donc intéressée à comparer différentes approches d'apprentissage en utilisant une même représentation des textes.

Nous avons donc figé la représentation des exemples, en sélectionnant l'index final à l'aide de IM. Nous avons tout d'abord fait évoluer le langage des hypothèses, c'est-à-dire la forme prise par les règles de classification apprises. La manière de construire ces règles reste invariante, à l'aide de l'algorithme CHARADE. Permettre l'emploi de l'absence des mots pour décrire une catégorie, puis chercher à caractériser l'absence d'une catégorie s'est révélé avoir un impact non négligeable sur les performances du système de catégorisation.

Nous avons ensuite étudié l'influence du présupposé d'apprentissage, en particulier en comparant quatre apprenants implémentant une approche distincte pour la généralisation. Nous considérons les résultats obtenus comme un défrichage de la question plutôt que comme une étude approfondie. Nous avons souhaité insisté sur l'impact local de chaque méthode d'apprentissage. Nous soutenons en effet la thèse que les différences entre apprenants n'émergent que de façon locale à un groupe de catégories. Ainsi, nous avons montré qu'ID3 engendrait un classifieur plus précis lorsque la catégorie était illustrée par des documents employant un vocabulaire similaire ou limité.

Limitations de nos travaux

Nos travaux correspondent à une première étude. En cela, ils sont sujets à certaines limites. Tout d'abord, nos conclusions ne sont étayées que par des expérimentations sur la collection Reuters-22173. La généralisation des résultats obtenus à d'autres collections reste en suspens. En particulier, il serait intéressant de voir si les propriétés que nous avons mises à jour se généralisent à des collections de taille plus importantes. Le cas de TREC [Har96] est particulièrement intéressant dans la mesure où les documents employés pour l'évaluation ne proviennent pas forcément de la même source que les documents ayant servi à construire les classifieurs. Cette propriété est d'autant plus critique lorsque l'on met en œuvre une réduction avant l'apprentissage.

Les algorithmes d'apprentissage que nous avons utilisés sont des algorithmes simples. En effet, ils n'intègrent aucune des caractérisques des apprenants récents pour éviter le sur-apprentissage, mais une comparaison avec des travaux récents [CS96] semble indiquer que la différence sur la qualité des apprenants n'est pas frappante[5]. Par ailleurs, nos observations sur la méthode des plus proches voisins, si elles semblent cohérentes avec l'approche proposée par Y. Yang [Yan96a], ne sont que partielles, puisqu'elles ne s'appuient que sur un sous-ensemble de la collection Reuters.

Finalement, nous avons motivé l'absence de validation statistique pour les comparaisons entre méthodes par l'inadéquation des tests statistiques habituellement en apprentissage et en recherche documentaire [Hul93]. Les conditions d'application des tests sont en effet difficile à satisfaire pour la collection Reuters-22173. Cette limitation est atténuée par le fait que nous ne revendiquons pas la supériorité d'un apprenant par rapport à un autre, mais cherchons à déterminer la nature des données sur lesquelles un apprenant réagit mieux.

Perspectives

À court terme, nous envisageons de modifier l'algorithme SCAR pour en faire un algorithme d'apprentissage robuste. En effet, c'est un algorithme inductif que nous avons

5. Cette comparaison n'est pas formelle, dans la mesure où les différentes expériences ont été effectuées sur la même collection, mais avec un découpage en ensemble d'apprentissage et ensemble de test différent.

employé jusqu'ici comme filtre en amont de CHARADE. Pourtant, il a la particularité de ne pas faire appel à une étape de réduction, et met en œuvre une exploration qui tire parti de la nature «creuse» des données textuelles. Nous pensons en effet qu'abandonner la réduction permet de traiter des collections documentaires, où les documents d'apprentissage et de test proviennent de différentes sources[6].

La mise en évidence de liens entre données textuelles et propriétés des apprenants suggère deux directions de recherche. Dans un premier temps, il s'agit de voir si ces liens se généralisent à d'autres collections documentaires. Certains apprenants se prêtent mieux à un type de données qu'à un autre. C'est pourquoi nous envisageons de combiner différents apprenants pour traiter une collection entière. Différentes approches ont déjà été proposées pour la combinaison de classifieurs en recherche documentaire [HPS96, LC96] et pondèrent les résultats obtenus par différents classifieurs pour fournir la réponse finale. Nous souhaitons explorer une voie alternative, proche de la compétition entre méthodes, où le classifieur est choisi en fonction des propriétés des données, comme la taille du vocabulaire ou l'existance de mots à fort pouvoir discriminant.

Du point de vue de l'apprentissage, la catégorisation de textes peut être considérée comme une tâche de classification multi-classe. En effet, plusieurs étiquettes peuvent être affectées à un même document. Or la majorité des techniques d'apprentissage suppose que la classification est une partition et non un recouvrement, c'est-à-dire qu'un exemple appartient à une et une seule classe. Les réseaux de neurones artificiels représentent une des rares approches qui ne possèdent pas cette contrainte. Les systèmes de catégorisation de textes contournent ce problème en transformant l'affectation de n catégories en n problèmes binaires. Ce faisant, ils perdent toute information sur la cooccurence des catégories. Nous estimons une comparaison entre approche binaire et approche multi-classe utile pour évaluer le coût de cette perte d'informations.

Finalement, nous avons adopté une représentation vectorielle s'appuyant sur les mots sans la remettre en cause. De nombreux travaux se sont en effet intéressés à ce problème, et obtiennent de résultats mitigés. L'une des revendications de l'apprentissage symbolique est sa capacité à prendre en compte des connaissances plus ou moins complexes lors de l'apprentissage. Dans le contexte de la catégorisation de textes, nous nous intéressons plus particulièrement aux aspects suivants. Certaines collections disposent de connaissances sur les catégories prédéfinies ; dans le cas de Reuters-22173, ces connaissances sont données sous la forme d'une hiérarchie. Cette caractéristique est à mettre en relation avec la notion de multi-classe mentionnée précédemment. Un deuxième aspect est lié à la structure des textes. Si les exemples sont étiquetés avec des marqueurs de la langue, il peut être possible d'abandonner la représentation vectorielle de textes et de conserver la structure de ces exemples. L'apprentissage symbolique dans des domaines structurés [Zuc96] nous semble une approche possible pour gérer de telles connaissances. Finalement, il semble restrictif de se limiter à la forme graphique (lexicale) des mots en abandonnant toute notion de sens. Une perspective à plus long terme est d'employer des techniques du traitement du langage pour réduire l'index sur des critères sémantiques et non plus calculatoires.

6. L'induction suppose que les données inconnues possèdent des propriétés similaires à celles de données. Un système de catégorisation construit à partir de documents financiers ne pourra pas étiqueter des documents politiques, qu'il fasse ou non appel à une étpae de réduction.

Bibliographie

[AB94] D. W. Aha and R. L. Bankert. Feature selection for case-based classification of cloud types: An empirical comparison. In D. W. Aha, editor, *Case-based Reasoning: Papers from the 1994 Workshop (Technical Report WS-94-01)*, Menlo Park, California, 1994. AAAI Press.

