ECOLE DES MINES DE PARIS

Collège doctoral

N° attribué par la bibliothèque

|__|__|__|__|__|__|__|__|__|__|

T H E S E

Pour obtenir le grade de

Docteur de l'Ecole des Mines de Paris

Spécialité « Informatique Temps Réel, Robotique et Automatique »

Présentée et soutenue publiquement

Par

CHARLES Christophe

17 décembre 2004

SearchXQ : une méthode d'aide à la navigation fondée sur Ω-means, algorithme de classification non-supervisée. Application sur un corpus juridique Français

Jury

M. GIRARDOT Jean-Jacques	Rapporteur
M. KAISER Daniel	Rapporteur
M. CONSTANT Patrick	Examinateur
M. MAHL Robert	Directeur de thèse
M. JOUVELOT Pierre	Examinateur

Remerciements

Je tiens à remercier tout particulièrement Robert Mahl, Directeur du Centre de Recherche en Informatique de l'Ecole des Mines de Paris, pour m'avoir accueilli au sein de son équipe. Je le remercie pour la liberté qu'il a su me laisser tout en me transmettant sa conception de la gestion documentaire.

Je remercie Jean-Jacques Girardot et Daniel Kayser d'avoir accepté d'être les rapporteurs de ce travail. Mes remerciements vont une fois encore à Daniel Kayser pour avoir accepté de présider le jury.

Je tiens à remercier Patrick Constant d'avoir eu l'amabilité de faire partie du jury. Mes remerciements vont également à l'équipe de la société Systal pour leur apport matériel.

Je remercie Pierre Jouvelot pour sa disponibilité et son enthousiasme ainsi que pour ses conseils, commentaires et critiques sur le manuscrit. Je suis très heureux qu'il fasse partie de mon jury.

Merci à toute l'équipe du Centre de Recherche en Informatique pour leur amitié, leur aide et leur soutien. Ces années passées au sein de ce centre furent très agréables.

Un merci tout particulier à Gérard Barbottin pour la relecture et la contribution à l'amélioration de ce manuscrit.

Je remercie enfin tous ceux qui m'ont soutenu et qui ont du faire preuve de patience durant la phase rédactionnelle.

Table des matières

Glossaire

Affixe. Soit un suffixe, soit un infixe ou soit un préfixe.

Bruit (*Noise*). Ensemble de documents non pertinents qui sont retournés en réponse à une requête dans un système documentaire.

Bigramme. (2-gramme) voir N-grammes.

Centroïde (*Centroid*). Représentation d'une classe par une combinaison linéaire d'éléments.

Classe (*Cluster*). Ensemble de documents (ou de segments) qui présentent des caractéristiques communes.

Classification (*Clustering*). Opération qui consiste à regrouper des documents (ou des segments) ayant des caractéristiques communes.

Collection. Voir corpus.

Corpus (Corpus, Collection). Ensemble de textes qui peuvent posséder des caractéristiques linguistiques communes.

Document. Entité pouvant contenir exclusivement ou de façon combinatoire du texte (document textuel), des images, des sons et des vidéos. Dans cette thèse, un document est exclusivement textuel.

Etiquette. Voir Thème.

Fichier inversé. Structure des données qui permet d'en améliorer l'accès à travers un système de RI.

Forme canonique. Voir Lemme.

Forme fléchie. Variation d'un mot : formes conjuguées, marques du pluriel ou du féminin.

Hapax. Terme qui n'apparaît qu'une seule fois dans un corpus.

Homonymie. Caractère des mots qui ont la même forme graphique mais un sens différent.

Hyperonyme. Se dit d'un terme dont le sens inclut le sens d'autres termes.

Lemme. Racine grammaticale des formes fléchies d'un terme (on parle parfois des marques de flexion d'un terme).

Lemmatisation (*Stemming*). Opération qui consiste à réduire un terme en lemme.

Médoïde (*Médoïd*). Représentation d'une classe par un nombre d'éléments.

Modèle vectoriel (*Vector Space Model*). Modèle qui représente les documents et les requêtes par un vecteur de termes.

Morphème. Unité minimale de signification.

Morphologie dérivationnelle. Etude des mots qui ont une racine commune mais qui différent soit par un suffixe, soit par un préfixe. Ces mots ont un sens différent.

Morphologie flexionnelle. Etude des formes fléchies d'un mot.

Mot vide (*Stopword*). Se dit des mots trop fréquents dans la base documentaire. Ces mots ne sont pas pris en compte dans une phase d'indexation ou de représentation des documents. Généralement, il s'agit des déterminants, articles, pronoms, conjonctions, etc.

N-grammes. N lettres successives d'un mot.

Occurrence. Se dit d'un terme qui apparaît dans un texte.

Partition. Groupe non vide de documents.

Pertinence. Ensemble de documents pertinents qui sont retournés en réponse à une requête dans un système documentaire.

Polysémie (*Polysemy*). Propriété d'un terme qui présente plusieurs sens.

Précision. Mesure la capacité d'un système de recherche d'information à retrouver uniquement l'ensemble des documents pertinents en réponse à une requête.

Rappel (*Recall*). Rapport entre le nombre de documents pertinents retournés en réponse à une requête et le nombre total de documents pertinents contenus dans la base documentaire pour la requête.

RI. Recherche d'information

Segment. Une partie d'un texte. Un segment peut correspondre à un ensemble de mots, une phrase, un paragraphe, une section de taille fixe, etc.

Silence. Ensemble de documents pertinents non retournés en réponse à une requête dans un système documentaire.

Similarité. Mesure qui quantifie la ressemblance entre deux documents.

Singleton. Se dit d'un ensemble ne contenant qu'un seul élément.

Synonymie. Relation entre des mots qui ont un sens très voisin.

Syntagme. Groupe d'éléments formant une unité dans une organisation hiérarchisée.

Terme. Désigne un mot ou un groupe de mots.

Thème (*Topic*). Désigne le syntagme caractérisant une classe.

Chapitre 1

Introduction

1.1 Contexte et objectif

L'explosion du nombre de documents disponibles sur Internet rend nécessaire le développement d'outils de recherche d'information de plus en plus performants. Si en 1993, seules quelques milliers de pages étaient disponibles, en 2003 plus de deux milliards de pages l'étaient, et aujourd'hui on note que Google recense plus de 4 milliards de documents en tout genre : html, xml, pdf, doc, etc. De plus, cette croissance, exponentielle, a été constatée [Bourdoncle, 1999] depuis plusieurs années, ce qui ne fera qu'amplifier le phénomène. Ainsi, la problématique majeure est de savoir comment retrouver l'information dans ce flux de données.
Un autre paramètre a évolué également ; c'est l'utilisateur : Internet ne s'adresse plus uniquement, par exemple, à des professionnels de la documentation, à des scientifiques ou encore à des universitaires, qui connaissent parfaitement à la fois leur base de documents et les mécanismes des outils de recherche. Aujourd'hui, Internet s'adresse au grand public et donc son contenu doit être facilement accessible, en termes de recherche.

Si un des domaines doit être particulièrement accessible, c'est bien le domaine juridique. En effet, selon la maxime « Nul n'est censé ignorer la loi », toute personne doit pouvoir accéder aux différents textes du droit. Cependant, encore faut-il que cela soit possible. Depuis plusieurs années, l'informatique, et plus précisément le domaine de l'ingénierie des connaissances, a pris une place importante dans la problématique de l'accès au droit. En effet, à l'image d'Internet, les bases de données juridiques et autres ressources de types terminologiques par exemple ont augmenté de volume de façon significative. De plus, la demande des internautes n'est pas en reste. Ainsi, Internet grâce à ses capacités de diffusion et d'accès à l'information devient un acteur majeur de l'accès au droit. L'instauration d'un service public de diffusion du droit par Internet (Décret n° 2002-1064 du 7 août 2002) en est un exemple.

Dès lors qu'un nombre de documents relatifs au droit français (et au droit européen) est mis à disposition gratuitement sur Internet, il faut mettre en œuvre des outils de recherche qui permettront de trouver facilement un document. En effet, la mise à disposition gratuite de documents n'est utile que si toute personne est capable de les retrouver. Si nul ne peut se soustraire à la loi, toute personne doit être en mesure d'accéder facilement aux textes.

L'accès à l'information sur Internet se traduit généralement par l'utilisation de moteurs de recherche. Ces moteurs de recherche nécessitent, pour la plupart, une requête en entrée et donnent en sortie une liste de documents triés, suivant un critère de pertinence qui leur est souvent propre. Si cette approche comble, en principe, les besoins de l'utilisateur, elle présente des lacunes dès lors qu'il s'agit de formuler correctement une requête ou de reformuler une requête après un premier retour insatisfaisant. En effet, si la requête n'est pas bonne, il est impossible de trouver les documents attendus.

Cette thèse s'inscrit ainsi dans cette problématique de formulation de requête. Nos travaux s'adressent à une catégorie d'utilisateurs qui ne savent pas comment formuler correctement une requête pour trouver l'information désirée. Dans ce cadre, nous avons mis en œuvre une méthode d'aide à la navigation qui repose sur une expansion successive de la requête, par une sélection de termes proposés. Cette approche de l'aide à la navigation qui consiste à « catégoriser dynamiquement » les résultats a été abordée par Bourdoncle [Bourdoncle, 1997] à travers le moteur Altavista. Elle nécessite de la part de l'utilisateur un effort supplémentaire mais modéré. Cet effort est largement récompensé par le fait que l'on peut disposer facilement d'un ensemble de termes susceptibles d'orienter correctement la recherche sans manipulation de documents.

1.2 Organisation de la thèse

Dans le chapitre 2, nous présentons un système classique de recherche d'information où l'utilisateur reste passif entre la requête et la présentation des résultats. Ce type de système est le plus répandu sur Internet. Un système de recherche d'information (RI) est fondé sur une succession d'étapes rappelées dans le chapitre à travers un ensemble de bases théoriques et de méthodes statistiques les plus couramment utilisées. D'autres types de système existent, qui demandent à l'utilisateur un effort supplémentaire (l'utilisateur est alors actif). Ces systèmes sont des aides à la recherche d'information. Ils diffèrent également dans la présentation des résultats (interfaces) et dans la navigation. Ils sont fondés sur le principe que l'utilisateur rencontre souvent des problèmes pour formuler son besoin.

Le chapitre 3 développe un état de l'art des méthodes de classification qui sont pour la plupart classiques dans le domaine. La classification est un processus qui permet d'organiser un ensemble de données en classes cohérentes ou homogènes. Elle s'applique, a priori, sur n'importe quel type de données : tableau de contingence, tableau de distances, etc. La

classification se déroule, dans la plupart des cas, en trois étapes, et à l'aide de quelques paramètres indispensables : mesure de ressemblance, structure de la classification et type d'algorithme. Il existe plusieurs catégories d'algorithmes de classification dont les deux plus utilisées sont les méthodes de partitionnement et les méthodes de classification hiérarchique. D'autres catégories existent, telles que les modèles probabilistes ou des modèles utilisant les liens hypertextes, et donc axées uniquement sur les documents Web. Ce chapitre nous permet de choisir un type de méthode, compte tenu de notre problématique.

Dans le chapitre 4, nous décrivons la notion du « corpus de référence ». Cette notion nous permet ainsi de décrire le contenu de notre corpus de référence qui sera en adéquation avec notre problématique de l'accès au droit.

En effet, un corpus peut être vu comme un ensemble de textes homogènes, c'est-à-dire que les textes partagent des caractéristiques communes, permettant l'application d'outils et de techniques d'extraction de connaissances dans le but d'acquérir des informations. Un corpus est également un support pour évaluer et comparer des méthodes ou des systèmes de recherche d'information. Des corpus composés de thèmes variés (il s'agit généralement de textes de sources journalistiques et donc traitant de sujets divers) sont disponibles pour évaluer des systèmes de RI. Notre approche d'aide à la recherche d'information sur le domaine juridique impose un choix de corpus adéquat dans ce domaine précis. Bien qu'aucun corpus ne soit *a priori* défini et établi dans le domaine du droit, à des fins d'évaluation, d'autres corpus existent qui conduisent à certaines considérations quant au choix de l'un d'eux.

Le futur corpus n'étant pas reconnu en tant que tel dans la communauté de la RI, il est alors utile de définir la notion de corpus de référence (définition des caractéristiques nécessaires) et de motiver notre choix à travers des considérations à la fois juridiques (nécessitant l'avis d'experts du domaine) et pratiques (le corpus doit permettre l'évaluation de la méthode, si possible sans l'avis d'experts). Dans ce chapitre, nous présentons également la méthodologie employée pour notre approche de l'aide à la navigation.

Avant de décrire notre algorithme de classification, nous nous intéressons dans le chapitre 5 à l'abstraction des documents composant le corpus, ce qui représente la première étape d'une méthode de classification. Dans ce chapitre, nous nous intéressons, dans un premier temps, aux différentes méthodes d'extraction de termes. Dans un second temps, nous essayons de mettre en place une méthode appropriée au domaine juridique dans le but d'abstraire les documents aux seuls termes juridiques. Ce chapitre permet de mettre en évidence que notre domaine d'application ne permet pas d'appliquer des méthodes classiques de réduction de termes, telles que la réduction par synonymie, sans perte d'informations utiles.

Dans le chapitre 6, nous proposons une nouvelle méthode de classification, appelée Ω-means. De type partitionnement, elle est inspirée, plus précisément, de l'algorithme k-means. Utilisant une matrice de similarité creuse, elle permet de classer un grand nombre de

documents sur un grand nombre de paramètres. La plupart des algorithmes de type partitionnement ont l'inconvénient majeur de ne pouvoir déterminer un nombre initial *K* de classes. Dans ce chapitre, nous proposons une méthode pour déterminer automatiquement la valeur de *K*.

Nous évaluons notre algorithme, dans le chapitre 7, sur notre corpus de référence (défini au chapitre 4) et sur un échantillon de celui-ci afin de la comparer avec d'autres méthodes connues. De plus, des évaluations sont effectuées avec une méthode de classification aléatoire. Nous constatons que la méthode donne des résultats suffisamment satisfaisants pour pouvoir appliquer l'algorithme récursivement.

Dans le chapitre 8, nous utilisons notre algorithme Ω-means afin de l'intégrer dans des modèles de navigation classiques tels que le plan de classement (approche statique) ou bien l'expansion de requêtes. Dans ce dernier modèle, nous proposons deux approches différentes : une approche dynamique et une nouvelle approche « semi-dynamique » appelée SearchXQ. Cette dernière approche est une combinaison de l'approche statique et dynamique et tend à pallier les inconvénients de celles-ci.

Chapitre 2

La recherche d'information

Résumé

Dans ce chapitre, on présente un système classique de recherche d'information où l'utilisateur reste passif entre la requête et la présentation des résultats. Ce type de système est le plus répandu sur Internet.

Un système de recherche d'information (RI) est fondé sur une succession d'étapes que nous rappelons à travers un ensemble de bases théoriques et de méthodes statistiques les plus couramment utilisées.

D'autres types de système existent qui demandent à l'utilisateur un effort supplémentaire (l'utilisateur est alors actif). Ces systèmes, présentés également dans ce chapitre, sont des aides à la recherche d'information. Ils diffèrent également dans la présentation des résultats (interfaces) et dans la navigation, et sont fondés sur le principe que l'utilisateur rencontre souvent des problèmes pour formuler son besoin.

2.1 Introduction

Le terme de « recherche d'information » regroupe plusieurs types d'approche. Cette diversité est relative à plusieurs critères. Le premier est la nature ou le format des documents : par exemple, certains documents sont structurés et d'autres pas ; le format est généralement textuel bien que d'autres formats existent, etc. Le second critère est lié à la méthode de recherche employée et enfin, le dernier critère est lié à l'usage de la méthode. Dans ce chapitre, nous nous intéressons exclusivement à la recherche documentaire.

Le processus de recherche documentaire est composé de trois étapes principales, que nous décomposons ici : indexation, recherche, présentation.

Un système de recherche documentaire doit, dans un premier temps, indexer une base documentaire (ou un corpus, une collection), c'est-à-dire que tous les documents de la base documentaire sont parcourus et, pour chacun d'entre eux, une partie des termes (mot ou groupe de mots) est extraite (les termes les plus « importants »). Dans un second temps, le système doit, pour une requête donnée, retrouver un ensemble de documents répondant aux termes de la requête. Pour retrouver les documents, on utilise une mesure de ressemblance entre les termes de la requête et ceux de chaque document, ces derniers étant stockés dans un fichier « inversé » (cette notion est détaillée dans le sous § 2.2.1.2). Enfin, on présente les documents répondant à la question dans un ordre qui dépend de la valeur de la mesure de ressemblance précédemment calculée (on parle également de mesure de pertinence). Cette approche de la recherche documentaire est à la fois classique et la plus répandue : elle ne nécessite pas l'intervention de l'utilisateur dans le processus de recherche.

D'autres approches de la recherche documentaire existent où l'utilisateur doit intervenir dans le but d'affiner sa requête. Ce sont des systèmes d'aide à la recherche d'information.

Dans ce chapitre, nous décrivons, dans un premier temps, un système de recherche d'information classique. Dans un second temps, nous présentons, pour chaque étape principale d'un système de recherche, les différentes techniques utilisées pour l'indexation, le filtrage et la pondération des termes et les mesures de ressemblance. Nous abordons ensuite les techniques d'évaluation des systèmes. Nous présentons ensuite les différentes approches de la navigation et les interfaces correspondantes. Dans un dernier temps, nous proposons d'étudier succinctement le comportement de l'utilisateur à travers les ressources de différents moteurs de recherche.

2.2 Recherche d'information (RI)

Nous avons vu que la recherche d'information (*Information Retrieval* en anglais) regroupe plusieurs approches, du fait, notamment de la diversité possible de l'information. Dans cette section, nous développons uniquement les aspects de la recherche. Quelle que soit

l'approche, le but principal est de retrouver l'information voulue par l'utilisateur dans une base documentaire (structurée ou pas).

Il existe plusieurs types d'information : textuelles, sonores et visuelles. Il existe aussi plusieurs types de documents : documents textuels, sons, images et vidéos. Nous nous intéressons uniquement, dans ce travail, aux documents textuels que nous désignons, dans toute la suite de ce mémoire, par documents ou textes.

La problématique principale de la recherche documentaire est de retrouver, parmi un ensemble de documents C, un sous-ensemble de documents $D \subset C$ dits *pertinents* en réponse à une requête donnée par l'utilisateur (*cf*. Figure 2.4 p. 33).

Un processus classique de recherche documentaire peut se décomposer en trois composants principaux exprimés ci-dessous :

1. indexation des documents du corpus ;
2. application d'un algorithme de recherche à partir de la requête de l'utilisateur ;
3. présentation des résultats à l'utilisateur.

Algorithme 2.1 – Les composants de la recherche de documents

La première étape de la RI est donc l'indexation de la base documentaire. Les documents sont parcourus un par un et, pour chacun, un ensemble d'informations est extrait. Cet extrait se caractérise, de façon classique, par une liste de termes : mots, groupes de mots, mots composés, etc., qui représentent le mieux le document ; on parle aussi de descripteurs de document. Bien que le but soit de représenter un document uniquement par ses descripteurs, il en est rarement ainsi. La seconde phase de l'indexation est de regrouper toutes les données dans un index dit inversé.

La seconde étape de ce processus est l'application d'un algorithme de recherche. L'algorithme prend en entrée la requête d'un utilisateur sur laquelle on applique une extraction de termes. A partir de ces termes, l'étape consiste à déterminer les documents qui répondent à la requête à l'aide d'une mesure de ressemblance entre les informations de la requête et celles de chaque document (stockées dans l'index inversé). La mesure de ressemblance est soit une mesure de similarité, soit une distance.

La dernière étape est la présentation de la liste des documents pertinents. Cette liste est ordonnée selon un *calcul de pertinence* effectué pour chaque document.