[AB95] D. Aha and R. L. Bankert. A comparative evaluation of sequential feature selection algorithms. In *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, pages 1–7, Ft. Lauderdale, 1995.

[AD91] H. Almuallin and T. G. Dietterich. Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI'91)*, San Jose, CA, 1991. AAAI Press.

[ADW94a] C. Apté, F. Damerau, and S. Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, July 1994.

[ADW94b] C. Apté, F. Damerau, and S. W. Weiss. Towards language independent automated learning of text categorization models. In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Dublin, Ireland, 1994.

[AKA91] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, (6):37–66, 1991.

[Arc95] D. Archambault. *Proposition de réseaux neuro-mimétiques pour des traitements du langage naturel.* PhD thesis, Université d'Orléans, 1995.

[BB63] H. Borko and M. Bernick. Automatic document classification. *Journal of the ACM*, pages 151–161, 1963.

[Ber90] C. Berrut. Indexing medical reports: the RIME approach. *Information Processing and Management*, 26(1):93–109, 1990.

[BFL+88] P. Biebericher, N. Fuhr, G. Lustig, M. Schwantner, and G. Knorz. The automatic indexing system AIR/PHYS – from research to application. In *Proceedings of the Eleventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1988.

[BFOS84] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and regression trees.* Wadsworth International Group, Belmont, California, 1984.

[BG93] F. Bergadano and A. Giordana. Comment guider le processus d'induction avec la théorie du domaine. In Y. Kodratoff, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Apprentissage symbolique – une approche de l'intelligence artificielle – Tome II*, pages 395–411. Cépaduès-Edition, 1993.

[BM96] T. A. H. Bell and A. Moffat. The design of a high performance information filtering system. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Zürich, 1996.

[Bou92] S. Boucheron. *Théorie de l'apprentissage : de l'approche formelle aux enjeux cognitifs*. Hermes, Paris, 1992.

[BP86] C. Berrut and P. Palmer. Solving grammatical ambiguities within a surface syntactic parser for automatic indexing. In *Proceedings of the Ninth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Pisa, Italy, 1986.

[Bun90] W. Buntine. A theory of learning classification rules. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI'90)*, pages 736–742, 1990.

[Car89] J. Carbonell. Paradigms for machine learning. *Artificial Intelligence*, (40):1–9, 1989.

[Car93] C. Cardie. Using decision trees to improve case-based learning. In *Proceedings of the Tenth International Conference on Machine Learning*, Amherst, MA, 1993. Morgan Kaufman.

[CD86] Y. Chiaramella and B. Defude. IOTA : un prototype de système expert en recherche d'information. In *Proceedings of the Sixth International Workshop Expert Systems and Their Applications*, Avignon, 1986.

[Cel90] G. Celeux, editor. *Analyse discriminante sur variables continues*. INRIA, 1990.

[CF94] R. Caruana and D. Freitag. Greedy attribute selection. In *Proceedings of the Eleventh International Conference on Machine Learning*, 1994.

[CH90] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, March 1990.

[Cha82] J. Chaumier. *Analyse et langages documentaires : le traitement linguistique de l'information documentaire*. Entreprise Moderne d'Edition, Paris, nouvelle edition, 1982.

[CKB87] B. Cestnik, I. Kononenko, and I. Bratko. ASSISTANT 86: a knowledge-elicitation tool for sophisticated users. In I. Bratko and N. Lavrač, editors, *Progress in Machine Learning*. Sigma Press, 1987.

[Cla85] W. J. Clancey. Heuristic classification. *Artificial Intelligence*, 27:289–350, 1985.

[CM93] C. Clark and S. Matwin. Using qualitative models to guide inductive learning. In *Proceedings of the Tenth International Conference on Machine Learning*, 1993.

[CMSW92] R. H. Creecy, B. M. Masand, S. J. Smith, and D. L. Waltz. Trading MIPS and memory for knowledge engineering: Classifying census returns on the connection machine. *Communication of the ACM*, (35):45–63, July 1992.

[CN89] P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3:261–284, 1989.

[Coh92] W. Cohen. Compiling prior knowledge into an explicit bias. In *Proceedings of the Ninth International Conference on Machine Learning*, Aberdeen, Scotland, 1992. Morgan Kaufmann.

[Coh95a] W. Cohen. Fast effective rule induction. In *Proceedings of the Twelveth International Conference on Machine Learning*, Lake Tahoe, California, 1995.

[Coh95b] W. Cohen. Learning to classify english text with ILP methods. In De Raedt, L, editor, *Advances in Inductive Logic Programming*. IOS Press, 1995.

[Coh95c] W. Cohen. Text categorization and relational learning. In *Proceedings of the Twelveth International Conference on Machine Learning*, Lake Tahoe, California, 1995.

[Coh96a] W. Cohen. Learning rules that classify e-mail. In *AAAI Spring Series Symposium on Machine Learning and Information Access*, Palo Alto, CA, 1996.

[Coh96b] W. Cohen. Learning trees and rules with set-valued features. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI'96)*, Portland, Oregon, 1996.

[Coo91] W. Cooper. Some inconsistencies and misnomers in probabilistic information retrieval. In *Proceedings of the Fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1991.

[Cro81] W. B. Croft. Document representation in probabilistic models of information retrieval. *Journal of the American Society for Information Science*, (451–457), November 1981.

[Cro83] W. B. Croft. Experiments with representation in a document retrieval system. *Information Technology: Research and Development*, (2):1–21, 1983.

[CS93] K. J. Cherkauer and J. W. Shawlik. Protein structure prediction: Selecting salient features from large candidate pools. In Hunter et al. [HSS93].

[CS96] W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Zürich, Switzerland, 1996.

[Dai94] B. Daille. *Approche mixte pour l'extraction de terminologie : statistique lexicale et filtres linguistiques.* Thèse de doctorat en informatique fondamentale, Université Paris 7, 1994.

[Das90] B. V. Dasarathy, editor. *Nearest Neighbors (NN) Norms: NN pattern classification techniques*, Los Alamitos, CA, 1990. IEEE Computer Society Press.

[DAS91] I. Dagan, A. Atai, and U. Schwall. Two languages are more informative than one. In *Proceedings of the Twenty-ninth Annual Meeting of the ACL*, Berkeley, CA, 1991.

[DDF+90] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and Harshman R. Indexing by latent semantic indexing. *Journal of the American Society for Information Science*, 41(6):391–407, September 1990.

[DFH96] I. Dagan, R. Feldman, and H. Hirsh. Keyword-based browsing and analysis of large document sets. In *Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, NV, April 1996. ISRI Press.

[DG83] M. Dillon and A. S. Gray. Fasit: A fully automatic syntactically based indexing system. *Journal of the American Society for Information Science*, 34(2):99–108, March 1983.

[DH73] R. O. Duda and P. E. Hart. *Pattern classification and scene analysis.* Wiley, New York, NY, 1973.

[Did71] E. Diday. La méthode des nuées dynamiques. *Revue de Statistique Appliquée*, 19(1):19–34, 1971.

[DM81] T. G. Dietterich and R. S. Michalski. Inductive learning of structural descriptions: Evaluation criteria and comparative review of selected methods. *Artificial Intelligence*, (16):257–294, 1981.

[Doa92] J. Doak. An evaluation of feature-selection methods and their application to computer security. Technical Report CSE-92-18, Davis: University of California, Department of Computer Science, 1992.

[Dun93] T. Duning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):62–74, 1993.

[EGWH+91] D. A. Evans, K. Ginther-Webster, M. Hart, R. G. Lefferts, and I. A. Monarch. Automatic indexing using selective nlp and first-order thesauri. In *Actes du Congrès International sur la Recherche d'informations assistée par ordinateur (RIAO-91)*, 1991.

[EMT92] C. Enguehard, P. Malvache, and P. Trigano. Indexation de textes : l'apprentissage de concepts. In *Proceedings of the Thirteenth International Conference on Computational Linguistics (COLING'92)*, 1992.

[Fag87] J. L. Fagan. *Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and Non-Syntactic semantic ambiguities.* PhD thesis, Department of Computer Science, Cornell University, 1987.

[FBY92] W. B. Frakes and R. Baeza-Yates, editors. *Information Retrieval: Data Structures and Algorithms.* Prentice-Hall, 1992.

[FHL+91] N. Fuhr, S. Hartmann, G. Lustig, M. Schwantner, and K. Tzeras. AIR/X — a Rule-Based Multistage Indexing System for Large Subject Fields. In *Actes du Congrès International sur la Recherche d'informations assistée par ordinateur (RIAO-91)*, Barcelona, Spain, 1991.

[FI90] U. M. Fayyad and K. B. Irani. What should be minimized in a decision tree? In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI'90)*, Boston, MA, 1990. AAAI Press.

[FI92] U. M. Fayyad and K. B. Irani. The attribute selection problem in decision tree generation. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI'92)*, San Jose, CA, 1992. AAAI Press.

[Fin95] S. Finch. Partial orders for document representation: A new methodology for combining document features. In *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, WA, 1995.

[Fuc93] C. Fuchs. *Linguistique et Traitements Automatiques des Langues.* Hachette, 1993.

[Fuh89] N. Fuhr. Models for retrieval with probabilistic indexing. *Information Processing and Management*, 25(1):55–72, 1989.

[FWD93] U. Fayyad, N. Weir, and S. Djorgovski. Automated analysis of a large-scale sky survey: the SKICAT system. In *Proceedings of the AAAI Knowledge Discovery in Databases Workshop*, 1993.

[Gan87] J.-G. Ganascia. *AGAPE et CHARADE: deux techniques d'apprentissage symbolique appliquées à la construction de bases de connaissances.* PhD thesis, Université de Paris-Sud (Orsay), 1987.

[Gan92] J.-G. Ganascia. Formalisation algébrique de CHARADE – TDIS: an algebraic formalization. Technical Report 92-32, LAFORIA-IBP, November 1992.

[Gau95] E. Gaussier. *Modèles statistiques et patrons morphosyntaxiques pour l'extraction de lexiques bilingues.* PhD thesis, Université Paris 7, 1995.

[GDP91] M. Gams, M. Drobnič, and M. Petrovšek. Learning from examples — a uniform view. *International Journal of Man-Machine Studies*, 34:49–68, 1991.

[Gol89] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, Mass., 1989.

[Gol96] J. Goldberg. CDM: an approach to learning in text categorization. *International Journal of Artificial Intelligence Tools*, 1996.

[GPH90] J.-G. Ganascia, J.-F. Puget, and N. Helft. A model of behaviour for machine learning. In J. L. Alty and L. I. Mikulich, editors, *Proceedingd of the International Conference on the Industrial Applications of Artificial Inteligence*, Leningrad, 1990.

[HANS90] P. Hayes, P. Andersen, I. Nirenburg, and L. Schmandt. TCS: A Shell for Content-Based Text Categorization. In *Proceeding of the Sixth IEEE CAIA*, pages 321–325, 1990.

[Har91] D. Harman. How effective is suffixing? *Journal of the American Society for Information Science*, 42:7–15, 1991.

[Har94] D. Harman, editor. *Proceedings of the Second Text Retrieval Conference*, 1994.

[Har95] D. Harman, editor. *Proceedings of the Third Text Retrieval Conference*, 1995.

[Har96] D. Harman, editor. *Proceedings of the Fourth Text Retrieval Conference*, 1996.

[Hau88] D. Haussler. Quantifying inductive bias: AI learning algorithms and valiant's learning framework. *Artificial Intelligence*, 36:177–221, 1988.

[Héb91] G. Hébrail. Automatic document discrimination: numeric and symbolic approaches. In E. Diday and Y. Lechevallier, editors, *Symbolic-Numeric Data Analysis and Learning*, pages 117–129, 1991.

[HH96] M. Hearst and H. Hirsh, editors. *AAAI Spring Series Symposium on Machine Learning and Information Access*, number SS-96-05, Palo Alto, CA, March 1996. AAAI Press.

[HL77] A. Henri-Labordère. *Analyse de données – Applications et méthodes partiques*. Masson, 1977.

[Hol75] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.

[Hol93] R. C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–91, 1993.

[HPS96] D. Hull, J. Pedersen, and H. Schütze. Method combination for document filtering. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Zürich, 1996.

[HS92] S. Heath, D. Kasif and S. Salzberg. Learning oblique decision trees. Technical report, Johns Hopkins University, Baltimore, MD, 1992.

[HS94] M. Holsheimer and A. Siebes. Data mining. The search for knowledge in databases. Technical Report CS-R9406, CWI, Amsterdam, January 1994.

[HSP96] D. A. Hull, H. Schütze, and J. O. Pedersen. Document routing as statistical classification. In M. Hearst and H. Hirsh, editors, *AAAI Spring Series Symposium on Machine Learning and Information Access*, Stanford University, Palo Alto, CA, 1996.

[HSS93] L. Hunter, D. Searls, and J. Shavlik, editors. *First International Conference on Intelligent Systems for Molecular Biology*. The AAAI Press, 1993.

[Hul93] D. Hull. Using statistical testing in the evaluation of retrieval performance. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 329–338, 1993.

[Hul94] D. Hull. *Information Retrieval using statistical classification*. PhD thesis, Department of Statistics, Stanford University, November 1994.

[HW73] M. Hollander and D. Wolfe. *Nonparametric Statistical Methods*. Wiley, New York, 1973.

[HW90] P. Hayes and S. Weinstein. CONSTRUE/TIS: a system for content-based indexing of a database of news stories. In *Second Annual Conference on Innovative Applications of Artificial Intelligence*, 1990.

[Jam85] M. James. *Classification algorithms*. Wiley, New York, 1985.

[JC95] Y. Jing and W. B. Croft. An association thesaurus for information retrieval. In *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, 1995. ACM Press.