2.2.1 Techniques d'indexation d'une base documentaire

L'étape d'indexation (d'un système de recherche d'information classique) peut se résumer de la façon suivante :

1. extraction des termes ;
2. filtrage des termes : élimination des termes vides (suivant une liste prédéfinie et/ou la fréquence des termes et/ou leur catégorie syntaxique) ;
3. lemmatisation éventuelle des termes (ou *stemming*) ;
4. attribution d'un poids aux termes de chaque document (création d'un index inversé).

Algorithme 2.2 – Indexation d'un corpus

L'indexation est le processus qui permet d'analyser chaque document de la collection et d'extraire pour chacun d'entre eux un ensemble d'informations. Ces informations sont ensuite stockées dans un index pour faciliter la recherche. L'indexation est une étape majeure dans un processus de recherche d'information : c'est elle qui conditionne la valeur d'un système documentaire.

Par définition selon [Mercier, 1997] : « l'indexation est l'opération qui consiste à analyser et à caractériser un document à l'aide de représentations des concepts contenus dans ce document, c'est-à-dire à transcrire en langage documentaire les concepts après les avoir extraits du document par une analyse. La transcription documentaire se fait grâce à des outils d'indexation tels que les thésaurus, la classification, qui permettent le choix et l'attribution de descripteurs ou mots-clés décrivant de la façon la plus exhaustive possible le contenu conceptuel d'un document. »

Dans un système de recherche documentaire classique, l'indexation consiste à extraire pour chaque document de la collection une liste de termes ainsi qu'une liste de paramètres pour chaque terme : par exemple le nombre d'occurrences du terme. L'étape suivante a pour but d'attribuer à chaque terme d'un document un poids, à l'aide d'une fonction de pondération préalablement choisie. La dernière étape établit un lien entre un terme et les documents qui le contiennent.

Lors de l'étape d'indexation, les données récupérées sont stockées dans des fichiers dénommés fichiers d'index. Il existe différentes façons de stocker ces informations. Quelques-unes sont décrites ci-dessous.

2.2.1.1 Matrices de bits

Dans ce mode de stockage, la présence d'un terme dans un document est codée de façon binaire : le terme est présent ou pas dans le document. Ainsi, une matrice de bits B est une matrice document-terme définie comme suit :

Soit $D = \{D_1, ..., D_j, ..., D_N\}$ l'ensemble des N documents du corpus *C*,

et $T = \{T_1, ..., T_i, ..., T_m\}$ l'ensemble des termes contenus dans *C*. On peut écrire :

$$B_{D,T} = \begin{pmatrix} b_{11} & \dots & b_{1j} & \dots & b_{1m} \\ \vdots & \ddots & \vdots & & \vdots \\ b_{i1} & \dots & b_{ij} & & \vdots \\ \vdots & & & \ddots & \vdots \\ b_{N1} & \dots & \dots & \dots & b_{Nm} \end{pmatrix}$$

où $b_{ij} \in \{0,1\}, \forall i \in [1, N], \forall j \in [1, m]$ et $b_{ij} = 1$ ssi le terme T_j est présent dans le document D_i.

2.2.1.2 Tableau inversé

La création d'un tableau inversé peut se faire à l'aide d'un tableau dit direct. Un tableau direct stocke, pour chaque document de la base documentaire, l'ensemble des termes le définissant. Dans un modèle vectoriel où chaque document est représenté par un vecteur de termes, cela consiste à stocker une matrice document-terme. Les tableaux directs sont très peu utilisés car ils sont peu efficaces pour la recherche documentaire. Pour chaque requête donnée, cela revient à énumérer tous les documents de la collection pour déterminer les documents qui contiennent les termes de la requête. Par conséquent, le système est lent et inadapté pour de grandes collections. Pour des collections de grande taille, les besoins de mémoire de ce modèle augmentent rapidement.

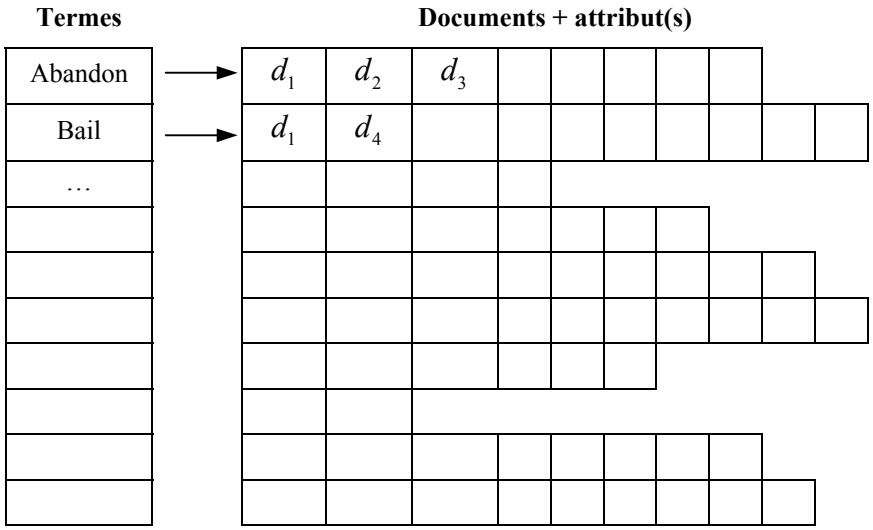

Figure 2.1 – Exemple de tableau inversé

Grâce aux tableaux inversés [Van Rijsbergen, 1979], on n'associe pas une liste de termes à chaque document mais à l'inverse, pour chaque terme du corpus, on associe la liste

des documents le contenant ainsi que d'autres informations comme, par exemple, le poids du terme dans un document le contenant.

L'utilisation d'un tableau inversé permet de diminuer considérablement les besoins en stockage. Un autre avantage est la diminution du temps nécessaire pour répondre à une requête portant sur un ensemble de termes car on accède directement et uniquement aux documents qui contiennent les termes de la requête.

Un exemple de tableau inversé est représenté Figure 2.1.

2.2.1.3 B-arbre

Dès que l'on aborde des volumes de grande taille, certaines opérations de l'indexation deviennent laborieuses, telles que la lecture et l'écriture de données et la lecture séquentielle de l'ensemble ordonné. L'objectif est alors de minimiser le nombre d'accès disque. Dans ce cas, l'indexation par les B-arbres est utilisée. En effet, l'accès aux données est rapide (quelques accès disque suffisent), de même que la suppression et l'ajout de données.

Un B-arbre généralise la notion d'arbre binaire équilibré et permet le stockage de plusieurs clefs dans chaque nœud de l'arbre (au lieu d'une clef unique dans un arbre binaire). La profondeur d'un B-arbre est donc inférieure à celle d'un arbre binaire.

On appelle ordre d'un B-arbre le nombre maximal de clés que peut contenir un nœud, augmenté d'une unité (l'ordre d'un B-arbre correspond donc au nombre maximal d'enfants de chaque nœud).

Un B-arbre ou arbre balancé, d'ordre *m*, possède les caractéristiques suivantes :

- la racine contient au minimum deux enfants (sauf si l'arbre est restreint à la racine) ;
- chaque nœud contient au plus $2m$ clefs ;
- chaque nœud, sauf la racine, contient au moins $m+1$ clefs ;
- un nœud est soit une feuille, soit il possède $2m+1$ enfants ;
- toutes les feuilles sont au même niveau : le B-arbre est ainsi équilibré ;
- toutes les clefs de arbres sont en ordre croissant et limitées par les clefs du nœud parent.

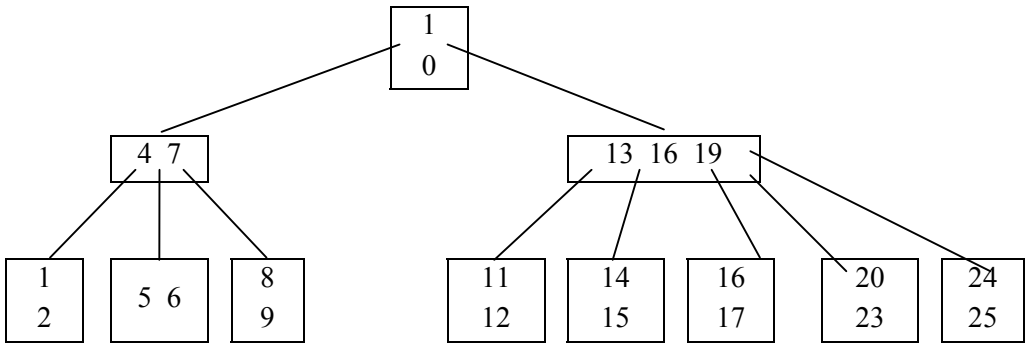

Figure 2.2 – Exemple de B-arbre d'ordre 2

2.3 Différentes relations entre les données et leur représentation

La représentation des données est l'une principales étapes de la conception d'un processus de RI qui influence de façon non négligeable les résultats du processus. Pour une méthode de classification, la représentation des données inclut souvent une matrice de relations entre éléments. Cette matrice M est dans la plupart des cas une matrice dite de similarité, c'est-à-dire que l'élément $M(i, j)$ est une mesure de similarité entre le document *i* et le document *j*. D'autres types de matrices existent car il est possible de classer différents types de données.

Un autre type est la matrice de cooccurrence de termes pour des données de type textuel où l'élément $M(i, j)$ représente le nombre de fois où l'élément i et l'élément j se retrouve dans le même document (ou dans le même paragraphe, segment, etc.).

Il existe différents modèles de représentation de données dont notamment les suivants :

- le modèle vectoriel ;
- le modèle des N-grammes.

2.3.1 Le modèle vectoriel

Dans un système de recherche documentaire, le modèle vectoriel introduit par [Salton, 1983] est généralement utilisé. Dans ce modèle, chaque document de la collection initiale est représenté par un ensemble de termes extraits lors d'une phase d'indexation. C'est le modèle que nous utiliserons dans cette étude. Un poids est attribué à tous les termes de chaque document. Un document est alors représenté par un vecteur dont la dimension est le nombre de termes différents de la collection et dont les composantes représentent les poids des termes présents dans le document. L'ensemble de ces vecteurs forme ainsi une matrice dite « documents/termes ». Dans un processus de recherche documentaire classique, la requête (composée de plusieurs termes) est également représentée par un vecteur du même espace. Ce vecteur est alors comparé à tous les vecteurs de la matrice à l'aide d'une fonction de similarité (ou d'une distance). Le calcul de toutes les mesures de similarité entre la requête et l'ensemble des vecteurs de document va permettre par la suite d'ordonner les documents en fonction de leur ressemblance avec la requête.

2.3.2 Le modèle des N-grammes

De façon générale, les N-grammes permettent de définir la probabilité d'apparition de suites de chaînes. L'une des applications du modèle des N-grammes concerne l'indexation de

corpus de grandes tailles [Lelu et Hallab, 2000]. Un N-gramme peut être aussi bien un N-uplet de caractères qu'un N-uplet de mots. Ce modèle ne représente pas les documents par un vecteur de termes mais par un vecteur de N-grammes dont les composantes sont les fréquences de ces N-grammes dans les documents correspondants. Généralement, dans les processus d'indexation, la valeur de N est soit égale à deux, soit égale à trois. On parle alors de bigramme ou de trigramme. Dans [Lelu et Hallab, 2000] une combinaison de bigrammes et de trigrammes est utilisée pour détecter les *termes composés* du corpus. L'avantage de ce modèle dans des processus d'indexation ou d'analyse textuelle est qu'il est indépendant de la langue.

D'autres modèles existent tels que le modèle de l'information syntaxique [Hindle, 1990], [Pereira et al., 1993] qui utilise des informations syntaxiques afin de détecter des relations de type nom-verbe. Chaque nom à classer sera représenté par un vecteur de verbes dont les composantes reflètent la distribution du lien entre le nom et le verbe. Afin d'exploiter objectivement la distribution des couples nom-verbe, ce type de modèle nécessite des corpus de grandes tailles pour obtenir des fréquences de couples élevées.

Dans la suite de cette thèse, seul le modèle vectoriel est utilisé.

2.4 Extraction de termes

On distingue différentes méthodes d'extraction de termes : les méthodes linguistiques fondées sur des règles syntaxiques, des méthodes statistiques qui reposent essentiellement sur la détection de segments répétitifs, les méthodes qui utilisent simultanément l'approche linguistique et l'approche statistique.

2.4.1 Statistique

La méthode statistique d'identification des termes consiste à déterminer les segments qui apparaissent suffisamment souvent dans les textes. Elle nécessite cependant de connaître certaines caractéristiques de la collection ou du corpus. Si ce dernier ne contient que peu de répétitions de segments ou si des termes n'apparaissent qu'une seule fois mais ont aussi leur importance, une méthode fondée sur des données statistiques risque de ne pas les considérer. Une méthode statistique a cependant l'avantage d'être multilingue car la langue dans laquelle les textes sont exprimés n'a finalement que peu d'importance.

L'outil Xtract [Smadja et McKeown, 1990] permet la création d'un lexique pour un corpus de grande taille par répétition d'un ensemble de mots. Cet outil extrait des termes grâce à la répétition de bigrammes.

Dans un premier temps, les bigrammes sont extraits et correspondent aux couples de mots. Une distance est attribuée aux différents tuples, ainsi, une distance égale à 1 pour le

tuple (w_1, w_2) indique que l'on retrouve dans les textes le mot w_1 suivi directement de w_2. Une distance égale à -1 équivaut à retrouver le mot w_1 précédé de w_2. Les tuples (w_1, w_2) et (w_2, w_1) sont considérés comme différents et sont tous les deux présents dans le lexique. A partir des bigrammes, on essaie de retrouver, sur une base fréquentielle, les n-grammes.

Une méthode statistique peut également être utilisée en complément d'une méthode syntaxique.

La méthode des segments répétés [Oueslati, 1999], [Rousselot et al., 1996] repose sur un calcul fréquentiel des ensembles de mots dont le premier, appelé *tête de syntagme*, possède certaines caractéristiques lexicales.

2.4.2 Syntaxique

Un deuxième type d'outils procède à l'identification de termes dans des textes grâce à des analyses syntaxiques menées sur ces textes. Ces outils fonctionnent avec des documents "étiquetés" où chaque mot du document est assigné à une catégorie syntaxique sur lesquels ils appliquent des règles de grammaire pour identifier les syntagmes.

Ces outils comportent donc un module d'étiquetage des textes qui, en utilisant éventuellement des dictionnaires, détermine la nature syntaxique de chaque mot des documents. Chaque mot est ainsi affecté à une étiquette. Dans le cas où plusieurs étiquettes sont possibles, l'outil déterminera l'étiquette la plus adéquate suivant le contexte. Dès lors que le texte est étiqueté, les règles de grammaires sont appliquées sur les phrases.

Ils intègrent ainsi des modules contenant, de façon plus ou moins simplifiée, les règles grammaticales d'une ou plusieurs langues ; ils dépendent alors de la langue dont ils connaissent la grammaire. La détermination des termes syntaxiquement pertinents se fera, entre autres, par l'utilisation de *mots frontière*. Une liste incluse dans l'outil, parfois paramétrable par l'utilisateur, comporte les mots qui déterminent la fin d'un terme et le commencement d'un nouveau. Ces mots frontière sont le plus souvent des verbes.

En sortie de ces outils, on obtient une liste des syntagmes pour les textes qui ont été soumis. Des indications lexicales ou grammaticales sur ces termes peuvent être données, par exemple la catégorie grammaticale des termes.

2.4.2.1 Sylex

Le logiciel Sylex [Constant, 1991], [Constant, 1995] est un analyseur linguistique cherchant à "comprendre et analyser un texte quelconque". L'outil présente une large part d'analyse syntaxique des textes, sur un "axe syntagmatique". L'analyse du texte sur un "axe paradigmatique" s'attache à la sémantique, s'enrichissant de chaque étape de l'analyse syntaxique. Cette dernière portion de l'analyse des textes n'apparaît cependant que peu dans

les sorties de l'outil. L'outil Sylex incorpore certaines règles de grammaire de la langue française (et d'autres telles que l'anglais, l'espagnol, etc.). Il se fonde sur la théorie de Tesnière [Tesniere, 1959], induisant que celle-ci permet de minimiser la création de concepts linguistiques, ce qui réduit d'autant les arbitraires potentiels. Ainsi, toute ambiguïté lexicale comme c'est, par exemple, le cas des mots qui sont à la fois des noms et des verbes, est partiellement résolue par le choix arbitraire d'une interprétation. L'avancement de l'analyse permet de revenir sur ces choix arbitraires. L'analyse syntaxique du texte se fait donc par "couches". La structuration syntaxique du texte évolue au fur et à mesure du passage dans ces différentes couches.

2.4.2.2 Syntex

Syntex [Bourigault, 1993] est un analyseur syntaxique. Il fonctionne sur des textes préalablement étiquetés et lemmatisés (voir § 2.5.3) dans lesquels sont indiquées, pour chaque mot, des informations telles que la catégorie grammaticale, les traits morphologiques (comme le genre et le nombre) et la forme lemmatisée. L'outil repère dans les textes les relations de dépendance syntaxiques suivantes : sujet de verbe, objet de verbe, complément prépositionnel de verbe, nom ou adjectif, épithète et attribut de nom. Grâce à des procédures d'apprentissage endogènes, le système acquiert lui-même les informations syntaxiques de sous-catégorisation qui lui sont nécessaires pour effectuer un repérage précis des termes complexes [Bourigault, 1993]. Sur la base des relations de dépendance établies, le module d'extraction de syntagmes construit un réseau de mots et de syntagmes. De ce réseau, on obtient l'ensemble des *candidats termes*, quelles que soient leurs catégories grammaticales et les contextes syntaxiques de chacun de ces termes.

2.5 Filtrage des termes

Le but du filtrage des termes est de réduire l'ensemble initial des termes (i.e. après extraction) afin d'améliorer les performances du processus de recherche d'information. Nous abordons, dans cette section, un ensemble de pré-traitements de nature statistique et morpho-syntaxique des termes. Ces pré-traitements peuvent se faire pendant la phase d'indexation.

2.5.1 Mots vides

Un mot vide (*stopword* en anglais) est par définition un mot non significatif dans un processus de recherche documentaire.

Pendant la phase d'indexation, on ne peut déterminer, *a priori*, les mots significatifs (ou mots-clef) de chaque document car chaque mot composant le document est un mot-clef potentiel.

Cependant, on peut éliminer les mots ayant certaines caractéristiques tels que par exemple les mots trop fréquents [Luhn, 1957]. Les articles et les déterminants en sont un exemple. Cette élimination, qui permet de réduire la taille de l'index, se justifie par le fait

qu'ils sont présents dans la quasi-totalité des documents du corpus et ne peuvent ainsi être discriminant dans une requête. La prise en compte de ces mots dans une requête risque de retourner le corpus en quasi-totalité.

Les mots vides peuvent être également éliminés à l'aide d'une liste préalablement définie de mots (*stop list*). En considérant comme corpus les codes du droit français, la liste des mots vides contient par exemple : code, livre, article, section, paragraphe, etc. Une autre approche est de considérer la catégorie syntaxique du mot et de ne garder que ceux qui entrent dans la catégorie voulue. Ainsi, dans certains systèmes, les adverbes, les adjectifs indéfinis, les pronoms, les verbes sont éliminés. Cette approche peut éventuellement se substituer à une liste prédéfinie.

2.5.2 Filtrage par pondération

Dans la section précédente, nous avons évoqué un filtrage des termes les plus fréquents dans le corpus. Dans cette section, nous généralisons cette approche selon une fonction de pondération. (*cf.* section 2.6). En effet, le filtrage peut s'étendre également aux termes ayant certaines caractéristiques statistiques, autres que la fréquence des termes dans le corpus. Dans la littérature, une approche est d'éliminer les mots fréquents dont la valeur de l'`Idf` (*cf.* section 2.6) est supérieure à 0.1. Cela permet de réduire le nombre de mots sans détérioration des performances de la recherche d'information.