[JKP94] G. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Machine Learning: Proceedings of the Eleventh International Conference*. Morgan Kaufmann, 1994.

[JR90] P. Jacobs and L. Rau. SCISOR: extracting information from on-line news. *Communications of the ACM*, 33(11):88–97, 1990.

[JR93] P. Jacobs and L. Rau. Innovation in text interpretation. *Artificial Intelligence*, 63:143–191, 1993.

[JR94] C. Jacquemin and J. Royauté. Retrieving terms and their variants in a lexicalized unification-based framework. In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Dublin, 1994.

[KA88] D. Kibler and D. W. Aha. Comparing instance-averaging with instance-filtering learning algorithms. In *Proceedings of the Third European Working Session on Learning*, pages 63–80, Glasgow, Scotland, 1988. Pitman.

[Kad91] C. M. Kadie. Quantifying the value of contructive induction, knowledge, and noise filtering on inductive learning. In *Proceedings of the Eighth International Workshop on Machine Learning*, 1991.

[Ker85] D. Kerkouba. Indexation automatique et aspects structurels du texte. In *Actes du Congrès International sur la Recherche d'informations assistée par ordinateur (RIAO-85)*, Grenoble, France, 1985.

[KFS95] R. D. King, C. Feng, and A. Sutherland. StatLog: Comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence*, (9):289–333, 1995.

[Kit74] J. Kittler. *Mathematical methods of feature selection in pattern recognition.* Cambridge : Department of Engineering, 1974.

[Kit86] J. Kittler. Feature selection and extraction. In T. Y. Young and K.-S. Fu, editors, *Handbook of Pattern Recognition and Image Processing*. Academic Press, 1986.

[KJL+94] R. Kohavi, G. John, R. Long, D. Manley, and K. Pfleger. MLC++: A Machine Learning Library in C++. In *Tools with Artificial Intelligence*, pages 740–743. IEEE Computer Society Press, 1994.

[Koh95] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the International Joint Conference in Artificial Intelligence*, Montréal, 1995.

[Kon94] I. Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In *Proceedings of the 1994 European Cconference on Machine Learning*, 1994.

[KR92] K. Kira and L. Rendell. A practical approach to feature selection. In *Proceedings of the Ninth International Conference on Machine Learning*, 1992.

[KS95] R. Kohavi and D. Sommerfield. Feature selection using the wrapper method: overfitting and dynamic search space topology. In *Proceedings of the First International Conference in Knowledge Discovery and Data Mining*, Montréal, 1995.

[Lan94] P. Langley. Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*, pages 1–5, New Orleans, LA, 1994. AAAI Press.

[LB85] H. Levesque and R. Brachman. A fundamental trade-off in knowledge representation and reasoning. In R. Brachman and H. Levesque, editors, *Readings in Artificial Intelligence*, pages 44–71. Morgan Kaufman, 1985.

[LC94] D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, 1994.

[LC96] L. S. Larkey and W. B. Croft. Combining classifiers in text categorization. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Zürich, 1996.

[Len82] D. Lenat. *Knowledge-based Systems in Artificial Intelligence*. McGraw-Hill, 1982.

[Lew91] D. D. Lewis. Evaluating text categorization. In *Proceedings of Speech and Natural Language Workshop*, Asilomar, 1991.

[Lew92] D. D. Lewis. *Representation and Learning in Information Retrieval.* PhD thesis, Graduate School of the University of Massachusetts, 1992.

[Lew95] D. D. Lewis. Evaluating and optimizing autonomous text classification systems. In *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1995.

[Lew96] D. D. Lewis. The TREC-4 filtering task. In D. K. Harman, editor, *Proceedings of the Fourth Text Retrieval Conference (TREC-4)*, Gaithersburg, MD, 1996. National Institute of Standards and Technology.

[LIT92] P. Langley, W. Iba, and K. Thompson. An analysis of bayesian classifiers. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI'92)*, pages 223–228, San Jose, CA, 1992. AAAI Press.

[Lov68] J. B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31, 1968.

[LPY94] E. D. Liddy, W. Paik, and E. S. Yu. Text categorization for multiple users based on semantic features from a machine-readable dictionary. *ACM Transactions on Information Systems*, 12(3):278–295, July 1994.

[LR94] D. Lewis and M. Ringuette. A comparison of two learning algorithms for text categorization. In *Proceedings of the Symposium on Document Analysis and Information Retrieval*, 1994.

[LS83] P. Langley and H. Simon. Rediscoveing chemistry in BACON.4. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: an Artificial Intelligence Approach*. Morgan Kaufman, 1983.

[LS94a] P. Langley and S. Sage. Induction of selection bayesian classifiers. In *Proceedings of the Tenth Conference on Uncertainty and Artificial Intelligence*, pages 399–406, Seattle, WA, 1994. Morgan Kaufman.

[LS94b] P. Langley and S. Sage. Oblivious decision trees and abstract cases. In *Working notes of the AAAI-94 Workshop on Case-Based Reasoning*, pages 113–117, Seattle, WA, 1994. AAAI Press.

[LSCP96] D. D. Lewis, R. E. Shapire, J. P. Callan, and R. Papka. Training algorithms for linear text classifiers. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Zürich, 1996.

[Luh58] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, Avril 1958.

[MA94] P. M. Murphy and D. W. Aha. UCI repository of machine learning databases, 1994.

[Mar61] M. E. Maron. Automatic indexing: an experimental inquiry. *Journal of the Association for Computing Machinery*, (8):404–417, 1961.

[Mar84] J.F. Marcotorchino. Utilisation des comparaisons par paires en statistique des contingences. Technical Report F-071, IBM-France Scientific Center, 1984.

[Mau91] M. L. Mauldin. Retrieval performance in FERRET. a conceptual information retrieval system. In *Proceedings of the Fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 347–355, 1991.

[MB91] W. Mekaouche and J.-C. Bassano. Multi-experts system for documentary research. In *Actes du Congrès International sur la Recherche d'informations assistée par ordinateur (RIAO-91)*, pages 394–413, Barcelona, 1991.

[MCM83] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors. *Machine Learning: An artificial intelligence approach.* Morgan Kaufmann, Los Altos, California, 1983.

[MF90] S. Muggleton and C. Feng. Efficient induction of logic programs. In *Proceedings of the First International Workshop on Algorithmic Learning Theory*, pages 368–381, Tokyo, Japan, 1990.

[MG95] I. Moulinier and J.-G. Ganascia. Confronting an existing machine learning algorithm to the text categorization task. In *Working notes, IJCAI-95 Workshop on New Approaches to Learning for Natural Language Processing*, Montréal, Canada, 1995.

[Mic83] R. Michalski. A theory and methodology of inductive learning. *Artificial Intelligence*, (20):111–161, 1983.

[Min89a] J. Mingers. An empirical comparison of pruning methods for decision-tree induction. *Machine Learning*, (4):227–243, 1989.

[Min89b] J. Mingers. An empirical comparison of selection measures for decision-tree induction. *Machine Learning*, (3):319–342, 1989.

[Mit82] T. Mitchell. Generalization as search. *Artificial Intelligence*, (18):203–226, 1982.

[MK95] M. Moulet and Y. Kodratoff. Découverte de connaissances dans les bases de données : présentation du problème et état de l'art. In *Actes des cinquièmes journées nationales du PRC-GDR Intelligence Artificielle*. Tekhea Editions, 1995.

[MKS94] S. K. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32, August 1994.

[MKSB93] S. Murthy, S. Kasif, S. Salzberg, and R. Beigel. OC1: Randomized induction of oblique decision trees. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI'93)*, Washington, DC, 1993. AAAI Press.

[ML78] R. S. Michalski and J. B. Larson. Selection of most representative training examples and incremental generation of VL_1 hypotheses: The underlying methodology and the description of programs ESEL and AQ11. Technical Report 867, Univeersity of Illinois, Departement of Computer Science, Urbana, IL, 1978.

[MLW92] B. Masand, G. Linoff, and D. Waltz. Classifying News Stories using Memory Based Reasoning. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Copenhagen, 1992.

[MMHL86] R. S. Michalski, I. Mozetič, J. Hong, and N. Lavrač. The multi-purpose incremental learning system AQ15 and its application to three medical domains. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 1041–1045, Philadelphia, PA, 1986. Morgan Kaufmann.

[MO94] R. Mooney and D. Ourston. Theory refinement combining analytical and empirical methods. *Artificial Intelligence*, 66(2):311–344, 1994.

[Mou96] I. Moulinier. A framework for comparing text categorization systems. In *Working notes of the AAAI Spring symposium on Machine Learning and Information Access*, 1996.

[MRG96] I. Moulinier, G. Raškinis, and J.-G. Ganascia. Text categorization: a symbolic approach. In *Proceedings of the 5th Symposium on Document Analysis and Information Retrieval*, Las Vegas, NV, 1996. ISRI.

[Nat91] B. K. Natarajan. *Machine learning : a theoretical approach.* Morgan Kaufmann, 1991.

[NH93] S. W. Norton and H. Hirsh. Learning DNF via probabilistic evidence combination. In *Proceedings of the Tenth International Conference on Machine Learning*, Amherst, Ma, 1993.

[Nuñ91] M. Nuñez. The use of background knowledge in decision tree induction. *Machine Learning*, 6:231–250, 1991.

[Osw86] J. Oswald. *Théorie de l'information, ou Analyse diacritique des systèmes.* Masson, 1986.

[Pai94] C. D. Paice. An evaluation method for stemming algorithms. In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Dublin, 1994.

[Paz90] M. J. Pazzani. Average case analysis of conjunctive learning algorithms. In *Proceedings of the Seventh International Conference on Machine Learning*, 1990.

[PH90] G. Pagallo and D. Haussler. Boolean feature discovery in empirical learning. *Machine Learning*, (5):71–99, 1990.

[Pic72] C. F. Picard. *Graphes et Questionnaires.* Gauthier-Villars, 1972.

[Pit86] J. Pitrat. *Textes, Ordinateurs et Compréhension.* Eyrolles, 1986.

[Pit94] L. Pitt. Computational learning theory: Introduction and survey. In *Proceedings of the Seventh Annual Conference on Computational Learning Theory (COLT-94)*, New Brunswick, NJ, 1994.

[PK92] M. Pazzani and D. Kibler. The utility of knowledge in inductive learning. *Machine Learning*, (9):57–94, 1992.

[Por80] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

[Qui83] J. R. Quinlan. Learning efficient classification procedures and their application to chess end games. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An artificial intelligence approach.* Morgan Kaufmann, Los Altos, California, 1983.

[Qui86a] J. R. Quinlan. The effect of noise on concept learning. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An artificial intelligence approach (Vol. 2).* Morgan Kaufmann, Los Altos, California, 1986.

[Qui86b] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[Qui90] J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.

[Qui92] J. R. Quinlan. *C4.5 Programs in Machine Learning.* Morgan Kaufmann, 1992.

[Raš95] G. Raškinis. Les algorithmes de sélection d'attributs pour le système d'apprentissage charade appliqué à la catégorisation de documents. Master's thesis, Université Pierre et Marie Curie, 1995.

[Ren88] L. Rendell. Learning hard concepts. In *Proceedings of the Third European Working Session on Learning*, pages 177–200, Glasgow, Scotland, 1988. Pitman.

[Riv87] R. L. Rivest. Learning decision lists. *Machine Learning*, 2:229–246, 1987.

[RJ91] L. Rau and P. Jacobs. Creating segmented databases from free text for text retrieval. In *Proceedings of the Fourteenth Annual International ACM*

SIGIR Conference on Research and Development in Information Retrieval, pages 337–346, 1991.

[RL94] E. Riloff and W. Lehnert. Information extraction as a basis for high-precision text classification. *ACM Transactions on Information Systems*, 12(3):296–333, July 1994.

[Roc71] J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART retrieval system: Experiments in Automatic Document Processing*. Prentice–Hall, 1971.

[Sab89] G. Sabah. *L'intelligence artificielle et le langage - processus de compréhension*, volume 2. Hermes, Paris, 1989.

[Sal68] G. Salton. *Automatic Information organization and retrieval.* McGraw-Hill, 1968.

[Sal89] G. Salton. *Automatic Text Processing: the Transformation, Analysis and Retrieval of Information by Computer.* Addison Wesley, 1989.

[San94] M. Sanderson. The Reuters Collection. In *BCS IRSG'94*, 1994.

[Sap90a] G. Saporta. Introduction à la discrimination: problématique et méthodes. In G. Celeux, editor, *Analyse discriminante sur variables continues*, Collection Didactique, pages 5–13. INRIA, 1990.

[Sap90b] G. Saporta. *Probabilités, analyse des données et statistique.* Editions Technip, Paris, 1990.

[Sar95] T. Saracevic. Evaluation of evaluation in information retrieval. In *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 138–146, Seattle, USA, 1995.

[SCF$^+$95] S. Salzberg, R. Chandar, H. Ford, S. Murthy, and R. White. Decision trees for automated identification of cosmic ray hits in Hubble space telescope images. *Publications of the Astronomical Society of the Pacific*, (107):1–10, May 1995.

[Sch93] J. C. Schlimmer. Efficiently inducing determinations: A complete and systematic search algorithm that uses optimal pruning. In *Proceedings of the Tenth International Conference on Machine Learning*, Amherst, MA, 1993. Morgan Kaufmann.

[Sha49] C. E. Shannon. *A mathematical theory of communication.* University of Illinois Press, 1949.

[Sho76] E. Shortliffe. *Computer based medical consultations:* MYCIN. America Elsevier, 1976.

[SHP95] H. Schütze, D. A. Hull, and J. O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 229–238, Seattle, WA, 1995.

[SJ71] K. Sparck Jones. *Automatic Keyword Classification for Information Retrieval.* Butterworths, 1971.