Les travaux de Lame [2002] sur le filtrage des termes juridiques ont montré que ces derniers n'avaient pas, *a priori*, de caractéristiques spécifiques. Son objectif était de détecter un éventuel seuil, suivant une fonction de pondération, qui permettrait de distinguer un terme juridique d'un terme non juridique, suivant que la valeur pondérale du terme soit au-dessus ou au-dessous du seuil.

Dans le Chapitre 5, nous étendrons ces travaux en ajoutant une composante syntaxique à celle statistique.

2.5.3 Lemmatisation

La lemmatisation est l'opération qui consiste à réduire les formes fléchies des mots à leur racine grammaticale. Par exemple, les mots « voiture », « voitures » et « voituriers » auront pour *lemme* « voiture ». La lemmatisation est fondée, par exemple, sur une analyse morpho-syntaxique ou sur des dictionnaires.

Un mot peut être modélisé comme suit : préfixe + racine + suffixe [Moens, 2000]. L'objectif de la lemmatisation est d'améliorer les performances d'un système de recherche d'information grâce à un appariement entre le vocabulaire de la requête et celui des documents et/ou à un appariement entre le vocabulaire des documents du corpus. Cette opération permet de réduire le nombre de termes dans un index, ce qui est intéressant du point de vue du stockage des données. La lemmatisation peut être vue comme un mécanisme d'expansion de

requête puisqu'on modifie l'ensemble des mots de la requête initiale donnée par un utilisateur. Enfin, elle permet également de lever l'ambiguïté relative au sens des mots dans un document.

La morphologie des mots porte sur la structure interne des mots. Il existe deux types de morphologie : flexionnelle et dérivationnelle. La morphologie flexionnelle consiste, pour un mot, à lui ajouter (ou retirer) les marques du pluriel ou les terminaisons de conjugaison par exemple. On n'obtient pas de nouveaux mots après avoir appliqué cette opération car le sens du mot que l'on obtient est généralement identique.

La morphologie dérivationnelle consiste, pour un mot, à lui ajouter (ou retirer) des préfixes, des suffixes. On obtient ainsi des nouveaux mots avec un sens différent du sens du mot initial.

La suffixation permet généralement de passer d'une classe grammaticale à une autre tandis que la préfixation joue un rôle important au niveau sémantique.

Les suffixes et les préfixes possèdent également un sens que l'on retrouve dans la définition des mots construits à partir d'un mot et d'un ensemble de préfixes et (ou) de suffixes.

Les principales approches de la lemmatisation sont les suivantes :

1. La méthode des dictionnaires : les mots et leurs bases lexicales sont stockés. On stocke des couples de la forme : (terme, base lexicale). L'avantage de cette approche est la qualité des résultats. Les inconvénients sont la taille des dictionnaires et éventuellement le nombre d'entrées dans le dictionnaire, c'est à dire le taux de couverture des mots du corpus par le dictionnaire.
2. La méthode de réduction par affixes : on réduit, successivement, les mots en base lexicale en enlevant les suffixes et les préfixes tant que cela est possible. La méthode repose également sur un ensemble d'heuristiques propres au langage des documents du corpus pour remplacer certaines lettres. Par exemple sur un corpus de documents en anglais, on peut remplacer le « y » par un « i ». Ceci permet de pallier certaines caractéristiques du langage et améliorer ainsi la qualité des bases lexicales.
3. La méthode par variation de lettres successives : cette méthode repose sur la fréquence des séquences de lettres successives. On peut ainsi détecter les affixes et réduire le mot. Pour chaque début de séquence de lettres, on calcule le nombre de lettres successives possibles dans le corpus pour cette séquence. Le nombre de lettres successives a tendance à décroître avec la longueur de la séquence sauf pour un affixe. On détecte les affixes grâce aux pics qu'ils génèrent au niveau du nombre de lettres successives. L'avantage de cette méthode est qu'elle est indépendante du langage du corpus. Cette méthode ne permet pas de distinguer un morphème grammatical d'un morphème lexical (ou lexème). Il existe une méthode similaire fondée sur l'entropie ; on calcule la probabilité qu'une lettre *i* soit suivie d'une lettre *j*. On détermine par défaut un seuil qui permet de détecter les affixes si celui-ci est atteint.

2.6 Pondération des termes

Il existe dans la littérature, un nombre relativement important de fonctions de pondération de termes [Salton & Buckley, 1988], [Harman, 1992]. Nous ne citerons ici que quelques fonctions parmi les plus couramment connues et/ou utilisées.

***Tf*(*w*, d_i)** (« Term frequency ») qui tient compte uniquement du nombre $occ(w, d_i)$ d'apparitions du terme *w* dans le document d_i.

***Idf*(*w*, d_i)** (« Inverse document frequency ») qui ne tient compte que du nombre de documents qui contiennent le terme *w*.

***Tf.Idf*(*w*, d_i)** (« Term frequency - Inverse document frequency ») est la combinaison des deux fonctions de pondération citées ci-dessus ($Tf.Idf(w, d_i) = Tf(w, d_i).Idf(w, d_i)$). Elle donne de l'importance aux termes qui apparaissent dans peu de documents mais dont la fréquence dans ces documents est élevée. Cette fonction est définie de la façon suivante :

$$Tf.Idf(w, d_i) = \frac{occ(w, d_i)}{|d_i|} \cdot \log\left(\frac{|C|}{\sum_{i=1}^{|C|} \delta(w, d_i)}\right) \qquad (2.1)$$

avec $|d_i|$ le nombre d'occurrences total des mots présents dans d_i, $C = \{d_i\}$ le corpus et $\delta(w, d_i)$ est égal à 1 si d_i contient w sinon 0.

Cette formule a été déclinée sous différentes formes, c'est-à-dire avec des normalisations variées.

2.7 Mesures de ressemblance

Les mesures de ressemblance regroupent différentes catégories : mesures de similarité, distances, etc. Ces mesures sont utilisées, comme précédemment évoqué, dans le calcul de la similarité entre la requête et chaque document du corpus. Cette mesure de similarité permet de prendre en compte le poids des termes de la requête et ceux des documents. Dans cette section, nous présentons quelques mesures, parmi les plus couramment utilisées.

L'une des mesures de similarité la plus fréquemment utilisée est cosine [Salton, 1983]. Elle consiste à calculer le cosinus de l'angle formé par le vecteur de la requête et le vecteur d'un document du corpus répondant à la requête (*cf.* Figure 2.3). C'est un produit scalaire qui prend en compte le poids des termes (poids des composantes du vecteur) et qui normalise le résultat suivant la taille des documents.

La mesure `cosine` est définie de la façon suivante :

$$\text{Cosine}(r,d) = \frac{\sum_{T_i \in d \cap r} w(T_i,r) \cdot w(T_i,d)}{\sqrt{\sum_{T_i \in r} w(T_i,r)^2 \cdot \sum_{T_i \in d} w(T_i,d)^2}} \quad (2.2)$$

avec T_i un terme, d un document, r la requête et $w(T_i,x)$ le poids de T_i dans x. Ce poids est généralement calculé avec `Tf.Idf` (*cf.* section 2.6).

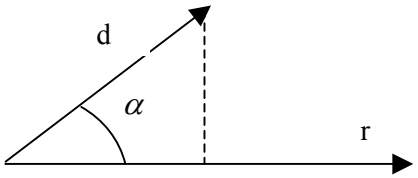

Figure 2.3 – Cosinus entre *d* et *r*, représentant respectivement un document du corpus et une requête.

D'autres mesures de similarité existent telles que les mesures de Dice ou de Jaccard par exemple.

2.8 Évaluation des résultats d'une requête

Un système de recherche fournit donc, pour une requête donnée par l'utilisateur, un ensemble de documents dont il est nécessaire d'évaluer la *pertinence* (*relevance* en anglais). Cette évaluation se fait par comparaison entre les réponses trouvées et celles considérées comme idéales.

Il existe dans la littérature différents critères d'évaluation très largement utilisés pour mesurer la qualité (ou pertinence) d'une recherche. Si ces critères peuvent avoir des limites pour l'évaluation des performances d'un système de RI (voir section 2.8.6), ils sont en revanche intéressants pour mesurer la qualité d'une classification, par exemple, la précision qui mesure la quantité de documents pertinents retrouvés.

Dans le cas d'un système de recherche, une liste triée par ordre de pertinence de documents est proposée à l'utilisateur à partir du corpus de travail. Cette liste contient généralement des documents pertinents et non pertinents pour une requête donnée. Dans cette liste, certains documents pertinents peuvent être oubliés : le système n'a pas pu les retrouver.

Ainsi, pour une requête donnée, le corpus se divise en quatre catégories de documents (voir Figure 2.4), l'objectif principal étant de retrouver le maximum de documents pertinents et un minimum de non pertinents. Un autre objectif est de retrouver tous les documents pertinents dans les premiers résultats. D'un autre point de vue (celui des critères), il s'agit de

diminuer le nombre de documents non pertinents trouvés ainsi que les documents pertinents non trouvés.

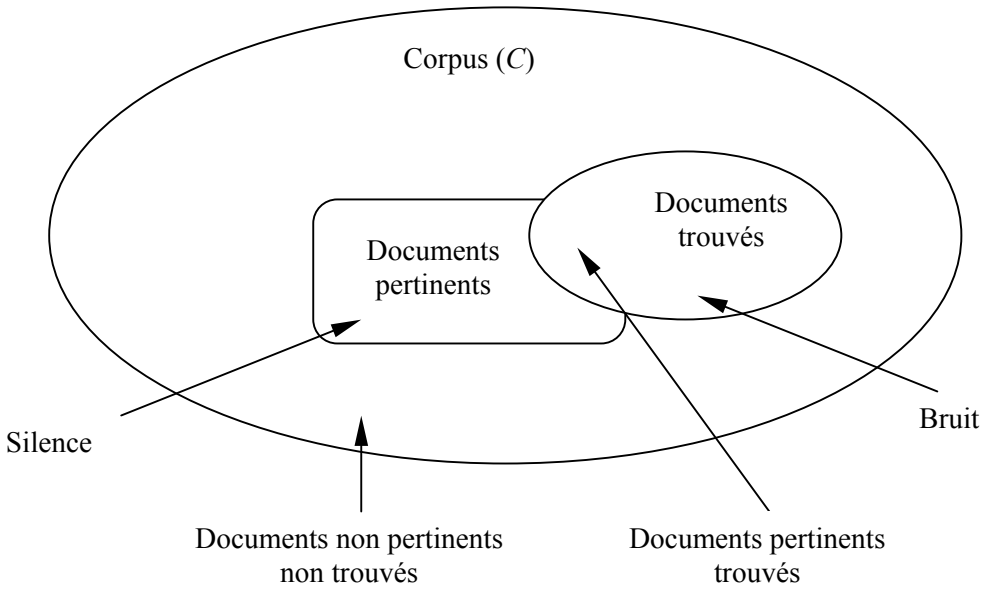

Figure 2.4 – Pertinence : découpage du corpus pour une requête donnée.

2.8.1 Précision

La précision est le rapport entre le nombre de documents pertinents trouvés et le nombre de documents trouvés.

$$précision = \frac{\#documents\_pertinents\_trouvés}{\#documents\_trouvés} \tag{2.3}$$

Il est possible de déterminer la précision à différents niveaux : la précision sur les dix premiers documents de la liste par exemple. En effet, il est intéressant de déterminer la précision sur les *n* premiers documents présentés à l'utilisateur. Ce dernier ne regarde généralement que la première page de résultats d'une requête [Silverstein et al., 1998].

La Figure 2.5 montre un exemple de courbe de précision et la courbe optimale. La courbe optimale est horizontale sur l'intervalle [0, *P*] où $P = \#documents\_pertinents$. Au-delà de *P*, la courbe décroît : il ne reste que des documents non pertinents.

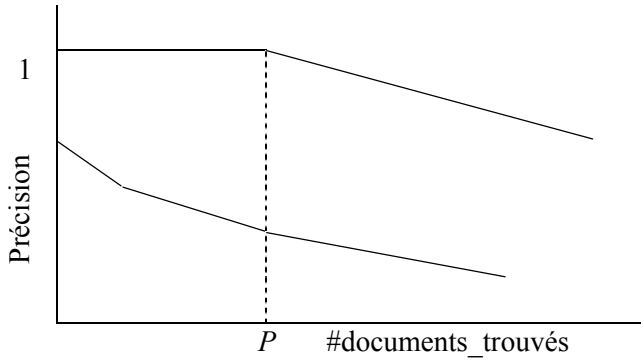

Figure 2.5 – Exemple d'une courbe de la précision et de la courbe optimale

Le bruit est le rapport entre le nombre de documents pertinents non retrouvés et le nombre total de documents trouvés. Par conséquent, le bruit est défini comme suit :

$$bruit = 1 - précision$$

2.8.2 Rappel

Le rappel ne tient compte que du nombre de documents pertinents trouvés par rapport au nombre de documents pertinents pour une requête donnée.

$$rappel = \frac{\#documents\_pertinents\_trouvés}{\#documents\_pertinents} \tag{2.4}$$

Le silence, qui correspond aux documents retrouvés non pertinents, est défini comme suit :

$$silence = 1 - rappel$$

2.8.3 Rappel/Précision

Il est possible de tenir compte à la fois de la précision et du rappel. Pour construire la courbe du rappel en fonction de la précision, il suffit de prendre la précision pour un ensemble fini de valeurs de rappel. Cette courbe est cependant peu significative pour comparer deux systèmes : sur la Figure 2.6, il est difficile de déterminer « la meilleure courbe » car cette notion dépend des attentes voulues pour un système de RI.

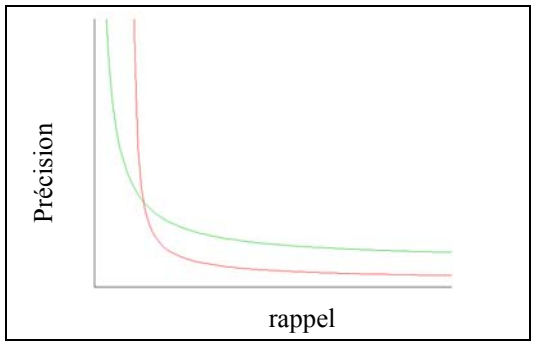

Figure 2.6 – Exemples de courbes de rappel et de précision obtenues sur un même corpus.

Il existe d'autres critères dans la littérature qui combinent le rappel et la précision tels que le critère *E-measure* ou le critère *F-measure* que nous décrivons succinctement ci-dessous.

2.8.4 E-measure

Ce critère est défini de la façon suivante [Van Rijsbergen, 1979] :

$$E = \frac{1}{\alpha\left(\frac{1}{P}\right) + (1-\alpha)\frac{1}{R}} \tag{2.5}$$

où P = précision et R = rappel.

Suivant la valeur de α qui mesure l'importance relative de la précision ou du rappel (α : compris entre 0 et 1), on va pouvoir donner plus à l'un ou à l'autre.

2.8.5 F-measure

Ce critère, cas particulier de *E-measure*, est défini comme suit :

$$F = \frac{2 \cdot R \cdot P}{R + P} \tag{2.6}$$

où R = rappel et P = précision.

Pour ce critère, $\alpha = 0.5$, c'est-à-dire que l'on donne autant d'importance à la précision qu'au rappel.

2.8.6 Limites

Les critères cités ci-dessus définissent la pertinence d'un document (et inversement la non pertinence) comme une notion booléenne : le document est pertinent ou pas. Une vue

booléenne de la pertinence est en partie plausible : un document peut être complètement hors sujet pour une requête donnée comme un document peut répondre parfaitement.

Cependant, il existe toute une gamme de documents pour lesquels la pertinence par rapport à la requête est fonction des demandes et des connaissances implicites de l'utilisateur : les documents sont donc plus ou moins pertinents suivant le point de vue de l'utilisateur. Selon [Lefèvre, 2000], la notion de pertinence doit être vue comme une fonction continue définissant un ordre sur l'ensemble des documents trouvés et permettant de les trier par pertinence décroissante. Notons que la notion de pertinence peut être vue également comme une fonction discrète.

L'ensemble des critères cités (précision, rappel, bruit, silence) quantifient les défauts et les qualités des systèmes de RI. Mais ces critères peuvent avoir une signification différente en fonction du contexte de la recherche [Lefèvre, 2000] :

- le bruit a peu d'importance pour un corpus de petite taille ;
- si le document recherché existe, le critère de performance est alors lié au temps passé à le retrouver et non au taux de silence ;
- pour des corpus de grande taille, le besoin est concentré sur la pertinence des dix ou vingt premiers résultats.

2.9 Interfaces et visualisation

La dernière étape d'un processus de recherche documentaire (voir Algorithme 2.1) est la présentation des données. Outre le fait qu'il faille présenter les résultats suivant une fonction de pertinence, la visualisation des résultats, la navigabilité et les moyens d'aide à la recherche, éventuels, mis à disposition de l'utilisateur sont également importants dans ce processus. Dans cette section, on s'intéresse aux différentes approches de navigabilité et aux interfaces correspondantes. Ces interfaces proviennent à la fois d'outils de recherche (Scatter/Gather par exemple) mais également d'outils grand public (Yahoo[1], Exalead[2] par exemple). Cette section résume les différentes approches des moteurs de recherche sur le Web.

2.9.1 Généralités sur les interfaces

Une interface utilisateur est l'outil de communication intermédiaire entre l'utilisateur et la machine (c'est-à-dire le moteur de recherche). La recherche d'information est un processus imprécis dans le sens où tout utilisateur n'est pas nécessairement prédisposé à formuler correctement une requête correspondant à ses besoins. On peut également imaginer qu'un utilisateur n'est pas forcément capable de dire si l'information qu'il cherche se trouve dans la liste de documents que le moteur lui retourne. La liste de documents est la page de résultats que retournent la plupart des moteurs de recherche sur le Web. Elle présente les

[1] http://www.yahoo.com
[2] http://www.exalead.com

caractéristiques de chaque document. Ces caractéristiques comprennent généralement le titre, un résumé, la taille du document, et d'autres informations.

Puisqu'une interface utilisateur doit être capable d'apporter une aide à l'utilisateur durant toute sa démarche de recherche d'information et si possible de façon intuitive, plusieurs aspects interviennent.

Voici différentes aides qu'il est possible de proposer à l'utilisateur :

1. aide à l'interprétation du besoin ;
2. aide à la formulation de la requête ;
3. aide à la sélection des ressources disponibles ;
4. aide à la compréhension des résultats.

Une interface utilisateur est un outil interactif entre l'utilisateur et la machine. Ben Schneiderman ([Schneiderman, 1997] dans [Hearst, 1999]) a répertorié, dans une optique d'efficacité, un ensemble de principes pour la conception d'une interface utilisateur dont quelques-uns sont cités ici :

1. Permettre un retour d'information

Cette notion recouvre les différentes méthodes que l'on peut mettre en œuvre pour lier les spécificités de la requête et la liste des documents retournés en réponse. Elle regroupe également les méthodes qui permettent de retourner les informations concernant les liens entre les différents documents retournés mais aussi les informations concernant les liens entre les documents retournés et l'ensemble des méta-données définissant le corpus (par exemple dans un ensemble de catégories prédéfinies).

2. Permettre un retour en arrière des actions

3. Garder en mémoire l'historique de la recherche

Lors d'une recherche d'information, un utilisateur peut définir ses paramètres de recherche et au fur et à mesure de sa recherche les modifier. Ce principe de mise en mémoire permet d'avancer dans sa recherche d'information, de prendre différents chemins et de pouvoir à tout moment revenir sur n'importe quelle session de recherche, c'est-à-dire de reprendre d'anciens paramètres de recherche.