[SJ72] K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[Ska94] D. Skalak. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 293–301, New Brunswick, NJ, 1994. Morgan Kauffmann.

[SM83] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill Book Company, New-York, 1983.

[Sme95] A. F. Smeaton. Low level language processing for large scale information retrieval: What techniques actually work. In *Proceedings of the workshop Terminology, Information Retrieval and Linguistics*, Rome, 1995. CNR.

[Ste65] M. E. Stevens. Automatic indexing: A state-of-the-art report. Technical report, U.S. Governement Printing Office, Washington DC (USA), 1965.

[Sun91] B. Sundheim, editor. *Proceedings of the Third Message Understanding Conference (MUC-3).* Morgan Kaufman, 1991.

[SvR88] A. Smeaton and C. J. van Rijsbergen. Experiments on incorporating syntactic processing of user queries into a document retrieval strategy. In *Proceedings of the Eleventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 31–51, 1988.

[Tho96a] J. Thomas. *Vers l'intégration de l'apprentissage symbolique automatique et de l'acquisition des connaissances basée sur les modèles : le système Enigme.* PhD thesis, Université Pierre et Marie Curie (Paris VI), 1996.

[Tho96b] J. Thorsten. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical Report CMU-CS-96-118, Carnegie Mellon University, 1996.

[TS94] G. G. Towell and J. W. Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, 70(1–2):119–165, 1994.

[Tur59] A. Turing. Computing machinery and intelligence. In *Mind 59*. 1959.

[Tur90] H. Turtle. *Inference Networks for Document Retrieval.* PhD thesis, Computer and Information Science Department, University of Massachussetts, October 1990.

[TWK94] T. Townsend-Weber and D. Kibler. Instance-based prediction of continuous values. In *Working Notes of the AAAI94 Workshop on Case-Based Reasoning*, Seattle, WA, 1994. AAAI Press.

[Utg89] P. E. Utgoff. Incremental induction of decision trees. *Machine Learning*, (4):161–186, 1989.

[Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.

[vR79] G. J. van Rijsbergen. *Information Retrieval.* Butterworths, 1979.

[VS87] N. Vleduts-Stokolov. Concept recognition in an automatic text-processing system for the life sciences. *Journal of the American Society for Information Science*, 38:269–287, 1987.

[Wie95] E. Wiener. A neural network approach to topic spotting in text. Master's thesis, Department of Computer Science, University of Colorado, 1995.

[WM94] J. Wnek and R. Michalski. Hypothesis-driven constructive induction in AQ17-HCI: A method and experiments. *Machine Learning*, 14:139–168, 1994.

[WPW95] E. Wiener, J. O. Pedersen, and A. S. Weigend. A neural network approach to topic spotting. In *Proceedings of the Fourth Symposium on Document Analysis and Information Retrieval (SDAIR'95)*, 1995.

[Yan94] Y. Yang. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994.

[Yan95] Y. Yang. Noise reduction in a statistical approach to text categorization. In *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1995.

[Yan96a] Y. Yang. An application of statistical learning to news story categorization. Technical report, Carnegie Mellon University, 1996.

[Yan96b] Y. Yang. Sampling strategies and learning efficiency in text categorization. In *AAAI Spring Series Symposium on Machine Learning and Information Access*, 1996.

[YW96] Y. Yang and W. J. Wilbur. Using corpus statistics to remove redundant words in text categorization. *Journal of the American Society of Information Science*, 1996.

[Zuc96] J.-D. Zucker. *Appariements et changements de représentation pour l'apprentissage symbolique.* PhD thesis, Université Pierre et Marie Curie (Paris VI), 1996.

Index

Annexe A

Fiche technique sur l'information mutuelle

À la fin des années 40, C. E. Shannon a défini la théorie de l'information pour donner une représentation mathématique des phénomènes de communication [Sha49]. La théorie de l'information a par la suite été l'objet de nombreux travaux (voir par exemple [Osw86]).

L'information d'un événement est d'autant plus grande que l'effet de surprise est fort, ce qui permet de lier l'information d'un événement et sa probabilité.

L'une des propriétés requises pour l'information est l'additivité : en effet, l'information apportée par deux événements indépendants est la somme des informations de chaque élément.

Si f est une fonction associée à la mesure d'information, f possède les propriétés suivantes :

- $f : [0,1] \to \mathbb{R}^+$
- $\forall p_1, p_2 \quad f(p_1 \times p_2) = f(p_1) + f(p_2)$
- f est décroissante et continue

La fonction $f(p) = -\log_2(p)$ vérifie ces propriétés et est communément associée à la mesure d'information.

Information d'une variable

L'information I contenue par une variable A ayant n modalités, de probabilités respectives $p_1, \cdots, p_n$ est définie comme la moyenne de l'information apportée par les différentes modalités :

$$I = -\sum_{i=1}^{i=n} p_i \times \log_2 p_i$$

Cette quantité I est aussi appelée entropie de Shannon de la variable A [Sha49] et notée $H(A)$.

Information conditionnée

Soient deux variables A et B. A est une variable ayant n modalités, de probabilités respectives $p_1, \cdots, p_n$, B prend m valeurs, avec les probabilités respectives $q_1, \cdots, q_m$. La quantité d'information apportée par la variable A lorsque B est connue est notée $H(A|B)$.

C'est la moyenne des quantités d'informations apportée par la variable A, connaissances les différentes modalités de B. Notons $H_j(A)$ l'information de A conditionnellement à la j$^{\text{e}}$ modalité de la variable B. Elle est calculée à partir du sous-ensemble des exemples ayant la modalité j de la variable B. On a alors : L'information de A conditionnée par la variable B, notée $H(A|B)$, est la moyenne des informations apportées par la variable A, connaissant les différentes modalités de B :

$$H(A|B) = \sum_{j=1}^{j=m} q_j \times H_j(A).$$

Information mutuelle entre deux variables

L'information mutuelle entre les variables A et B permet de mesurer l'information de la variable A contenue dans la variable B. C'est la portion d'information commune aux variables A et B. Elle est définie par :

$$I(A, B) = H(A) - H(A|B)$$

et possède les propriétés suivantes :

- elle est symétrique
- elle est toujours positive
- elle est minimale lorsque les variables A et B sont indépendantes
- elle est maximale lorsque les variables A et B sont liées (la connaissance de la modalité de B entraîne la connaissance de la modalité de A)

Les notions d'information conditionnée et d'informations mutuelles peuvent être étendues à plus de deux variables.