4. Mettre en place plusieurs interfaces correspondant au profil utilisateur : débutant ou expert en RI.

Le principal compromis dans la conception d'une interface utilisateur est la simplicité par opposition à la puissance. Les interfaces qui sont simples d'utilisation le sont au détriment de la flexibilité, voire de l'efficacité de l'utilisation. Une interface riche en options va permettre à un utilisateur averti ou expert en RI de paramétrer l'outil à sa convenance, d'utiliser les options avancées et ainsi d'améliorer l'efficacité de la recherche. Par conséquent, pour un novice en RI, une interface simple sera préférable car la prise en main est rapide et seules les fonctionnalités de base seront disponibles. Ainsi, l'utilisateur n'a pas à apprendre le fonctionnement des modes de recherche de l'outil. Ce type d'interface est généralement intuitive. Cette interface sera donc peu flexible et moins efficace. Alternativement, une interface pour des experts ou des utilisateurs expérimentés proposera beaucoup plus d'options et/ou différents modes d'interaction et permettra d'utiliser toute la quintessence de l'outil. Une bonne conception d'interfaces permettra de passer d'une interface à l'autre facilement (ce que l'on retrouve généralement pour les moteurs de recherche sur le Web, par exemple, qui possèdent ces deux types d'interface).

Nous venons de voir que la principale différence entre ces deux types d'interface était la simplicité par opposition à la puissance de recherche. En regardant plus en détail, cette différence se traduit en partie par la quantité d'informations qui est fournie à l'utilisateur. Un utilisateur débutant dans la RI (ou avec un outil de recherche) et/ou pour une collection spécifique de documents (par exemple le droit français) ne pourra faire les bons choix tout de suite car il aura peu de connaissance à la fois sur l'outil et la collection de documents. Par contre, un expert habitué de l'outil et familiarisé avec la collection de documents pourra donner/choisir les bons termes avec éventuellement un poids spécifique à chacun d'eux pour préciser sa requête.

2.9.2 Modèles d'interaction avec les interfaces

Le processus d'accès à l'information peut être vu comme un cycle intégrant les phases suivantes [Salton, 1989] :

1. définition de la requête ;
2. soumission de la requête ;
3. récupération et examen des résultats ;
4. arrêt ou reformulation de sa requête.

La Figure 2.7 montre plus en détail le processus standard de l'accès à l'information.

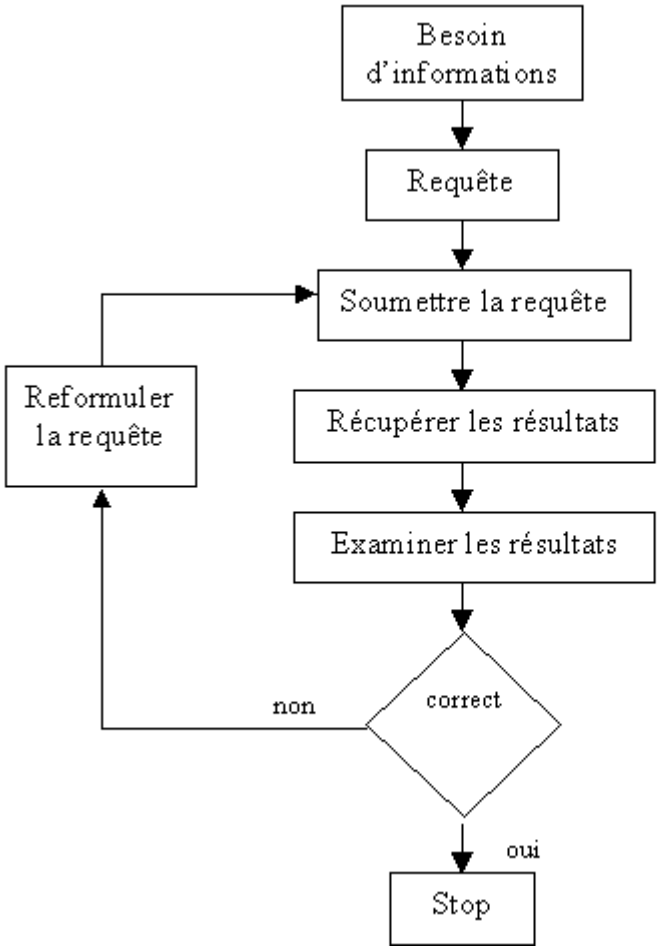

Figure 2.7 – Représentation schématique d'un processus de recherche d'information

Ce modèle d'accès à l'information est le plus répandu en ce qui concerne les moteurs de recherche sur Internet : en réponse à une requête, une liste de documents est proposée à l'utilisateur. Cette liste de documents peut être parfois très longue, c'est-à-dire pouvant aller jusqu'à quelques centaines de milliers de documents. Dans ce cas, l'inconvénient majeur de ce modèle se dévoile si les documents dit pertinents ne se trouvent pas parmi les premiers documents retournés. L'utilisateur est alors contraint de reformuler sa requête à travers un apprentissage personnel qui consiste à déterminer le mot ou l'ensemble de mots à ajouter ou à enlever de la requête initiale pour atteindre les documents voulus. Cet apprentissage se traduit par une étape de lecture : par exemple la lecture du titre des documents, et la lecture des résumés ou la lecture des documents dans leur totalité. Cet apprentissage est un processus itératif qui s'arrête lorsque l'utilisateur atteint les documents pertinents.

Le second modèle, le plus utilisé par les moteurs de recherche, est la catégorisation des documents.

2.9.3 Catégorisation d'une collection

Il existe aujourd'hui plusieurs collections ou bases de données interrogeables en ligne. Par exemple, la base de données biomédicale MEDLINE[1] qui contient environ 11 millions de publications, dont les plus anciennes datent de 1965. Cette base de données propose plusieurs interfaces d'interrogation : interface simple, interface avancée et une interface « de recherche et de navigation » utilisant les termes MeSH (Medical Subject Headings). Les termes MesH sont des termes biomédicaux définis par la National Library of Medecine et leur nombre est supérieur à 19 000 termes. Ils sont classés dans une hiérarchie possédant jusqu'à neuf niveaux. L'interface de recherche et d'informations permet à l'utilisateur d'ajouter facilement un terme biomédical à une requête en le choisissant parmi une hiérarchie de termes MeSH proposée, puis d'étendre sa requête avec le vocabulaire adéquat. La navigabilité de cette interface se fait au niveau de la hiérarchie de termes. Cet ensemble de termes qui représente généralement les différentes thématiques du domaine a deux principaux objectifs :

- Le premier est d'organiser l'ensemble des documents de la collection.
- Le second est d'apporter une aide à la recherche d'information à l'aide d'un processus de spécification de requête.

Cette spécification permet de créer des requêtes avec un vocabulaire adéquat, celui du domaine, et permet par conséquent d'améliorer la précision des réponses. Par exemple, il est facile d'imaginer qu'un utilisateur ne possédant pas le vocabulaire adéquat pose une requête telle que « maladie du cœur » au lieu du terme spécifique « maladie cardiaque » qu'il trouvera dans la hiérarchie. Ainsi ce type de structuration d'informations est le premier pas vers la création d'une passerelle entre le vocabulaire de la collection et le vocabulaire de l'utilisateur.

Cependant, ce type d'interface proposant une aide à la recherche par une sélection d'un terme dans une hiérarchie (ou simplement une liste) ne présente pas que des avantages. Ces interfaces peuvent s'avérer inefficaces si la liste de termes proposés est trop longue et demandent ainsi un temps excessif à l'utilisateur pour choisir un terme. Un second inconvénient est lié à la spécificité même des termes pour certaines collections. Des termes trop spécifiques ne permettront pas à certains utilisateurs de choisir le terme adéquat ou de naviguer à travers une hiérarchie. Une solution est alors d'apporter une aide à la sélection des termes.

2.9.4 Catégorisation de plusieurs collections

La catégorisation de plusieurs collections la plus connue sur le Web est celle de l'annuaire Yahoo qui regroupe les pages Web en index thématique composé de milliers de

[1] http://medline.cos.com
Cette base de donnée couvre plusieurs domaines comme la médecine dentaire, la médecine vétérinaire, le système de santé ou la science préclinique.

catégories. Aujourd'hui, la plupart des moteurs de recherche sur le Web proposent également une recherche thématique. L'avantage de ce type d'interface est que l'utilisateur qui ne sait initialement comment exprimer sa requête peut naviguer à travers la hiérarchie des catégories. A chaque étape, l'utilisateur choisit la catégorie qui correspond le mieux à ses besoins d'information. Cependant, cette approche peut s'avérer inefficace pour des besoins précis pour lesquels on ne peut retrouver les catégories correspondantes. Dans le cas où l'utilisateur trouverait le chemin des catégories correspondant à ses besoins, il doit encore parcourir une liste de sites Web qui lui est retournée et parcourir les sites qui lui semblent pertinents pour retrouver l'information qu'il désire. Un autre inconvénient concerne l'étiquetage des catégories ; leurs noms ne peuvent pas être utilisés dans une recherche classique car ce sont des étiquettes et non forcément des termes issus des documents.

Nous venons de voir que les interfaces utilisant des catégories statiques, c'est-à-dire découplées de toute requête, peuvent s'avérer peu efficaces à la fois dans la navigation et l'aide apportée. Une solution est alors d'aider l'utilisateur par rapport à sa requête initiale en lui proposant une liste de thèmes liés à sa requête.

2.9.5 Aspect automatique

L'extraction de thèmes pour une collection donnée peut être effectuée de façon automatique. On peut utiliser la classification non-supervisée et les co-occurrences. Cet aspect automatique de la classification peut être vu de façon globale ou de façon locale : la classification locale prend en compte la requête contrairement à la classification globale.

La méthode Scatter/Gather [Cutting et al., 1992], [Cutting et al., 1993] consiste en une classification non-supervisée locale qui intègre une interface pour l'aide à la recherche d'information. L'algorithme de cette méthode est détaillée dans le chapitre suivant (*cf*. section 3.5.11). L'aide se présente sous forme de liste de thèmes où chaque thème est défini par un résumé composé d'une liste de mots[1] (voir Figure 2.8). Le nombre de thèmes correspond au nombre de classes de la partition retrouvée. L'utilisateur choisit parmi cette liste de thèmes (en utilisant les résumés pour le contexte) celui qui correspond le mieux à ses besoins et relance l'algorithme avec le nouveau thème. Une nouvelle classification est effectuée sur le sous-ensemble de documents correspondant au thème sélectionné puis une nouvelle liste de thèmes est ainsi proposée. A noter que cet algorithme peut être intégré dans un système de recherche classique puisqu'il intervient en sortie, c'est-à-dire qu'il récupère la liste des documents réponses. L'expérimentation dans [Pirolli et al., 1996a] montre que la méthode Scatter/Gather appliquée sur une collection de grande taille retourne correctement certains thèmes de la collection.

[1] Les mots définissant les thèmes et donc les classes sont les mots qui représentent le mieux chaque classe.

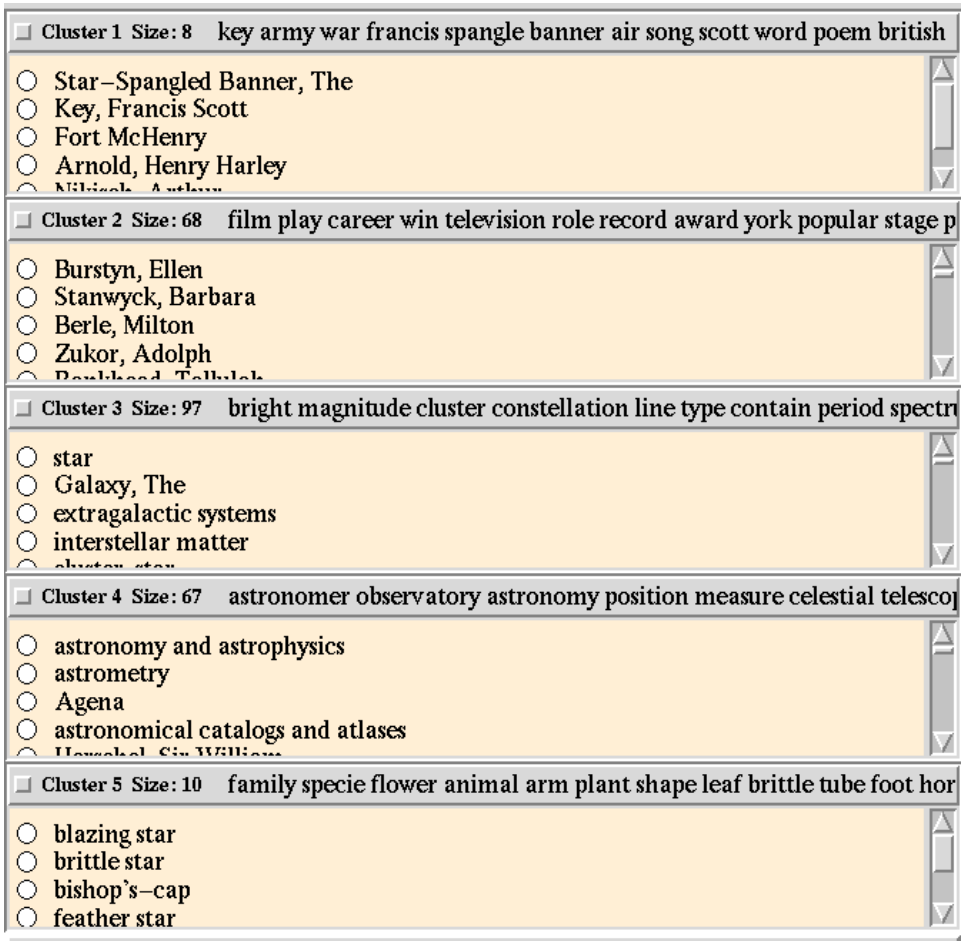

Figure 2.8 – Scatter/Gather : exemple de classification[1]

Des travaux de Hearst sur l'aide à la compréhension des résultats ont abouti à une interface appelée *TileBar* [Hearst, 1995]. Dans cette interface, l'auteur donne à l'utilisateur des informations concernant la relation entre les termes de la requête et ceux des documents présentés. Un exemple d'utilisation de cette interface est donné dans la Figure 2.9. On remarque que les documents sont découpés en thèmes (technique du *text tiling* [Hearst, 1997]). Pour chaque thème, on visualise la présence plus ou moins prononcée (nuance de gris) de chacun des termes de la requête. Le but de l'auteur est de présenter simultanément :

1. la longueur du document ;
2. la fréquence d'apparition des différents termes dans le document ;
3. la distribution des termes dans un document.

L'interface est fondée sur l'hypothèse suivante : un document est d'autant plus pertinent que tous les termes de la requête partagent un même thème commun.

[1] Figure provenant de la page : http://www.sims.berkeley.edu/~hearst/sg-example1.html.

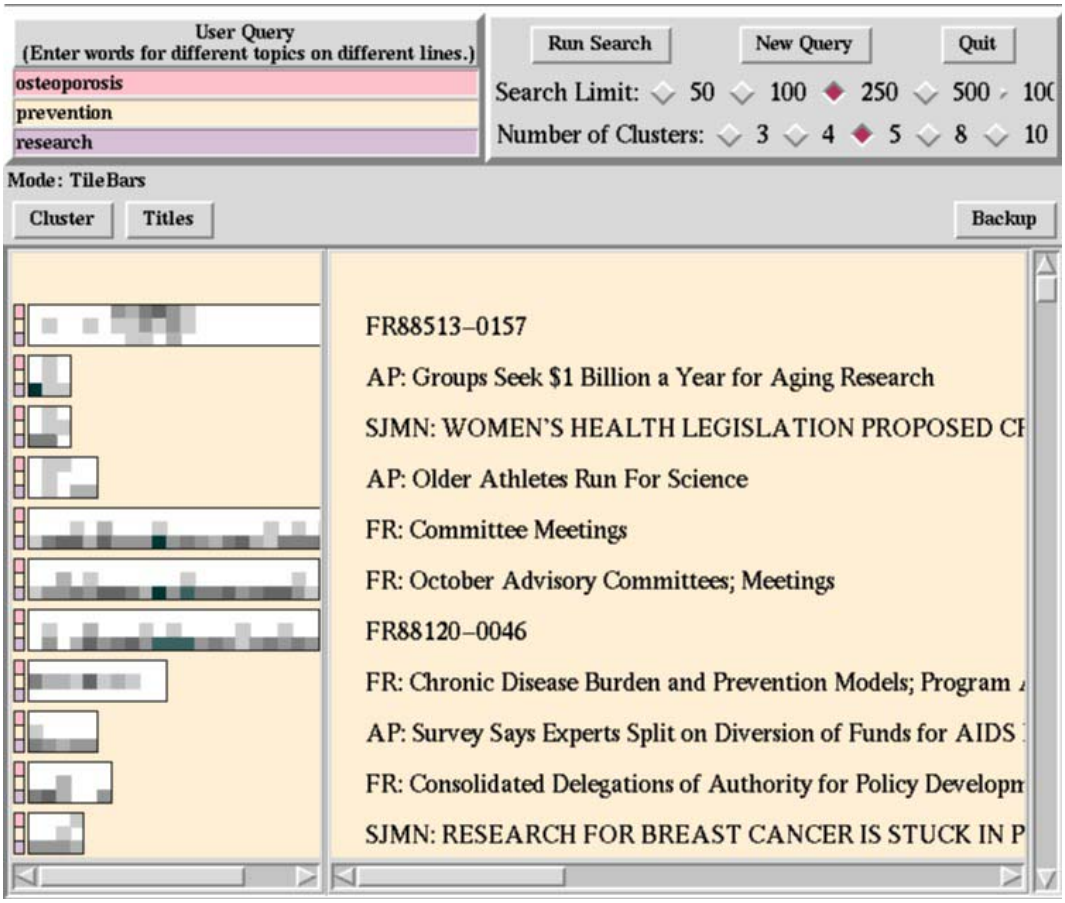

Figure 2.9 – TileBar : un exemple[1]

L'aide à la recherche d'information sur le Web émerge depuis quelques années et se manifeste par différentes approches tant sur la technologie employée que sur l'interface : cartographie, proposition de mots-clés, recherche similaire, etc.

Une des premières approches pour aider l'utilisateur à filtrer et à reformuler sa première requête est la technologie *LiveTopics*[2] [Bourdoncle, 1997]. L'algorithme, appelé « méthode des mots associés », utilise les cooccurrences pour créer des grappes de mots, à partir des distances physiques des mots dans le texte. Ces grappes sont censées représenter des catégories conceptuelles qui, présentées à l'utilisateur, permettront de le guider dans sa quête d'informations.

Depuis quelques années, une autre réalisation du même auteur a vu le jour sous le nom d'Exalead dont l'interface est présenté dans la Figure 2.10. Cette approche est intéressante car elle présente, pour une requête donnée req_init : une liste de mots-clefs associés ainsi qu'une liste de catégories. Cette approche permet en quelques cliques d'affiner la requête initiale : $req\_final = req\_init \ AND \ keyword$.

[1] Figure provenant de la page : http://www.sims.berkeley.edu/~hearst/tb-example.html.
[2] Cette technologie a été utilisée un temps par le moteur de recherche Altavista avec la fonction *Refined*.

Les mots-clefs sont soit des syntagmes nominaux, dans la plupart des cas, soit des syntagmes verbaux. Ces syntagmes verbaux sont dus apparemment à des erreurs d'indexation car l'objectif, d'après l'auteur, est de détecter uniquement des syntagmes nominaux. La profondeur du nombre de cliques est généralement de quatre au maximum. Au-delà, il n'y a plus de proposition de termes et moins de dix documents sont retournés en réponse à la requête.

L'approche Exalead a pour principal avantage de donner à l'utilisateur une vision rapide des concepts liés à sa requête. La présentation des concepts (des classes dans Scatter/Gather) sous forme de liste de termes nécessite un travail supplémentaire pour l'utilisateur. L'approche Exalead est donc un compromis intéressant entre l'activité demandée à l'utilisateur et l'aide fournie.

Voici quelques caractéristiques de l'outil :

- méthode statistique ;
- multilinguisme (aucun dictionnaire utilisé) ;
- détection de syntagmes nominaux, lemmatisation ;

et quelques caractéristiques de l'interface :

- ordre des mots de la requête pris en compte. Par exemple, les requêtes « contrat travail » et « travail contrat » ne donneront pas les mêmes mots-clefs ;
- la requête est assignée à un mot-clef si cela est possible. Par exemple, la requête « contrat travail » est assignée au mot-clef « contrat de travail », c'est-à-dire que la recherche se fait sur ce mot-clef et non sur la requête initiale.

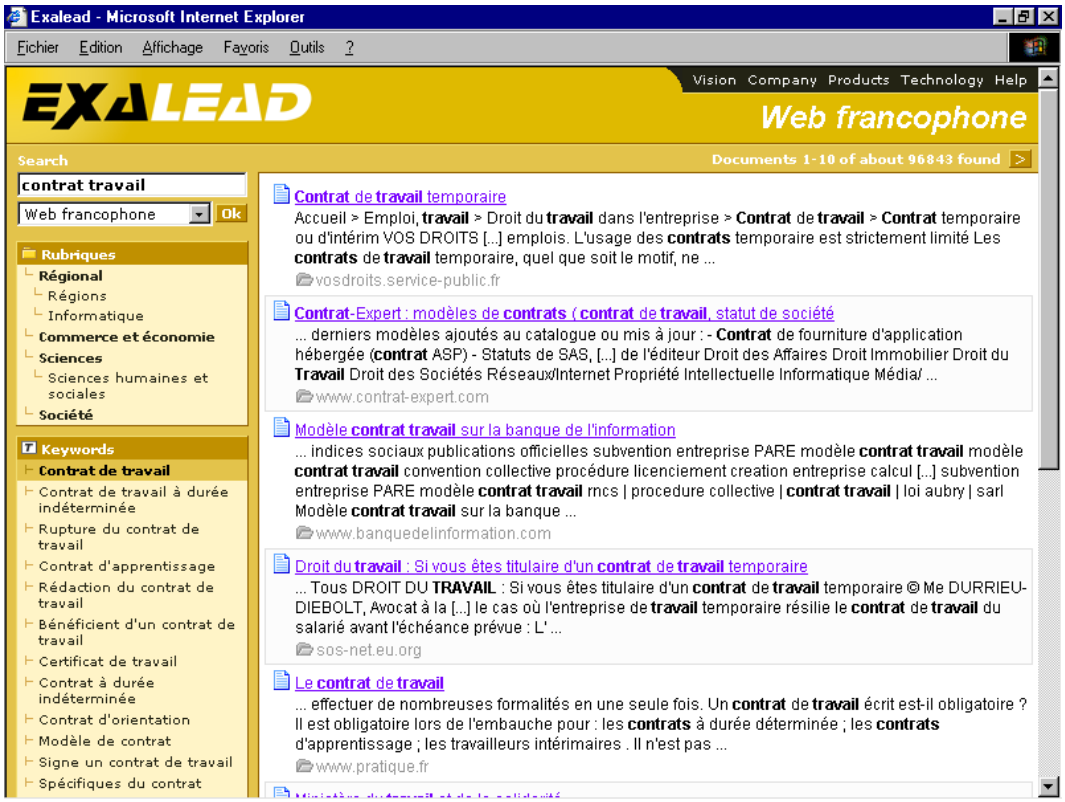

Figure 2.10 – Requête « contrat travail » sur Exalead

D'autres sites proposent ce type d'aide et d'interface, par exemple les moteurs Wisenut[1] ou Kartoo[2]. Ce dernier intègre, de plus, une interface sous forme de carte. Les sites qui répondent à la requête sont présentés sous forme de graphe où les nœuds sont les sites et les arcs représentent des termes communs entre les deux nœuds correspondant.

Dans le Tableau 2.1, on présente, pour ces différents moteurs offrant le même type d'aide, les réponses à la requête « contrat » qui est un mot juridique. Cet exemple n'est assurément pas représentatif de la puissance de chaque moteur. Pour chaque terme proposé par les différents moteurs, nous avons distingué les termes juridiques de ceux non juridiques. Pour cela, nous avons utilisé la base terminologique d'Eurodicautom (http://europa.eu.int/eurodicautom/Controller). On peut remarquer que le moteur Wisenut propose des termes en deçà des deux autres moteurs en terme de qualité. Pour Exalead et Kartoo, le nombre de termes juridiques proposés est quasiment équivalent. De plus, en terme de qualité de réponses, on peut considérer que ces moteurs donnent un résultat acceptable en prenant en compte le fait qu'ils sont non spécialisés. En revanche, en terme de quantité, ces moteurs sont limités. Enfin, on peut remarquer que les listes de termes juridiques proposées par ces deux moteurs sont totalement différentes.

Moteurs					
Exalead		Wisenut		Kartoo[3]	
Type de contrat	*	Du contrat		Contrat de qualification	*
Contrat d'apprentissage	*	Le contrat		Contrat social	*
Contrat d'association	*	Emailjob annuaire des sociétés		Jean Jacques Rousseau	
Complémentaire santé		Contrat type	*	Conseil régional	
Intitulé du poste				Groupe Darty	
				Paiement sécurisé	
				Contrat de travail	*
				Rédigés	
				Habitation	*
				…	

Tableau 2.1 – Résultats de la requête « contrat » sur Exalead, Wisenut et Kartoo

2.10 Utilisateur

Nous venons de parcourir les principales approches d'aide à la recherche d'information. Dans cette section, nous nous intéressons au comportement de l'utilisateur face au système de recherche. L'étude du comportement de l'utilisateur est intéressante car elle permet de donner des indications sur ses attentes et de mettre en évidence les lacunes éventuelles des systèmes de recherche.

[1] www.wisenut.com
[2] www.kartoo.com
[3] Pour ce moteur, nous n'avons pas énuméré tous les mots-clefs proposés (15 au total). En revanche tous les termes juridiques sont présentés dans ce tableau.

Cette étude du comportement est cependant difficile à mettre en place car les données nécessaires ne sont pas toujours suffisantes. Cette étude se fonde sur les fichiers de LOG des moteurs de recherche, ressource disponible la plus adéquate.

Dans [Silverstein et al., 1998], l'étude du fichier de LOG porte sur le moteur Altavista sur une période de six semaines, ce qui correspond environ à un milliard de requêtes sur un corpus totalisant 280 Go. Ce nombre de requêtes correspond à l'ensemble des requêtes posées comprenant les requêtes vides, les mêmes requêtes posées plusieurs fois, etc. Le nombre de requêtes non vides et uniques est d'environ 153000. Cet article permet de mettre en évidence différents aspects :

- 62,1 % des requêtes sont composées d'au plus deux termes pour une moyenne de 2.35 termes par requête ;
- environ 80 % des requêtes ne contiennent pas d'opérateur booléen (et, ou, sauf), pour une moyenne inférieure à un opérateur booléen par requête ;
- dans 78 % des sessions, seule une requête est posée, pour une moyenne de 2.02 requête par session ;
- dans 85 % des cas, seule la première page de résultats est consultée pour une moyenne de 1.39 ;
- dans 12 % des cas, la requête change d'un terme (par ajout ou retrait). Dans 35 %, la requête est complètement différente et, dans le reste des cas, la requête est modifiée.

A partir de ces statistiques, on peut donc en déduire que l'utilisateur utilise peu d'opérateurs (d'où l'importance de l'opérateur par défaut) et peu de termes dans la requête. De plus, il regarde dans la plupart des cas uniquement la première page de résultats (généralement dix par défaut) ce qui nécessite d'avoir une pertinence élevée pour les premiers résultats. Enfin, les dernières statistiques concernant la modification d'une requête montrent que les utilisateurs ont des difficultés à exprimer leurs besoins. Ces statistiques sont, néanmoins, anciennes et on peut naturellement penser que l'utilisateur type a évolué.

Dans cette optique, nous nous intéressons maintenant au comportement actuel de l'utilisateur, qui plus est sur le domaine juridique[1]. Au sein du Centre de recherche en informatique de l'Ecole des mines de Paris, nous disposons de fichiers de LOG d'interrogations qui portent sur différents moteurs. Nous avons choisi le fichier correspondant à l'une des versions[2] du moteur Pertimm : notre choix s'est orienté sur le fichier contenant des retours sur la plus grande période de temps. Ce fichier correspond à six mois d'utilisation pour un total de 18500 requêtes (soit une moyenne de 100 requêtes par jour uniquement pour ce moteur) pour 9902 requêtes distinctes.

[1] A noter qu'il n'existe pas d'études récentes dans la littérature. Toutefois, des statistiques sur les mots les plus couramment utilisés sur différents moteurs sont publiées.
[2] La version utilisée est celle disponible à l'adresse suivante : http://pertimm.ensmp.fr/cgi.

Dans le Tableau 2.2, la répartition des requêtes indique que le Journal officiel et les codes (dont la liste est donnée en Annexe A) sont les principales ressources demandées. Les codes, à la différence du J.O., sont disponibles sous forme de catégorisation hiérarchique d'articles : ce corpus est détaillé dans le chapitre 3. On peut donc en déduire, en vue de la demande sur cette ressource, que cette forme d'organisation de l'information est peu adéquate pour certains utilisateurs, c'est-à-dire pour les utilisateurs qui ne sont pas habitués au domaine. Nous donnons, à titre indicatif, le nombre de requêtes uniques c'est-à-dire celles posées une seule fois sur le moteur de recherche.

Type de ressources	#Requêtes (en %)	#Requêtes uniques (en %)
Journal officiel	41.53	35.48
Codes	54.03	62.48
Textes européens	4.44	2.04

Tableau 2.2 – Répartition des ressources consultées : J.O., codes, textes européens

Dans le Tableau 2.3, nous présentons la profondeur de la consultation, c'est-à-dire le nombre de pages de réponses consultées. On remarque que le pourcentage de consultation de la seule première page est équivalent au résultat trouvé par Silverstein. Le nombre de résultats par page est un paramètre de l'interface. La première page peut présenter 10 documents (par défaut et dans la plupart des cas) ou jusqu'à 50 documents par exemple.

Profondeur de consultation	#Requêtes (en %)
Première page	87.75
Seconde page	4.64
Troisième page	2.33
> troisième page	5.28

Tableau 2.3 – Nombre de pages de réponses consultées

Le Tableau 2.4 présente le nombre de répétitions des requêtes, sans considération de session. Pour les requêtes dont l'occurrence est supérieure à 1, une partie de la quantité provient des requêtes pour lesquelles plusieurs pages de réponses ont été consultées.

#Occurrence de la requête	#Requêtes (en %)
1 fois	61.26
2 fois	20.24
3 fois	9.28
> 3 fois	9.19

Tableau 2.4 – Répartition du nombre d'apparitions des requêtes

Dans le Tableau 2.5, on présente les caractéristiques des requêtes, en termes de mots par requête. Pour éviter l'influence des multiples pages de réponses consultées pour une même requête, les statistiques sont fondées sur les requêtes uniques. De plus, les requêtes ont été filtrées avec l'ensemble des mots vides suivants : « le la les un une des du de ». On remarque que 59,5% des requêtes sont composées de deux mots au plus, ce qui est quasiment équivalent au résultat de Silverstein.

#Mots par requête	#Requêtes uniques (en %)
1 mot	24
2 mots	35.05
3 mots	22.12
4 mots	11.30
> 3 mots	7.53

Tableau 2.5 – Répartition du nombre de mots par requête

La moyenne est de 2.51 mots par requête et le maximum est de 43 mots. Cette valeur est légèrement supérieure à celle trouvée par Silverstein. Cette valeur élevée de 43 mots se traduit dans la requête par la présence de toutes les références des codes :

$req\_init = mot_1\ AND$ K $\ ref_code_1\ AND\ ref_code_2\ AND$K .

Les modifications de requêtes sont du même ordre que celles présentées dans l'étude de 1998 : certaines requêtes sont modifiées d'une requête à l'autre par simple ajout ou suppression d'un mot. Nous constatons que certaines requêtes sont posées successivement sur les corpus. Tout ceci indique un manque de réponses pertinentes lors de la requête initiale.

Dans le Tableau 2.6, on présente les mots les plus cités sur l'ensemble des requêtes.

Mots
Code
Loi
Arrêté
Décret
Article
Travail
Droit
Sécurité

Tableau 2.6 – Les mots les plus cités.

En conclusion, on constate que le comportement de l'utilisateur n'a guère changé, au cours de ces dernières années, face à une interface classique de recherche d'information. Les difficultés de l'utilisateur à cerner son besoin et les méthodes pour améliorer une requête initiale restent identiques.

2.11 Conclusion

Dans ce chapitre, nous avons présenté les bases théoriques de la recherche documentaire et les traitements statistiques associés les plus couramment utilisés. La recherche documentaire regroupe plusieurs approches qui, dans certains cas, font intervenir l'utilisateur dans la quête des documents pertinents.

Les systèmes d'aide à la recherche d'information sont variés dans l'approche, la technique et l'interface. Le but de ces systèmes est de faire intervenir l'utilisateur pour améliorer leur performance. Cependant, cette aide doit être la plus simple et la plus efficace possible. L'approche utilisant la présentation de termes (et non de classes) et décrivant la requête nous semble la plus appropriée car elle est sans doute la plus rapide pour l'utilisateur. Elle est néanmoins difficile : les résultats sur Wisenut ne sont pas probants. C'est donc vers ce type d'aide que nous tournons notre recherche.

Chapitre 3

Techniques usuelles de classification

Résumé

La classification est un processus qui permet d'organiser un ensemble de données en classes cohérentes ou homogènes. Elle s'applique, a priori, sur n'importe quel type de données : tableau de contingence, tableau de distances, etc.

La classification se déroule, dans la plupart des cas, en trois étapes, et à l'aide de quelques paramètres indispensables : mesure de ressemblance, structure de la classification et type d'algorithme.

Il existe plusieurs catégories d'algorithmes de classification dont les deux plus utilisées sont les méthodes de partitionnement et les méthodes de classification hiérarchique. D'autres catégories existent, telles que les modèles probabilistes ou des modèles utilisant les liens hypertextes, et donc axées uniquement sur les documents Web.

3.1 Introduction

La classification est l'organisation d'un ensemble de données en *classes homogènes*. Elle a pour but de simplifier la représentation des données initiales. La classification automatique, appelée également classification non-supervisée (*clustering*), recouvre l'ensemble des méthodes permettant la construction automatique de telles classifications.

Les méthodes de classification ont donc un objectif précis : former des classes cohérentes (ou homogènes) et bien isolées [Govaert, 2003][1]. L'adjectif cohérent veut dire que les éléments appartenant à une classe partagent de nombreuses caractéristiques communes et donc se ressemblent fortement. Par isolée, on veut dire que deux classes ne se ressemblent pas, c'est-à-dire qu'elles ne partagent pas du tout les mêmes caractéristiques.

Les méthodes de classification se sont initialement développées d'un point de vue heuristique autour de méthodes optimisant des critères métriques. Ainsi, les deux algorithmes les plus couramment utilisés sont, d'une part, l'algorithme des centres mobiles (ou *k-means*) pour la recherche de partitions et, d'autre part, l'algorithme de classification hiérarchique ascendante de Ward pour la recherche de hiérarchies. Ces méthodes utilisent toutes deux le critère de l'inertie intraclasse.

De telles méthodes nécessitent de choisir une métrique mesurant la similarité entre les éléments de la collection à classer et un critère mesurant le degré de cohésion des classes et de séparation des classes.

D'autres approches algorithmiques sont apparues plus récemment, axées sur la statistique autour de modèles probabilistes de classification. Ces approches permettent de donner, par exemple, une meilleure interprétation des résultats sur de grandes collections comparée à celles des méthodes inférant des résultats à partir d'échantillons d'une collection.

Enfin, une approche algorithmique récente [Kleinberg, 1999], [Gibson et al., 1998] est applicable à des collections de documents Web, et utilise les caractéristiques propres à cette catégorie de collections. De plus, des approches hybrides [Pirolli et al., 1996], [Weiss et al., 1996] mêlant les techniques des méthodes de classification classique et des méthodes spécifiques au Web sont également apparues. Elles reposent sur le principe que le lien hypertexte a une importance linguistique forte.

Nous venons d'énoncer les axes algorithmiques principaux de la classification non-supervisée dont chacun possède des avantages, des inconvénients, des hypothèses d'application et enfin des interprétations de résultats. Au-delà de ces axes, l'un des récents

[1] Chapitre rédigé par Gildas Brossier, pp 235-262.

défis de la classification a été l'application des méthodes classiques sur des collections de type Web. Cette application induit une évolution de l'hypothèse de classification.

Dans ce chapitre, nous décrivons l'approche générique de la classification, qui peut se substituer à la plupart des méthodes, et l'hypothèse de classification. Nous présentons, ensuite, différentes approches de taxonomie possibles des méthodes, sur un fond de modèles mathématiques. Puis les différentes caractéristiques de la classification sont présentées. Enfin, nous décrivons différentes méthodes de classification suivant les algorithmes énoncés ci-dessus : les méthodes fondées sur les centres, les méthodes hiérarchiques ascendantes et descendantes, les méthodes fondées sur des modèles probabilistes, les méthodes fondées sur les liens hypertextes et, pour finir, les méthodes hybrides.

3.2 Approche générique

L'approche générique d'une méthode de classification automatique consiste en un ensemble de choix méthodologiques [Govaert, 2003] majeurs, énumérés ci-dessous :

- choix de la mesure de ressemblance entre les éléments à classer, c'est-à-dire choix d'une distance ou d'une mesure de similarité ;
- choix de la structure de classification : partition en K classes (et éventuellement choix de K), hiérarchie indicée, etc. ;
- choix de la méthode de classification.

D'autres choix que l'on peut considérer comme mineurs incluent :

- la sélection des attributs *pertinents* afin de représenter les éléments : ces attributs seront la base de la classification ;
- l'attribution d'un poids pour tous les attributs de chaque élément ;
- l'évaluation de la complexité en temps de calcul et de stockage des données ;
- la méthode de mise à jour des données et l'évaluation de la complexité en temps de calcul.

Les méthodes de classification comprennent trois étapes selon la Figure 3.1 [Govaert, 2003]. Ces trois étapes sont décrites ci-dessous.

- Le tableau de données initial X se présente généralement sous la forme d'une matrice d'individus x variable. La première étape du schéma est celle qui transforme ce tableau en une matrice de distance D ou une matrice de similarité.

- La deuxième étape du schéma général est la méthode algorithmique de la classification proprement dite. Elle correspond à la transformation de la matrice de distance D en une

matrice de distance *U* qui sera représentable par une structure de classification (partition, hiérarchie, arbre, pyramide).

- La troisième et dernière étape est la représentation des données, qui infère que chaque structure est liée à une distance appliquée sur la matrice *U*. De plus, on peut retrouver la matrice à partir de cette structure et avec une distance.