Annexe B

Algorithmes en pseudo-code de quelques apprenants

B.1 Implémentation de l'étoile : AQ

Version simplifiée de AQ

Soit E^+ *l'ensemble des exemples positifs du concept C*
Soit E^- *l'ensemble des exemples négatifs du concept C*
Soit Desc la description vide
Tant que Desc ne couvre pas tous les exemples de E^+

- *sélectionner un exemple positif non couvert par Desc, appelé Germe*
- *Étoile = étoile(Germe,* E^-*)*
- *choisir la meilleure conjonction Meilleur de Étoile*
- *Desc = Desc* $\vee$ *Meilleur*

étoile(Germe, E)

Soit Étoile l'ensemble contenant la conjonction vide
Tant qu'une description de Étoile couvre un exemple de E

- *choisir un exemple* e^- *couvert par une conjonction de Étoile*
- *spécialiser les conjonctions de Étoile pour exclure* e^- *:*
  - *trouver l'ensemble des littéraux Lit qui couvrent Germe et ne couvrent pas* e^-
  - *spécialiser Étoile avec ces littéraux, i.e Étoile* $= \{x \wedge y | x \in$ *Étoile*$, y \in$ *Lit*$\}$
  - *éliminer de Étoile toutes les conjonctions plus spécifiques que d'autres dans Étoile*
- *Tant que la taille de Étoile est supérieur à un seuil MaxÉtoile*
  - *enlever la plus mauvaise conjonction de Étoile*

B.2 CHARADE

VERSION SIMPLIFIÉE DE CHARADE

Soit E l'ensemble des exemples

Soit Desc l'ensemble des descripteurs (ceux apparaissant dans les exemples et ceux obtenus par saturation)

Soit k la profondeur, λ la redondance et f le fractionnement

Soit Règles, un ensemble initialement vide

Soit File, une file initialisée avec la description vide

Tant que File n'est pas vide et qu'il existe un exemple qui n'est pas couvert λ fois

- *Soit Elt, le premier élément de File (Elt est une description)*
- *Soit $E' = \gamma(Elt)$*
- *Si tous les exemples de E' ont la même classe c alors*
  - *Si $|E'|/|\gamma(c)| \geq f$ alors*
    - *ajouter la règle Elt $\rightarrow$ c à Règles*
    - *incrémenter la couverture des exemples de E'*
- *Sinon File=spécialiseNœud(Elt,File)*

spécialiseNœud(Elt,File)

- *FileAux = $\{Elt \wedge x | x \in Desc\}$*
- *Retirer de FileAux tous les éléments déjà dans File ou incohérents*[1]
- *Pour chaque élément d de FileAux*
  - *Si le nombre de descripteurs de d est inférieur à k et si d couvre des exemples dont la couverture n'atteint pas λ et s'il existe au moins une classe c telle que $|\gamma(d)|/|\gamma(c)| \geq f$ alors*
    - *enfiler d à la fin de File*
- *Retourner File*

1. Un exemple d'élément incohérent est la description prime = v $\wedge$ prime =f.

Annexe C

Liste des catégories officielles

Notons que toutes les matières premières sont notées par leur nom. La catégorie correspond en fait à différents types d'information comme le cours ou la production de ces matières premières. Il en est de même pour les monnaies.

earn	bénéfices des entreprises/rapports prévisionnels sur les bénéfices
acq	fusions/achats d'entreprises
money-fx	cours monétaires
grain	céréales
cocoa	cacao
gas	essence
rice	riz
silver	argent
gold	or
lead	plomb
copper	cuivre
reserves	réserves
rubber	caoutchouc
dlr	dollar
trade	commerce (extérieur ?)
yen	yen
interest	taux d'intérêts
coffee	café
sugar	sucre
dmk	mark allemand
fuel	combustible/carburant
sunseed	graines de tournesol
soybean	graines de soja
palm-oil	huile de palme
wheat	blé

corn	maïs
iron-steel	fer, acier
barley	orge
cotton	coton
carcass	carcasse (?)
jobs	chômage
zinc	zinc
gnp	produit national brut
oilseed	graines oléagineuses
veg-oil	huile végétale
ship	envoi par mer
cpi	indice des prix à la consommation
crude	pétrole brut
retail	vente de détail
ipi	indice de la production industrielle
money-supply	fonds, ressources
soy-oil	huile de soja

Annexe D

Ajout de connaissances lors de l'apprentissage

Un attribut artificiel : la taille des textes

Comme l'ont montré les chapitres 7 à 9, certaines catégories sont plus difficiles à caractériser que d'autres. Plus particulièrement, la majorité des textes courts (comprenant une ou deux phrases) est assez difficile à discriminer. Forte de cette observation, nous avons introduit un attribut artificiel relatif à la taille des dépêches. Cet attribut *size* est un attribut ordonné qui prend trois valeurs *petit*, *moyen* et *grand*. Nous considérons comme *petit* un texte comprenant moins de 10 unités, comme *moyen* un texte comprenant entre 10 et 80 unités et comme *grand* un texte comportant plus de 80 unités.

Les résultats que nous avons obtenus montrent que les apprenants symboliques CHARADE et ID3 utilisent l'information convoyée par cet attribut pour discriminer les catégories. Le gain dans la mesure F_1 est donné à la table D.1. Nous remarquons un gain plus sensible pour les différentes versions de CHARADE que pour ID3. En fait, CHARADE utilise l'attribut *size* dans de nombreux cas, car les unités retenues par IM mutuelle ne suffisent pas à discriminer les exemples. L'attribut *size* n'est utilisé par ID3 que pour un nombre limité de catégories. En effet, toutes les unités ne sont pas utilisées lors de l'apprentissage.

Si nous étudions le comportement des classifieurs par catégorie, nous voyons que les classifieurs engendrés par CHARADEP effectuent plus de déductions lorsque l'attribut

Apprenant	*size* ?	F_1
CHARADEP	non	0,732
CHARADEP	oui	0,763
CHARADEDL	non	0,783
CHARADEDL	oui	0,790
ID3	non	0,776
ID3	oui	0,780
ID3/10	non	0,782
ID3/10	oui	0,785

TAB. D.1 - *Performances comparées des classifieurs appris avec l'attribut size et sans cet attribut. Le nombre d'attributs est 75.*