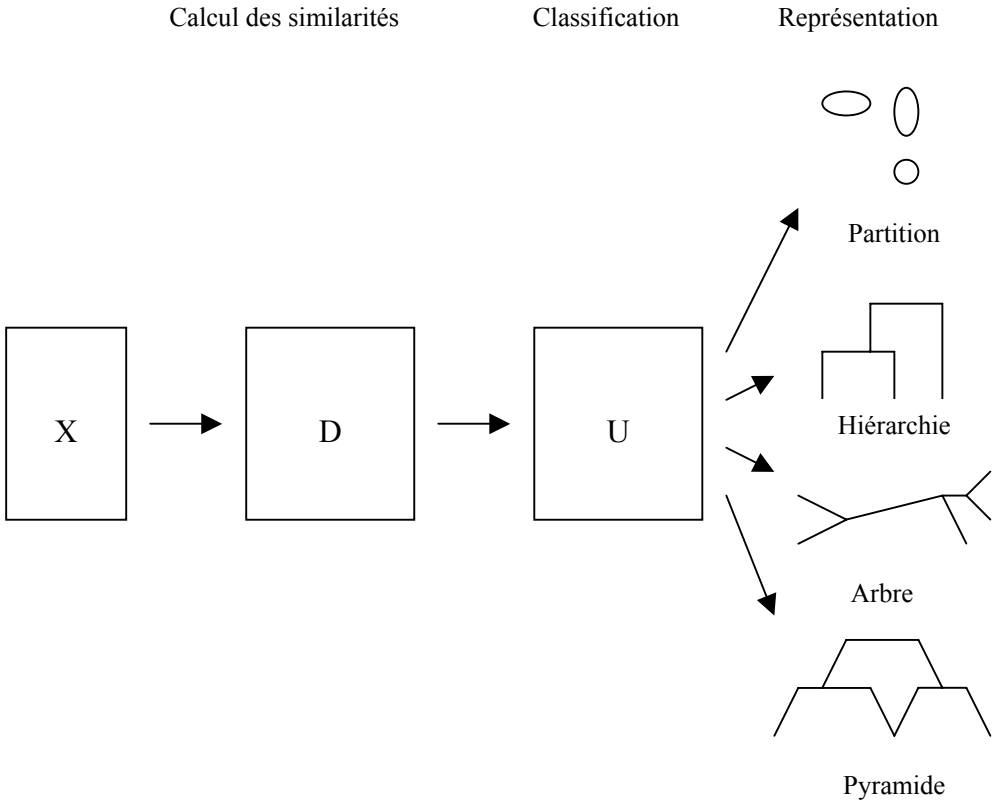

Figure 3.1 – Schéma général de la classification

3.3 L'hypothèse de classification

L'hypothèse de classification exprimée par Van Rijsbergen [1979] est directement liée à la recherche d'information ; elle est énoncée ci-dessous.

Hypothèse[1] : *les documents proches les uns des autres, en terme de distance, ont tendance à être pertinents pour une même requête.*

Cette hypothèse suggère qu'en partant d'une classification de tous les documents d'une collection (classification dite globale), on puisse retrouver, pour une requête donnée, la plupart des documents pertinents à partir des classes construites précédemment.

[1] *Closely associated documents tend to be relevant to the same request.*

D'autres hypothèses sont utilisées dans la recherche d'information. Certaines d'entre elles ont été révisées [Hearst & Pederson, 1996].

Ces hypothèses sont revues en détail dans le chapitre 3, et commentées par rapport à notre méthode de classification du chapitre 5.

3.4 Taxonomie des méthodes de classification

La taxonomie des algorithmes de classification non-supervisée a fait l'objet de plusieurs discussions dans la littérature [Jain et al., 1999], [Estivil-Castro, 2002]. La Figure 3.2 représente la taxonomie proposée par [Jain et al., 1999]. Cette taxonomie est très répandue dans la littérature.

Traditionnellement, les algorithmes de classification non-supervisée sont divisés en deux groupes : les algorithmes de partitionnement et les algorithmes de classification hiérarchique, ces derniers étant également divisés en deux sous-groupes : descendants et ascendants.

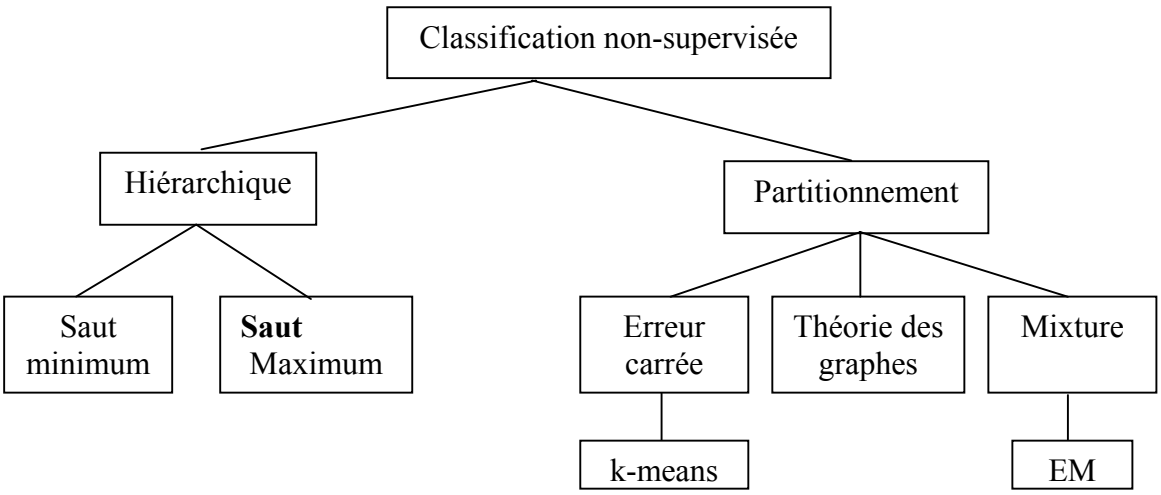

Figure 3.2 - Un exemple de taxonomie des algorithmes de classification non-supervisée

Dans [Estivill-Castro, 2002], l'auteur remet en cause la taxonomie proposée par [Jain et al., 1999]. Cependant, son but n'est pas de créer une nouvelle taxonomie mais de définir précisément les algorithmes afin de pouvoir les comparer entre eux. Ce dernier point évoqué est bien le cœur du problème, d'après l'auteur, dont l'objectif n'est pas, effectivement, de définir une taxonomie en soi. Dans un but de comparaison adéquat, un algorithme est défini, selon Estivill-Castro, par un « contexte » composé du couple de notions suivantes : le principe d'induction (la fonction d'objectivité) et le modèle (mathématique, structurel). Ainsi, ce couple de notions, s'il est correctement défini par les auteurs, permet de comparer les différents algorithmes entre eux. Son approche sur la définition des algorithmes regroupe

ainsi, à titre d'exemple, une partie des algorithmes de partitionnement dans une section intitulée « *representative-based clustering* » et diffère ainsi des taxonomies classiques.

Les principales caractéristiques liées à la classification non-supervisée, indépendamment des catégories de méthodes, sont les suivantes [Berkin, 2002] :

- type de données que l'algorithme peut gérer ;
- passage à l'échelle pour des ensembles de données de grande taille ;
- capacité à gérer des données de grandes dimensions ;
- capacité à trouver des classes de formes irrégulières ;
- complexité en temps de calcul ;
- complexité en stockage ;
- dépendance par rapport à l'ordonnancement des données ;
- type d'affectation des données : partitionnement (encore appelé « dur ») par opposition à appartenance graduelle (encore appelée « douce ») -voir § 3.5.1- ;
- confiance dans la connaissance acquise et dans les paramètres de l'utilisateur ;
- interprétation des résultats.

Cette liste est non-exhaustive ; d'autres caractéristiques peuvent être prises en compte comme, par exemple, la capacité à mettre à jour la classification par simple ajout de données.

Dans les sections suivantes, nous présentons les différentes approches de la classification et les différentes méthodes correspondantes. Pour ce faire, nous utilisons une autre taxonomie que celle de Jain mais tout aussi classique dans la littérature [Berkin, 2002]. Nous classons les méthodes dans 4 grandes catégories : les méthodes à partir de centres, les méthodes hiérarchiques, les méthodes probabilistes et les méthodes à partir de liens hypertextes.

Toutes les caractéristiques citées précédemment ne seront pas mentionnées pour les algorithmes décrits dans les sections suivantes ; seules quelques-unes parmi les plus importantes seront indiquées. De plus les différentes méthodes ne partagent pas toutes les caractéristiques énumérées plus haut : par exemple, les méthodes hiérarchiques ne peuvent gérer le passage à l'échelle.

3.5 Algorithmes de classification à partir de centres

Dans cette section, nous présentons un ensemble d'algorithmes pour lesquels les classes sont représentées par un centre. Ce dernier est soit une combinaison d'attributs des éléments de la classe, soit un sous-ensemble de la classe.

Ces algorithmes sont, pour la plupart, de type *k-means* et proposent généralement une partition *dure* comme résultat.

3.5.1 Partitionnement et appartenance graduelle

Les algorithmes de partitionnement permettent de créer, à partir d'un ensemble de documents, une partition *P* de *K* classes C_i avec $i \in [1, K]$. Ces *K* classes sont telles que :

$$\forall i \neq j, \qquad C_i \cap C_j = 0$$
$$\bigcup_i C_i = P$$

Les algorithmes d'appartenance graduelle permettent de créer une partition floue (on parle alors de classification floue) où les documents n'appartiennent pas une classe et une seule, comme dans une partition. En effet, chaque document possède un degré d'appartenance pour chaque classe. Il est cependant possible qu'un document ait un degré d'appartenance nul pour une ou plusieurs classes. Un exemple de partition floue est présenté Figure 3.3.

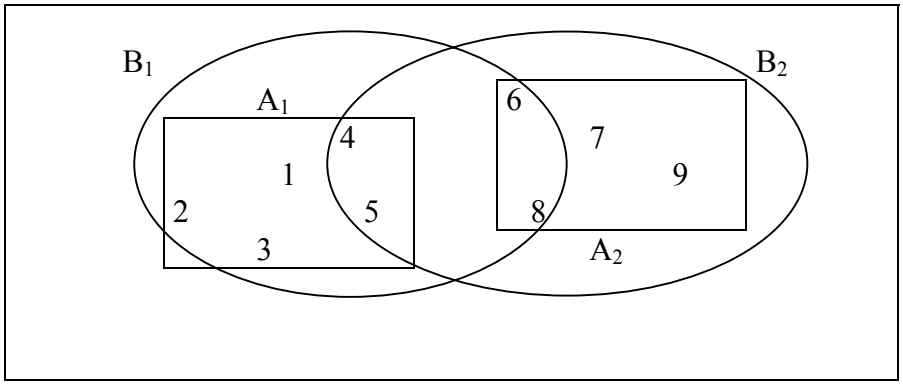

Figure 3.3 – Exemple de classification douce (les classes B_1 et B_2) et dure (les classes A_1 et A_2) avec *K*=2

Dans les algorithmes de classification à partir de centres, chaque classe est représentée par un centre qui peut être défini soit par un centroïde, soit par un médoïde (voir la section 3.5.2 pour une description des différents modèles de centre d'une classe). Le type de centre a une incidence sur le résultat de la partition finale. C'est aussi un critère de classification des différentes méthodes de partitionnement.

3.5.2 Représentation d'une classe

La représentation d'une classe est une description de l'ensemble des individus constituant la classe. Cette description, caractérisée par un ensemble de paramètres, permet de définir la forme et la taille de la classe dans un espace de données.

Dans les algorithmes de classification, la description d'une classe est simplement caractérisée par une combinaison linéaire des individus (également appelé centroïde ou *centroid* en anglais) ou un axe médian (médoïde ou *medoid* en anglais) ne donnant qu'une forme implicite, celle du centre de la classe. Une telle description permet d'assigner chaque

individu à un centroïde ou à un axe médian suivant une règle, qui est usuellement l'affectation d'un individu au centre qui lui est le plus proche.

3.5.2.1 Représentation avec les centroïdes

La représentation d'une classe *C* est définie comme étant la moyenne des éléments présents dans *C*. Plus précisément, un centroïde est un vecteur de termes pondérés pour lequel chaque composante correspond à la moyenne arithmétique des composantes d_i correspondantes, de tous les vecteurs d'individus présents dans *C*. Pour une classe *C* donnée, le vecteur du centroïde *V*(*C*) est défini par l'équation 2.1.

$$V(C) = \frac{1}{|C|} \sum_{i=1}^{|C|} d_i \tag{3.1}$$

Cette représentation de classe peut poser des problèmes dans certains cas :

- Les vecteurs du centroïde de chaque classe peuvent être très proches suivant le type de données représentant les documents. Par exemple, les documents représentés uniquement par un ensemble de mots impliqueront des vecteurs proches. On suppose que ce phénomène sera atténué pour les documents représentés uniquement par un ensemble de syntagmes nominaux.
- Le calcul des centroïdes s'avère une tâche coûteuse en temps de calcul et en stockage mémoire pour des corpus de grande taille. A noter qu'un seuil peut être appliqué aux vecteurs des centroïdes pour éliminer les composantes de poids faible [Salton, 1983]. Cette approche améliore le stockage des vecteurs (moins de données) et atténue ainsi l'effet de proximité des vecteurs. Cependant, le seuil peut être difficile à déterminer (dans [Salton, 1983], ce seuil est de 1).

3.5.2.2 Représentation avec les médoïdes

Un deuxième mode de représentation des classes consiste à prendre *k* individus parmi tous les individus d'une classe. Ces *k* éléments sont centraux vis à vis de la classe, c'est-à-dire qu'ils sont proches du centre géométrique de la classe. Cette représentation permet de représenter la classe non pas par un point unique mais par un *noyau* censé définir au mieux la classe. Le choix de la valeur de *k* est empirique (dans [Diday et al., 1982], $k = 3$). Le choix des éléments du noyau est fondé sur un calcul de distances entre tous les individus de la classe. Pour une classe C_j donnée, le noyau $N(C_j)$ correspond aux individus qui ont la plus petite somme de distances par rapport aux autres individus de la classe.

Pour une classe C_j, *X* est un élément du noyau s'il minimise l'inertie I_j définie par l'équation (3.2) :

$$I_j = \sum_{c \in C_j} \mu_j d^2(X, c) \tag{3.2}$$

avec μ_j un poids.

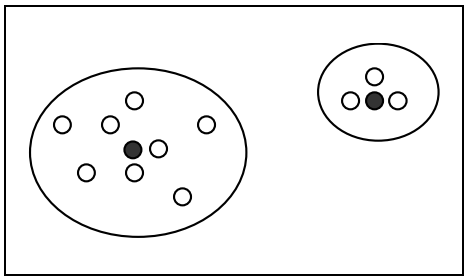

Figure 3.4 – Exemple de centroïde

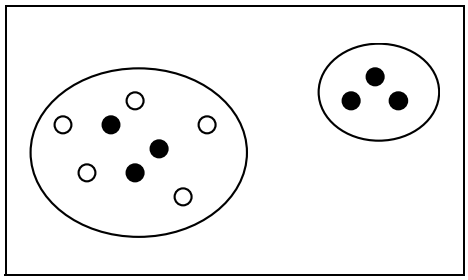

Figure 3.5 – Exemple de médoïde

A noter que le choix de la valeur de *k* doit être supérieur à 1 pour éviter d'assigner des documents dans une classe basée sur des termes trop spécifiques à l'individu.

3.5.3 Autres Caractéristiques

- Les algorithmes de partitionnement sont généralement itératifs, c'est-à-dire que plusieurs passes sont nécessaires pour obtenir une convergence de l'algorithme. Il existe cependant des algorithmes du type « *single-pass* » qui nécessitent une seule et unique itération.
- La convergence d'une méthode peut être caractérisée de différentes façons.
- Ces algorithmes ont généralement une complexité en temps de calcul linéaire.
- La partition finale dépend de la partition initiale.

3.5.4 Catégories

Il existe deux ensemble de méthodes de partitionnement :

- le premier regroupe les méthodes K-centroïdes telles que la méthode k-means, les méthodes fondées sur k-means ainsi que la méthode des centres mobiles ;
- le second regroupe les méthodes K-médoïdes telles que la méthode des nuées dynamiques.

Dans les sections suivantes, nous présentons quelques méthodes fondées sur les centroïdes, dont la plus connue est k-means, ainsi que celles fondées sur les médoïdes.

3.5.5 Algorithme k-means

L'algorithme k-means a été proposé par MacQueen [MacQueen, 1967] et repose sur la méthode de Forgy [Forgy, 1965]. L'algorithme de base est décrit ci-après :

Pour une valeur *K* donnée :

1. sélectionner les *K* centroïdes initiaux ;
2. affecter chaque individu au centroïde le plus proche ;
3. recalculer les centroïdes ;
4. répéter 2 et 3 tant que le critère d'arrêt n'est pas respecté.

Algorithme 3.1 – K-means

Les centroïdes des classes sont calculés après chaque affectation d'un individu à une classe.

La complexité en temps de calcul de l'algorithme est linéaire en $O(I \cdot K \cdot m)$ où *I* est le nombre d'itérations nécessaires pour la convergence, *K* le nombre de centres et *m* le nombre de documents. Le nombre d'itérations est relativement faible (moins de 10 généralement). Si le nombre de classes est très inférieur au nombre de documents (comme dans la plupart des cas), dans ce cas la complexité devient $O(m)$.

Un inconvénient de cette méthode est que la partition finale dépend de la partition initiale. Le calcul des centroïdes, après chaque affectation d'un individu, influence le résultat de la partition finale. En effet, ce résultat dépend de l'ordre d'affectation des documents.

Un autre inconvénient de cette méthode est que le nombre de classes est un paramètre de l'algorithme. Le choix des centres initiaux est également problématique puisqu'il influence la partition initiale :

- La première approche pour choisir les centres initiaux est simplement le choix aléatoire, qui donne généralement des résultats mitigés ;
- une seconde approche est de prendre les *K* premiers documents du corpus dans l'ordre d'affectation.

Plusieurs travaux ont été menés pour pallier ces deux inconvénients majeurs de la méthode. L'algorithme X-means [Pelleg et Moore, 2000] (décrit dans la section 3.5.8) est une extension de k-means qui permet de calculer la valeur de *K*. Dans [Bradley et al., 1998], l'algorithme étend k-means avec notamment des améliorations pour rendre les résultats moins dépendants de l'ordre d'affectation des documents.

Finalement, l'algorithme k-means est très populaire du fait qu'il est très facile à comprendre et à mettre en œuvre. Le degré d'appartenance d'un document à une classe étant binaire et la pondération de chaque document étant constante, cela facilite son utilisation dans d'autres domaines de recherche tels que la génomique [Nédellec et al., 2001], par exemple.

3.5.6 Algorithme fuzzy k-means

L'algorithme fuzzy k-means (FKM) [Bezdek, 1981] est proche de l'algorithme k-means, à la différence près que FKM utilise une fonction d'appartenance graduelle au lieu d'une fonction de partitionnement.

$$FKM\left(X,C\right)=\sum_{i=1}^{n}\sum_{j=1}^{k}u_{ij}^{r}\left\|x_i-c_j\right\|^2 \tag{3.3}$$

avec $\sum_{j=1}^{k}u_{ij}=1,\forall i$ et $u_{ij}\geq 0$ et avec $r\geq 1$, où r est le degré de regroupement entre classes et C l'ensemble des centres.
Le paramètre u_{ij} représente le degré d'apparte nance du document x_i au centre c_j.

3.5.7 Algorithme K-Harmonic means

L'algorithme « K-Harmonic means » (KHM) [Zhang, 2000] est similaire à k-means uniquement dans le sens où cette approche est une méthode itérative basée sur les centres. Le but de cette approche est de pallier le problème de l'initialisation des centres, que l'on rencontre pour les méthodes k-means ou EM (voir section 3.6), par exemple. Cet algorithme diffère notamment de k-means par le critère d'optimisation (ou fonction d'objectivité). En effet, ce critère est fondé sur la moyenne harmonique de la distance de chaque document avec tous les centres :

$$KHM\left(X,C\right)=\sum_{i=1}^{n}\frac{k}{\sum_{j=1}^{k}\frac{1}{\left\|x_i-c_j\right\|^p}} \tag{3.4}$$

Cette fonction d'objectivité donne un bon score (c'est-à-dire un poids faible) pour tout document qui se trouve proche d'au moins un centre. Cette fonction se comporte comme la fonction *min* de k-means. Ceci est la propriété voulue de cette fonction pour mesurer la qualité des classes retrouvées. A noter que *p* est un paramètre de l'algorithme.

La seconde différence avec k-means est l'appartenance graduelle m_{KHM} d'un document à une classe et l'utilisation d'une fonction de pondération dynamique w_{KHM} pour les documents qui tend à donner un poids élevé pour les documents éloignés de tous centres. Cette fonction est dite dynamique car elle est attribuée à tous les documents après chaque itération.

$$m_{KHM}\left(c_j \mid x_i\right) = \frac{\left\|x_i - c_j\right\|^{-p-2}}{\sum_{l=1}^{k}\left\|x_i - c_l\right\|^{-p-2}} \tag{3.5}$$

$$w_{KHM}\left(x_i\right) = \frac{\sum_{j=1}^{k}\left\|x_i - c_j\right\|^{-p-2}}{\left(\sum_{j=1}^{k}\left\|x_i - c_j\right\|^{-p}\right)^2} \tag{3.6}$$

La complexité en temps de calcul et en stockage est proche de k-means et EM. La différence se fait au niveau de la convergence de l'algorithme.

3.5.8 Algorithme X-means

L'algorithme X-means [Pelleg et Moore, 2000] est fondé sur k-means avec une extension pour tenter de déterminer la meilleure valeur du nombre de classes *K*. X-means comporte une succession de 2 étapes principales. La première est une simple application de k-means jusqu'à l'obtention de la convergence de celui-ci. La seconde, correspondant à l'extension de l'algorithme, détermine le nombre de classes existantes à se diviser en deux. Cette étape utilise le critère BIC (*bayesian information criterion*) sur un ensemble de classes. Les classes finalement divisées sont celles qui optimisent la valeur du critère BIC. Cet algorithme permet de couvrir un intervalle de valeur possible de *K* et de choisir la meilleure valeur parmi cet intervalle.

Voici à présent, une liste de méthodes utilisant les axes médians.

3.5.9 L'algorithme avec simple passe (« *single-pass* »)

1. Choisir le noyau de la première classe (par exemple le premier document du corpus).
2. Calculer la distance entre un document non assigné et le noyau de chaque classe. Garder la plus petite distance.
3. Si cette distance est plus petite qu'un seuil alors
   Affecter ce document à la classe correspondante.
   Recalculer le noyau de cette classe.
   Sinon
   Créer une nouvelle classe avec le document comme noyau.
4. Répéter 2 et 3 tant qu'il reste des documents non affectés.

Algorithme 3.2 – Simple passe

Les avantages de cette méthode sont, d'une part, sa simplicité et sa rapidité et, d'autre part, le fait qu'une seule passe sur les données est nécessaire. On peut ajouter que cette méthode peut être utile comme point de départ d'autres méthodes.

L'inconvénient est que la partition finale dépend de l'ordre dans lequel les documents sont affectés.

Les résultats aboutissent généralement à la création de classes de grande taille.

3.5.10 La méthode des nuées dynamiques

La méthode des nuées dynamiques a été développée par Diday [Diday et al., 1982] et reprise dans [Saporta, 1990]. Pour cette méthode, les centres ne sont pas représentés par un seul élément comme pour la méthode des centres mobiles ou par un centroïde mais par un noyau composé de *d* documents. Ce noyau est censé être plus représentatif de la classe qu'un centroïde. L'algorithme est alors le suivant :

1. déterminer une partition initiale ;
2. calculer le noyau de toutes les classes ;
3. affecter chaque document à la classe qui lui est la plus proche ;
4. répéter 2 et 3 tant que la convergence n'est pas atteinte.

Algorithme 3.3 – Nuées dynamiques

Une des propriétés de cet algorithme est que chaque partition obtenue (i.e. pour chaque itération) est indépendante de l'ordre dans lequel les documents sont affectés. Cette propriété est liée au fait que les noyaux des classes sont calculés une fois par itération et que, pour chaque itération, tous les documents sont affectés.

Une autre propriété de cette méthode est sa faible sensibilité au choix de la distance utilisée [Schütze et Siverstein, 1997].

3.5.11 La méthode Scatter/Gather

La méthode Scatter/Gather [Cutting et al., 1992] ne peut pas être définie uniquement comme une méthode de classification de documents. En effet, l'idée prédominante des auteurs de cette méthode est d'utiliser une méthode de classification pour améliorer l'accès à l'information : « *This technique is directed towards information access with non-specific goals and serves as a complement to more focused techniques.* ». Ainsi, les auteurs ont développé une méthode de classification pour l'utiliser en tant qu'outil de navigation qui fait intervenir le jugement de l'utilisateur dans le processus de recherche d'information. Cette méthode est de plus interactive, c'est-à-dire que les utilisateurs choisissent un ensemble de classes parmi celles proposées et pour lequel un nouvel ensemble de classes sera généré dynamiquement. Cette approche a pour objectif d'améliorer successivement la précision des réponses (l'interface est présentée dans le chapitre 1).

La méthode combine plusieurs notions telles que la navigation interactive, la classification dynamique et le retour d'informations. Le fonctionnement global de la méthode

est le suivant : à partir d'une requête, un ensemble de classes (« *Scatter* ») est présenté à l'utilisateur. Les classes sont générées à partir d'un ensemble de *n* documents retourné par un moteur de recherche en réponse à une requête. L'utilisateur choisit ensuite les classes qui lui semblent pertinentes (« *Gather* »). Puis ces classes sont à nouveau classées. Le mécanisme est itératif. Chaque classe présentée à l'utilisateur est définie par un ensemble de mots centraux (désigné dans l'article comme « *cluster digest* »). Cet ensemble de mots est supposé résumer au mieux la classe : ce sont les *m* mots qui apparaissent le plus fréquemment dans la classe. Le paramètre *m* peut être soit défini de façon adaptative en fonction du nombre de documents *d* d'une classe, soit prédéfini.

Les auteurs définissent deux nouveaux algorithmes pour la méthode Scatter/Gather : *Buckshot* et *Fractionation*. Ces algorithmes sont tous deux conçus pour déterminer les centres initiaux. Bien que nouveaux, ces algorithmes utilisent néanmoins comme support un algorithme de classification hiérarchique mais de façon locale, c'est-à-dire sur des petits ensembles de documents : l'algorithme de saut moyen (décrit dans la section 3.7.8). L'utilisation d'un tel algorithme est justifiée par les auteurs, qui supposent qu'il existe des algorithmes de classification donnant de meilleurs résultats mais qui tournent plus lentement.

L'algorithme *Buckshot* applique l'algorithme de saut moyen sur un échantillon de documents choisis aléatoirement pour obtenir *K* classes. L'inconvénient de cette approche est la dépendance des résultats en fonction du choix aléatoire des documents c'est-à-dire que l'on peut obtenir des partitions finales différentes, et donc des centres différents, pour chaque application de l'algorithme sur le même corpus.

L'algorithme *Fractionation* commence par scinder le corpus en *p* groupes de même taille $\left(\frac{|C|}{p}\right)$ avec $p > K$. Sur chacun de ces groupes, l'algorithme de saut moyen est appliqué. Le processus est ensuite appliqué itérativement sur la nouvelle partition jusqu'à l'obtention de *K* classes. Cet algorithme donne de meilleurs résultats que *Buckshot*, mais ce dernier est en revanche plus rapide. L'algorithme *Fractionation* est donc utilisé pour déterminer les centres initiaux et l'algorithme *Buckshot* est utilisé dans la fonction d'affectation.

La méthode Scatter/Gather détermine, dans un premier temps, les *K* centres initiaux puis, dans un second, affecte chaque document au centre qui lui est le plus proche. Le centre d'une classe est défini comme le centroïde de la classe des *d* documents les plus centraux de la classe. Une méthode d'affectation composée de trois étapes est ensuite appliquée, jusqu'à l'obtention d'une condition d'arrêt. La première de ces trois étapes consiste à recalculer les centres et à affecter tous les documents aux nouveaux centres. La seconde étape divise chaque classe obtenue en deux classes en utilisant l'algorithme Buckshot (avec $K = 2$ pour chaque classe). La dernière étape permet de regrouper les classes qui sont très similaires.

3.5.12 Exploitation des optima locaux

Toutes les méthodes décrites précédemment ne possèdent pas de méthodes d'optimisation globale. Généralement, les méthodes de classification ne possèdent que des méthodes d'optimisation locale. Les résultats sont donc différents suivant l'initialisation des classes ou des centres. Plusieurs approches sont possibles pour exploiter cette caractéristique :

On applique l'algorithme plusieurs fois c'est-à-dire sur des partitions initiales différentes à chaque fois. On obtient ainsi en résultat des partitions finales différentes que l'on peut exploiter suivant deux façons. La première est de garder uniquement la partition qui optimise le critère (ou fonction d'objectivité). La seconde est de déduire de ces ensembles les « formes fortes » (*cf*. Figure 3.3). Les formes fortes sont les ensembles de documents que l'on retrouve dans les mêmes classes, quelle que soit la partition initiale.

3.5.13 Choix du nombre de centres

Le critère d'inertie utilisé, par exemple, dans k-means ou dans les nuées dynamiques est lié au nombre de classes et donc au nombre de centres. Le critère d'inertie vaut zéro si chaque document du corpus est présent dans une classe singleton. Ceci permet de trouver la meilleure partition en optimisant le critère d'inertie. Le choix du nombre de centres est donc primordial pour déterminer la « meilleure » partition finale. Il existe différentes approches pour tenter de résoudre ce problème récurrent :

- K est connu ;
- on impose des contraintes en ce qui concerne les classes : nombre maximum de documents par classe par exemple ;
- on effectue plusieurs classifications avec des valeurs différentes de K (intervalle de valeurs) et on détermine la partition qui minimise le critère d'inertie ;
- on définit un critère pour détecter automatiquement le nombre de classes. Par exemple, le critère BIC dans X-means ;
- on définit une méthode de classification indépendante du nombre de centres. Par exemple la méthode K-Harmonic.

3.6 Modèles probabilistes

L'hypothèse faite dans toute approche probabiliste de la classification automatique est de considérer que les données forment un échantillon aléatoire X_1, K, X_n, issu d'une population, et de s'appuyer sur l'analyse de la distribution de probabilité de cette population pour définir une classification [Govaert, 2003]. Le modèle probabiliste le plus connu est représenté par l'algorithme EM.

L'algorithme EM (*Expectation-Maximization*) est une technique itérative de maximisation de la loi de vraisemblance en présence de données incomplètes [Dempster et al., 1977]. Nous n'en donnons ci-dessous qu'un aperçu.

Il alterne successivement deux phases (E et M).

- Phase E : calcul des probabilités d'appartenance des X_i aux classes conditionnellement au paramètre courant.
- Phase M : phase de maximisation de la vraisemblance.

L'une des propriétés de l'algorithme est d'améliorer la vraisemblance des paramètres après chaque itération, conduisant ainsi à la convergence.

Une variante de cet algorithme, l'algorithme CEM, ajoute une phase classifiante aux deux étapes principales [Celeux, 1992].

Les différentes méthodes suivant des approches paramétriques et non paramétriques sont détaillées dans [Govaert, 2003], [Saint-Jean, 2001].

3.7 Algorithmes hiérarchiques

Le but de la classification hiérarchique est de construire une série de partitions de classes sous forme hiérarchique dont l'ensemble est caractérisé par un dendogramme (arbre hiérarchique indicé). Les documents et les classes sont respectivement représentés dans un dendogramme par les feuilles et les nœuds (*cf*. Figure 3.6).

Il existe deux approches pour construire un dendogramme :

- La classification agglomérative (ou ascendante) : chaque document est initialement affecté à une classe singleton. A chaque étape, on associe les 2 classes les plus similaires jusqu'à l'obtention d'une racine (classe qui regroupe toutes les classes).
- La classification divisive (ou descendante) : à partir d'une classe racine (regroupant tous les documents), une classe sera subdivisée à chaque étape jusqu'à l'obtention d'un ensemble de classes singletons. A chaque étape, il est nécessaire de choisir la classe qui sera subdivisée.

Les méthodes descendantes sont peu utilisées dans la littérature. Pour cette raison, nous nous attacherons principalement qu'aux méthodes ascendantes.

3.7.1 Algorithme générique

La plupart des méthodes de classification hiérarchique ascendante peuvent être décrites par l'algorithme générique ci-dessous [El-Hamdouchi et Willet, 1986] :

1. Calculer les $\frac{p(p-1)}{2}$ distances entre les p classes prises 2 à 2 et les stocker dans une matrice de distances.
2. Regrouper les 2 classes les plus similaires.
3. Recalculer les distances de la matrice entre la nouvelle classe et toutes les autres classes.
4. Répéter 2 et 3 tant qu'une classe racine n'est pas obtenue.

Algorithme 3.4 – Algorithme générique de classification hiérarchique

La première étape de cet algorithme générique, qui consiste à agréger 2 classes, donne lieu à de multiples méthodes : les méthodes du saut minimum, du saut maximum, du saut moyen de groupe, de Ward et la méthode des centroïdes. Ces différentes méthodes sont décrites dans les sections suivantes. Les méthodes de classification hiérarchique diffèrent dans l'interprétation de la notion de classes proches.

3.7.2 Formule de Lance-Williams

L'étape 2 de l'algorithme générique peut être calculée en utilisant la « formule de mise à jour » de Lance-Williams [Lance et Williams, 1967]. La distance entre la nouvelle classe C_r composée des classes C_i et C_j et toutes les autres classes C_k est définie de la façon suivante :

$$\forall C_k, \quad d(C_r, C_k) = a(C_i)\,d(C_i, C_k) + a(C_j)\,d(C_j, C_k) + b\,d(C_i, C_j) + c\left|d(C_i, C_k) - d(C_j, C_k)\right| \tag{3.7}$$

En d'autres termes, le calcul de toutes les distances, entre une nouvelle classe C_r composée de 2 classes C_i et C_j et toutes les autres classes C_k, est ainsi vu comme une fonction linéaire de la distance entre la classe C_k et les 2 classes C_i et C_j.

La formule de Lance-Williams s'applique à toutes les méthodes de classification hiérarchique énoncées ci-après. Pour ce faire, ces méthodes peuvent être définies à l'aide d'un ensemble de 3 coefficients (à une distance d donnée entre 2 classes). Le Tableau 3.1 résume l'ensemble des coefficients pour chaque méthode.

Méthode	$a(l)$	b	c
Saut minimum	$1/2$	0	$-1/2$
Saut maximum	$1/2$	0	$1/2$
Saut moyen de groupe	$\frac{\lvert l\rvert}{\lvert C_i\rvert+\lvert C_j\rvert}$	0	0
Centroïdes	$\frac{\lvert C_i\rvert}{\lvert C_i\rvert+\lvert C_j\rvert}$	$-\frac{\lvert C_i\rvert\cdot\lvert C_j\rvert}{\left(\lvert C_i\rvert+\lvert C_j\rvert\right)^2}$	0
Médiane	$1/2$	$-1/4$	0
Variance minimum (Ward)	$\frac{\lvert l\rvert+\lvert C_k\rvert}{\lvert C_i\rvert+\lvert C_j\rvert+\lvert C_k\rvert}$	$-\frac{\lvert C_k\rvert}{\lvert C_i\rvert+\lvert C_j\rvert+\lvert C_k\rvert}$	0

Tableau 3.1 - Paramètres de la formule de Lance-Williams pour différentes méthodes

3.7.3 Complexité

Bien que la formule de Lance-Williams permette de mettre à jour facilement la matrice des distances, les méthodes de classification hiérarchique n'en sont pas moins très coûteuses en temps de calcul et en stockage. La construction d'un dendogramme nécessite de calculer initialement toutes les distances entre les documents pris deux à deux. Ce calcul se fait en $O\left(p^2\right)$ avec p correspondant au nombre de documents à classer. Si le nombre de documents à classer est important, il devient impossible de stocker les p^2 distances en mémoire. Il faut alors recalculer toutes les distances nécessaires (celles qui n'ont pu être gardées en mémoire) à chaque étape de la construction de la hiérarchie. Ce calcul, à la volée, d'une partie des distances à chaque étape augmente de façon significative le temps de calcul. Ce facteur limite l'utilisation de ces algorithmes pour des corpus de grande taille.

Pour mieux comprendre les différentes méthodes d'agrégation de 2 classes qui ont été citées précédemment, chaque méthode est décrite à partir de la section 3.7.5.

3.7.4 Représentation d'un dendogramme

Un dendogramme est un arbre hiérarchique H sur lequel on applique un indice défini par une fonction f croissante définie de la façon suivante :

$$f : H \rightarrow R+$$
$$\forall h \neq h', \qquad h \subset h' \Rightarrow f(h) < f(h')$$

Dans ce cas, cet arbre hiérarchique est également appelé une *hiérarchie indicée*. Une telle hiérarchie est illustrée par la Figure 3.6.

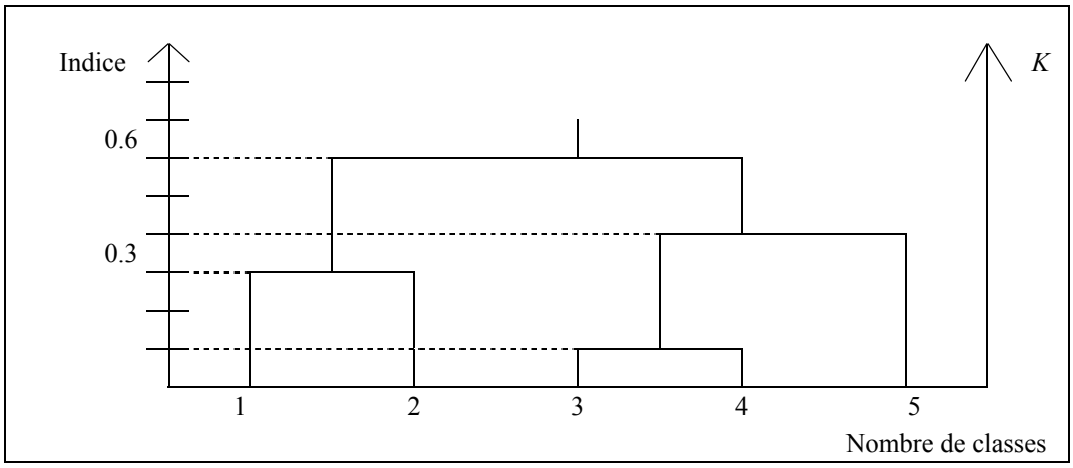

Figure 3.6 - Représentation d'un dendogramme

Cette forme de représentation facilite la visualisation d'une hiérarchie. L'indice permet de retrouver la partition pour une valeur K donnée ou pour une valeur de cet indice. En d'autres termes, une partition est obtenue par coupure horizontale du dendogramme, pour une valeur d'indice donnée. Par exemple, si l'on désire obtenir deux classes, le résultat sera obtenu, en coupant le dendogramme de la Figure 3.6, une partition composée des ensembles {1,2} et {3,4,5}. Par contre, si l'on veut la partition correspondant à une valeur d'indice égale à 3.5, alors on obtiendra les ensembles {1,2}, {3,4} et {5}.