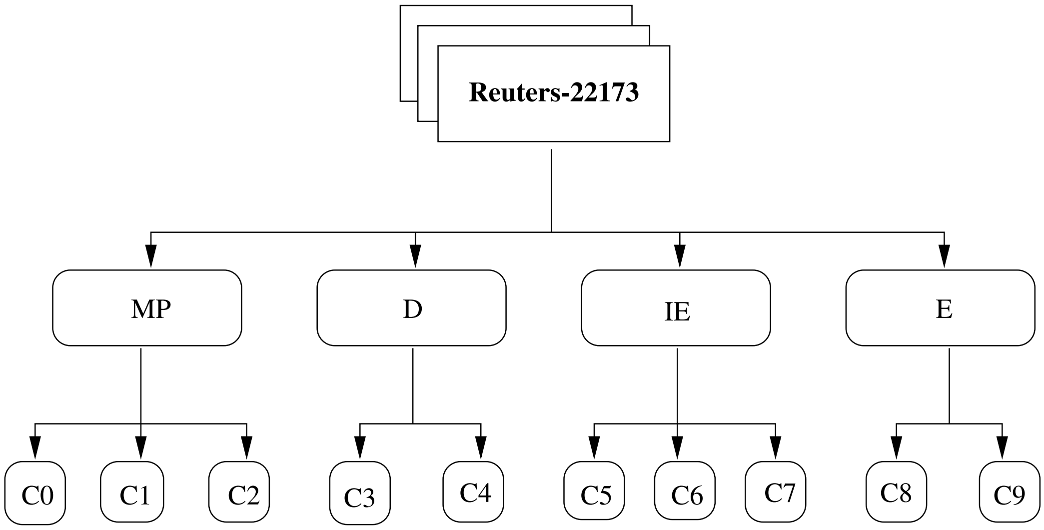


FIG. D.1 - *Structuration de l'apprentissage*

size est présent ; ces déductions ne sont cependant pas toujours correctes, ce qui entraîne une augmentation du rappel mais une stagnation de la précision. Le comportement des classifieurs induits par CHARADEDL est semblable, mais ils ont tendance à diminuer le nombre d'erreurs effectuées. Les arbres construits par ID3, quels que soient les paramètres, ne tirent avantage de l'attribut *size* que pour limiter le nombre d'erreurs. Cet attribut apparaît souvent proche de la racine et conduit à des arbres plus complexes.

Hiérarchies de catégories

À aucun moment, nous n'avons fait intervenir de connaissances du domaine dans l'apprentissage. Dans ce cas, pourquoi promouvoir l'apprentissage symbolique? La transformation d'un problème à n catégories en n sous-problèmes à une catégorie, si elle est nécessaire pour réduire la complexité du traitement, est restrictive dans la mesure où l'on perd des informations sur les catégories qui sont fréquemment affectées en même temps.

Nous nous intéressons à présent à utiliser une hiérarchie définie sur les catégories pour structurer l'apprentissage. La collection Reuters-22173 se prête bien à une telle étude, puisque les catégories sont regroupées en une hiérarchie grossière comprenant 4 grands groupes : les devises, les matières premières, les indicateurs économiques et les entreprises. On peut imaginer deux utilisations de cette hiérarchie. La première est de créer un unique attribut hiérarchique qui peut prendre différentes valeurs de la hiérarchie. Cette approche est cependant coûteuse et difficile à mettre en œuvre lorsqu'un même exemple peut avoir plusieurs catégories. Nous avons opté pour la deuxième méthode qui consiste à structurer l'apprentissage en fonction de cette hiérarchie. Dans un premier temps, nous cherchons à distinguer les quatre grands groupes. Puis, dans un groupe donné, nous cherchons ensuite à caractériser les catégories individuelles (cf. figure D.1).

Les expérimentations ont été rendues possibles grâce au logiciel ENIGME [Tho96a] avec lequel est interfacé CHARADE. Les exemples sont représentés avec un vocabulaire différent pour chaque groupe, puis à nouveau décrits par des unités différentes au sein de chaque groupe pour les différentes catégories. Les résultats obtenus ne sont pas satisfaisants. La reconnaissance des groupes de catégories est très bonne (le F_1 vaut 90%), mais cela ne suffit pas. En effet, CHARADE ne parvient pas mieux à apprendre des classifieurs locaux à chaque groupe que des classifieurs sans hiérarchie. La performance du système global est de 54% avec CHARADEP.

Pourquoi les classifieurs appris ne sont-ils pas plus performants? En effet, les travaux décrits dans [WPW95] montrent qu'une structuration hiérarchique de réseaux de neurones atteint des résultats meilleurs qu'un réseau plat. Nous voyons deux motifs à notre échec. Tout d'abord, les auteurs de [WPW95] définissent leur propre regroupement des catégories, en séparant les matières premières en plusieurs groupes, par exemple. Mais le point le plus important nous apparaît être une faille dans la représentation des exemples. En effet, les unités sélectionnées pour représenter les différentes catégories localement à un groupe ne sont pas très différentes. Une approche différente de la réduction semble donc nécessaire.

Finalement, certaines collections comme OSHUMED [LSCP96] ou celle utilisée pour le système décrit dans [FHL+91] disposent d'un thesaurus et d'une directive pour l'indexation, qui favorise la catégorie la plus spécifique en cas d'ambiguïté. Cette directive est délicate à prendre en compte de manière directe lors de l'apprentissage, puisque l'objectif de la majorité des apprenants symboliques est de trouver une expression la plus générale possible.

Apprentissage multi-classe

L'information contenue dans les hiérarchies peut être complétée lors de l'apprentissage par la connaissance des catégories qui cooccurrent régulièrement. En effet, la catégorisation de textes correspond à un problème de classification multi-classe, dans le sens où plusieurs catégories peuvent être affectées à un même texte. L'attitude adoptée pour prendre en compte cette caractéristique est de transformer la tâche : on construit n classifieurs binaires, lorsqu'il existe n catégories possibles à affecter. Une telle approche permet d'employer des algorithmes d'apprentissage classiques, comme ID3 ou NaiveBayes, qui s'appuient sur des classes sans recouvrement des exemples. De plus chaque exemple ne peut appartenir qu'à une seule classe. Cependant, considérer la catégorisation comme n problèmes indépendants entraîne une perte d'informations.

Prenons à nouveau l'exemple de la collection Reuters-22173. Certaines catégories sont toujours affectées à des documents en conjonction avec d'autres, par exemple la catégorie *wheat* (blé). Cette catégorie est de plus intéressante dans la mesure où elle est affectée majoritairement avec deux autres catégories *grain* (céréales) et *corn* (maïs). En particulier, si l'on décide d'affecter la catégorie *wheat* à une dépêche, la probabilité d'affecter aussi la catégorie *grain* est très forte. À l'opposé, il serait utile de pouvoir caractériser les dépêches qui possèdent la catégorie *grain* mais pas la catégorie *wheat*.

En théorie, l'apprenant CHARADE se prête bien à évaluer l'intérêt d'affecter plusieurs catégories en même temps. En pratique, la complexité de l'apprentissage augmente, puisque le nombre de descripteurs augmente rapidement dès que l'on veut décrire plus d'une catégorie. Nous avons mené des expérimentations préliminaires sur quelques couples de catégories a priori intéressantes, comme *gold* (or) et *silver* (argent), *wheat* et *grain* ou encore *dlr* (dollar) et *money-fx* (cours monétaires). Les résultats obtenus jusqu'ici sont peu concluants : en effet CHARADEP ne parvient généralement pas à engendrer des règles concluant sur les deux catégories, les exemples couverts par ces règles étant en général couverts par des règles plus générales. L'une des raisons que nous avançons pour expliquer cet échec partiel est l'absence d'un nombre suffisant d'exemples pour obtenir des généralisations fiables.

CHARADE n'est pas la seule méthode qui permette de prendre en compte les multiples catégories : les réseaux de neurones en sont un autre exemple, limités comme CHARADE par leur complexité lorsque le nombre d'entrées augmente. Cependant, les réseaux sont encore plus sensibles que CHARADE au manque de données lors de l'apprentissage. Les méthodes des plus proches voisins semblent apporter une solution à ce problème de multiples catégories comme l'ont montré les travaux de [Yan94, LC96].