3.7.5 Saut minimum

La méthode du saut minimum [Croft, 1977] (également appelée lien simple ou *single link*) regroupe en une classe C_r deux classes C_i et C_j ayant une distance minimum entre elles. Le saut minimum correspond à la plus petite distance entre un document appartenant à la première classe et un document appartenant à la deuxième. La mise à jour des distances, entre la nouvelle classe et les autres, est peu coûteuse pour cette méthode. On utilise uniquement les distances qui ont été calculées à la première étape en prenant :

$$\forall C_k, \qquad d(C_r, C_k) = \min(d(C_i, C_k), d(C_j, C_k)) \tag{3.8}$$

L'inconvénient de cette approche est qu'elle est très sensible au bruit. En revanche, les avantages de cette méthode sont multiples : la mise en œuvre de l'algorithme est simple, les classes ne sont pas représentées (par un centroïde par exemple) ce qui ne nécessite pas de recalculer la matrice de similarité.

Cette méthode a tendance à créer un petit nombre de classes de grande taille.

3.7.6 Arbre de couverture minimum

La construction d'une hiérarchie de type saut minimum (décrite ci-dessus) est fortement liée aux algorithmes qui ont été développés pour la recherche de l'arbre de couverture minimum (MST ou Minimum Spanning Tree) pour un graphe *G* donné. On peut citer de façon non exhaustive les algorithmes connus de Kruskal [Kruskal, 1956] et de Prim [Prim, 1957]. Les données d'un arbre de recouvrement minimum sont suffisantes soit pour construire une hiérarchie de liens simples ou à partir des données initiales de *G*, on peut construire simultanément, soit un arbre de recouvrement minimum, soit une hiérarchie de liens simples. Les classes de la méthode du saut minimum sont des sous-graphes de l'arbre de recouvrement minimum.

L'algorithme est le suivant :

1. Débuter l'arbre minimum avec les deux sommets les plus proches.
2. Déterminer le sommet qui est le plus proche de n'importe quel sommet de l'arbre minimum.
3. Ajouter ce sommet à l'arbre minimum.
4. Répéter 2 et 3 tant qu'il reste des sommets non connectés à l'arbre minimum.

Algorithme 3.5 – Arbre de couverture minimum

Ce lien avec l'arbre de recouvrement minimum permet aussi de mettre en évidence un défaut connu appelé « effet de chaîne ». Ce dernier peut regrouper deux sommets éloignés assez rapidement dans la hiérarchie s'il existe une chaîne de sommets reliant les deux sommets avec une distance totale faible.

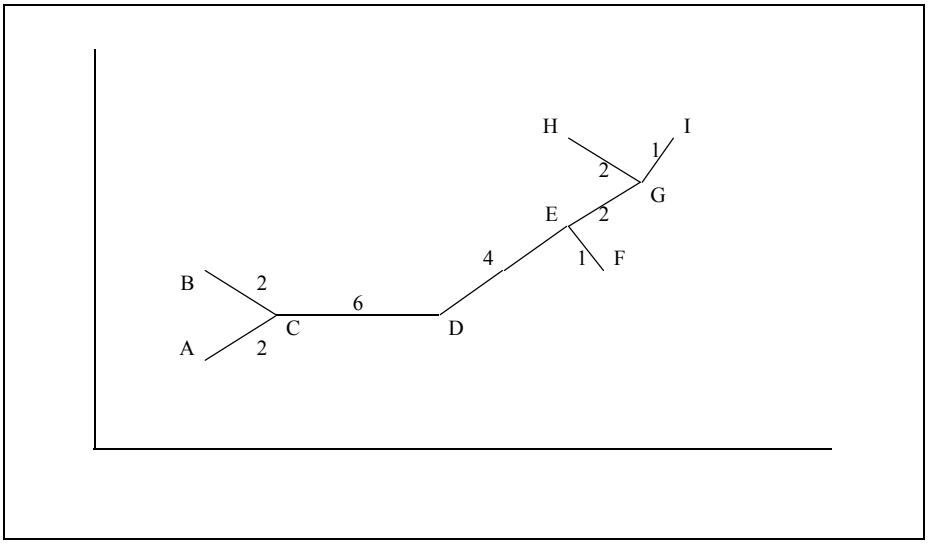

Figure 3.7 – Exemple d'arbre de recouvrement minimum

La construction d'un arbre de recouvrement minimum peut être aussi à la base d'une méthode hiérarchique descendante. A partir de cet arbre de recouvrement, la méthode consiste à couper les arcs ayant les plus grandes distances pour générer des classes. Par exemple, la Figure 3.7 montre un arbre de recouvrement minimum avec neuf documents en dimension deux. En coupant l'arc de plus grande distance (i.e. l'arc CD de longueur 6), on obtient deux classes composées des points {A, B, C} pour l'une et des points {D, E, F, G, H, I} pour l'autre.

3.7.7 Saut maximum

La méthode du saut maximum (ou « *complete Link* ») [Voorhees, 1986a] regroupe en une classe C_r deux classes C_i et C_j ayant une distance maximum entre elles. Elle correspond à la plus grande distance entre un document appartenant à la première classe et un document appartenant à la deuxième. La mise à jour des distances est aussi simple que la méthode précédente :

$$\forall C_k, \qquad d\left(C_r, C_k\right) = \max\left(d\left(C_i, C_k\right), d\left(C_j, C_k\right)\right) \tag{3.9}$$

Cette approche est moins sensible au bruit que la précédente. La complexité en temps de calcul est en $O(n^3)$ et celle en espace est de $O(n^2)$.

L'inconvénient de cette approche est son application sur des corpus de grande taille : coûteux en temps de calcul. De plus, cette méthode a tendance à créer des petites classes fortement liées.

La méthode du saut minimum et celle du saut maximum représentent deux extrêmes dans la mesure de similarité entre deux classes [Duda et Hart, 1973]. Ce genre d'approche est sensible aux bruits (éléments « perturbateurs » dans les documents). Un compromis pour atténuer ces inconvénients est l'utilisation de la moyenne comme distance entre classes : la moyenne arithmétique ou centroïde par exemple.

3.7.8 Saut moyen de groupe

La méthode du saut moyen de groupe (ou « *group average* ») regroupe les deux classes dont la moyenne de toutes les distances possibles entre un document de la première classe et un document de la deuxième est la plus petite.

3.7.9 Méthode de Ward

La méthode de Ward [Ward, 1963], également appelée la méthode de variance minimale, regroupe à chaque étape les deux classes qui minimisent l'inertie intraclasse (somme des distances des documents au centroïde).

Dans cette méthode, les classes sont représentées par un centre de gravité et possèdent un poids qui peut être par exemple le nombre d'éléments présents dans la classe.

Soient C_i et C_j deux classes, la distance de Ward est définie de la façon suivante :

$$d\left(C_i, C_j\right) = \frac{p\left(C_i\right) \cdot p\left(C_j\right)}{p\left(C_i\right) + p\left(C_j\right)} \cdot d^2\left(g\left(C_i\right), g\left(C_j\right)\right) \tag{3.10}$$

où $p\left(C_i\right)$ et $g\left(C_i\right)$ sont respectivement le poids et le centre de gravité de la classe C_i.

Cette méthode possède une propriété d'optimisation locale, c'est-à-dire qu'à chaque étape de l'algorithme, on minimise le critère d'inertie intraclasse. On peut remarquer que cette méthode utilise le même principe que la méthode k-means, c'est-à-dire minimise la somme des carrés des erreurs internes. C'est, en quel que sorte, la méthode hiérarchique analogue à la méthode de partitionnement k-means. Cette méthode génère des classes homogènes.

3.7.10 Méthode des centroïdes

Cette méthode représente chaque classe par un centroïde (*cf.* section 3.5.2.1). Les deux classes dont la distance entre centroïdes est minimale sont regroupées et un nouveau centroïde est calculé. L'opération est ensuite répétée. Cette méthode est la plus simple en ce qui concerne le calcul parmi toutes les méthodes hiérarchiques utilisant une « moyenne ».

3.7.11 Méthode du plus proche voisin

Cette méthode regroupe, à chaque étape, les deux plus proches voisins réciproques en suivant une chaîne de voisins réciproques. Le plus proche voisin d'un document E_i est le document E_j pour lequel la distance $d\left(E_i, E_j\right)$ est la plus petite. Les plus proches voisins réciproques sont par définition deux documents E_i et E_j pour lesquels E_i est le plus proche voisin de E_j et inversement.

3.7.12 Comparaison des méthodes

Des expérimentations ont été menées par [Griffiths et al., 1984] sur les méthodes du saut minimum, saut maximum, saut moyen de groupe et la méthode de Ward. Les résultats montrent que la méthode du saut minimum, bien qu'elle soit la plus facile à mettre en œuvre,

donne les plus mauvais résultats ; la méthode de Ward, quant à elle, donne les meilleurs résultats.

3.7.13 Caractéristiques des méthodes hiérarchiques

Dans les sections précédentes, nous avons décrit différentes approches pour la classification hiérarchique ascendante. Bien que ces approches semblent différentes, elles ont un ensemble de caractéristiques communes dues au fait qu'elles sont toutes fondées sur un algorithme générique.

La première caractéristique est le manque de critère d'optimisation global de ces méthodes. Ces dernières font appel chacune à des critères d'optimisation différents qui vont permettre de regrouper deux classes à un niveau local à chaque étape.

La seconde est le fait qu'à chaque étape, on regroupe définitivement les deux classes les plus similaires sans critère d'optimisation global. A chaque étape, on regroupe les deux classes les plus similaires selon un critère d'optimisation local et à partir d'une matrice de distances pré-calculées.

3.7.14 Limitations

Si nous résumons les sections précédentes, les méthodes de classification hiérarchique posent des problèmes sur des plans différents.

Sur le plan technique, ces méthodes nécessitent le calcul d'une matrice de distances (ou matrice de similarité) en pré-traitement, ce qui peut poser des problèmes de stockage mémoire pour de grands corpus. Les algorithmes utilisés sont lents, du fait de la complexité en temps de calcul, souvent en $O\left(p^2\right)$. Bien que la formule générique de mise à jour de Lance-Williams permette d'améliorer la complexité en temps de calcul, ces méthodes sont difficilement utilisables pour des corpus de grande taille.

Sur le plan conceptuel, ces méthodes utilisent des critères d'optimisation locaux qui n'induisent pas forcément une optimisation globale des résultats. De plus, les regroupements de classes sont définitifs, ce qui ne permet pas d'optimisation postérieure à la classification. Certains travaux ont malgré tout été menés pour pallier ce problème conceptuel : par exemple une méthode fondée sur des déplacements de branches de l'arbre pour améliorer la structure des classes [Fisher, 1996].

3.7.15 Avantages

Ces méthodes ont deux avantages majeurs, en regard des méthodes de partitionnement. Ces avantages sont les suivants :

- Il n'est pas nécessaire de détecter le nombre de classes finales puisque l'on fait un regroupement de classes deux par deux jusqu'à l'obtention d'une classe racine. Par extension, il n'y a pas de fonction d'initialisation, fonction qui est propre aux méthodes de partitionnement.
- Une construction suffit (équivalent à une itération ou une passe pour les algorithmes de partitionnement) pour atteindre le critère d'optimisation.

3.7.16 Chameleon

Chameleon [Karypis et al., 1999a] est un algorithme de classification hiérarchique agglomérative qui utilise une modélisation dynamique pour l'agrégation de classes. A travers cet aspect dynamique, le but est de pouvoir retrouver des classes avec des formes irrégulières et des tailles différentes (*cf*. Figure 6.1 du Chapitre 6).
Chameleon combine un algorithme de partitionnement de graphe G(V, E) et un nouvel algorithme hiérarchique ascendant. Cet algorithme « hybride » possède deux étapes pour retrouver des classes dans un ensemble de données.

- La première étape consiste à créer un ensemble de petites classes à partir des données initiales. Pour cette phase, une matrice de similarité est utilisée. Celle-ci regroupe toutes les similarités entre les données prises deux à deux. A partir de cette matrice, un graphe des k plus proches voisins sera, dans un premier temps, construit (la méthode du plus proche voisin est décrite dans la section 3.7.11). Dans ce graphe, il existera un lien entre une donnée u et une donnée v si v fait partie des k plus proches voisins de u et si u fait partie des k plus proches voisins de v. La valeur k est un paramètre de l'algorithme. Les auteurs stipulent que l'utilisation de ce type de graphe présente plusieurs avantages. La plupart des méthodes de classification hiérarchique utilisent toutes les données de la matrice de similarité et non juste une partie des données. Le premier avantage est que seules les données très *proches* entre elles (en terme de proximité) sont reliées. Le second avantage est en terme de temps de calcul, qui est ainsi diminué pour la suite de l'algorithme, car seule une partie des données initiales est utilisée.
  Une fois le graphe des k plus proches voisins créé, les auteurs appliquent dans un second temps un algorithme de partionnement de graphes appelé hMETIS [Karypis et Kumar, 1998] pour créer un ensemble de classes initiales.
- La seconde phase de Chameleon va combiner itérativement cet ensemble de petites classes initiales en utilisant un nouvel algorithme de classification hiérarchique. Cet algorithme repose sur deux mesures pour regrouper deux classes : l'interconnectivité relative (*Relative Interconnectivity* ou *RI*) et la proximité relative (*Relative Closeness* ou *RC*). Ainsi deux classes sont regroupées si la valeur de leur *RI* et de leur *RC* est élevée, c'est-à-dire si les deux classes sont très connectées entre elles et si elles sont très proches.

Interconnectivité relative

L'interconnectivité relative entre deux classes est la valeur absolue de l'interconnectivité entre ces deux classes normalisée par la moyenne arithmétique de l'interconnectivité de chaque classe. Ainsi l'interconnectivité entre deux classes C_i et C_j est définie de la façon suivante :

$$RI\left(C_i, C_j\right) = \frac{\left|EC\left(C_i, C_j\right)\right|}{\frac{1}{2}\left(EC\left(C_i\right) + EC\left(C_j\right)\right)} \tag{3.11}$$

où $EC\left(C_i, C_j\right)$ est la somme des arcs du graphe des *K* plus proches voisins regroupant les classes C_i et C_j, $EC\left(C_i\right)$ est la somme minimum des arcs pour diviser en deux la classe C_i, et $EC\left(C_j\right)$ est la somme minimum des arcs pour diviser en deux la classe C_j. *EC* est utilisé pour *Edge Cut*.

Proximité relative

Les concepts évoqués pour la proximité relative sont analogues à ceux définis pour l'interconnectivité relative. La proximité absolue entre deux classes C_i et C_j est la moyenne pondérée des arcs (alors que l'interconnectivité absolue est la somme des arcs) qui connectent un élément de C_i à un élément de C_j.

$$RC\left(C_i, C_j\right) = \frac{\overline{S}_{EC\left(C_i, C_j\right)}}{\frac{\left|C_i\right|}{\left|C_i\right| + \left|C_j\right|}\overline{S}_{EC\left(C_i\right)} + \frac{\left|C_j\right|}{\left|C_i\right| + \left|C_j\right|}\overline{S}_{EC\left(C_j\right)}} \tag{3.12}$$

où $\left|C_i\right|$ est le nombre d'éléments de la classe C_i.

Pour regrouper deux classes en utilisant ces deux mesures, plusieurs approches sont possibles :

- La première est d'utiliser deux seuils T_{RI} et T_{RC} respectivement pour RI et RC. Ainsi, pour une classe C_i, on calcule les deux mesures pour chaque classe C_j. Si, pour une classe C_j, la valeur de chaque mesure excède le seuil correspondant, alors les deux classes C_i et C_j sont regroupées. Si cette condition est remplie par plusieurs classes C_j, alors la classe C_i sera regroupée avec la classe C_j qui lui est la plus interconnectée, c'est-à-dire celle qui aura la plus grande valeur de $EC\left(C_i, C_j\right)$.
- La seconde est de maximiser une fonction combinant les deux mesures :

$$RI\left(C_i, C_j\right) \cdot RC\left(C_i, C_j\right)^{\alpha} \tag{3.13}$$

Cette fonction reflète le but des auteurs consistant à regrouper deux classes en fonction des deux mesures. La valeur de α va accroître l'importance d'une mesure par rapport à l'autre. Si $\alpha = 1$, alors les deux mesures ont la même importance. Par contre, si les auteurs veulent donner plus d'importance à la proximité relative alors $\alpha > 1$. Pour donner plus d'importance à l'interconnectivité relative, alors $\alpha < 1$.

La complexité en temps de calcul est de $O\left(Nm + N\log\left(N\right) + m^2\log\left(m\right)\right)$ où m est le nombre de classes initiales.

L'avantage de cette méthode est de retrouver des classes de formes irrégulières, de taille et de densité différentes.

L'inconvénient est que l'algorithme est peu performant en temps de calcul pour des données de grande dimension.

Les étapes de Chameleon sont résumées dans Algorithme 3.6.

1. Création d'un graphe des *k* plus proches voisins.
2. Partitionner le graphe résultant de l'étape précédente.
3. Appliquer un algorithme hiérarchique agglomératif sur l'ensemble des classes.

Algorithme 3.6 – Chameleon

Dans la section suivante, nous nous attachons à des méthodes dont l'application est spécifique au Web.

3.8 Classification basée sur les liens hypertextes

Les méthodes de classification abordées dans les sections précédentes (classification basée sur les centres) peuvent s'appliquer, *a priori*, sur n'importe quel type de données, avec une distance ou une mesure de similarité donnée pour certains algorithmes. Ces données sont représentées par des tableaux de distances, des tableaux de contingences, etc. Avec l'émergence d'Internet dans le domaine public, ces caractéristiques et notamment les liens hypertextes font que de nouvelles méthodes de classification documentaires ont été développées. Ces méthodes ne sont plus uniquement fondées sur le contenu du texte (mots, phrases, titre, etc.) mais également sur les liens hypertextes que contiennent les textes. L'hypothèse principale pour justifier l'utilisation de ces liens hypertextes repose sur l'aspect suivant : un lien hypertexte indique un lien sémantique entre les documents [Kleinberg, 1999]. La notion de sémantique peut s'expliquer par le fait suivant : le propriétaire ou le concepteur d'une page Web insère dans celle-ci des liens hypertextes pointant vers d'autres pages qui sont *censées* être thématiquement proches de celle de sa page.

C'est-à-dire que l'on suppose aisément que le concepteur est à même de savoir si les documents pointés sont bien « liés sémantiquement » à sa page.

Les algorithmes de classification fondés sur les liens hypertextes modélisent les données (ensemble de documents Web) à l'aide d'un graphe $G(V,A)$; ce graphe est orienté dans la plupart des travaux. Dans un graphe $G(V,A)$, V représente l'ensemble des nœuds, c'est-à-dire une page Web dans notre cas, et A représente un arc (pour un graphe orienté), c'est-à-dire un lien hypertexte.

3.8.1 Algorithme d'agrégation

Les méthodes de classification fondées sur les liens hypertextuels ont été développées dans un premier temps sur la base d'algorithmes des graphes. La « méthode par interrelations » ou agrégation développée par [Botafogo et Schneiderman, 1991] utilise un algorithme des graphes pour regrouper les documents qui sont fortement connectés entre eux. Le but est de retrouver des ensembles de documents qui sont sémantiquement fortement reliés Cette méthode est itérative et « coupe » un ensemble de liens à chaque étape de l'algorithme. Elle utilise une « mesure de compacité » comprise entre 0 et 1 pour un graphe ou un sous-graphe (classe) qui permet de savoir si ce graphe est « sémantiquement bien connecté » ou non. Cette mesure représente une distance moyenne de liens entre les différents nœuds. Bien que cette méthode regroupe les documents fortement connectés, elle n'est cependant pas assez discriminante. Elle est améliorée dans [Botafago, 1993].

3.8.2 Algorithme de co-citations

L'algorithme de co-citations est une autre approche de classification à partir de liens hypertextes. L'analyse sur les co-citations d'auteurs (ACA) a pour but de regrouper les auteurs par spécialité. Elle permet également de déceler les relations qui peuvent exister entre différents auteurs d'un même domaine d'activité, et ce à travers leurs publications scientifiques. [White et McCain, 1998] ont utilisé l'analyse de co-citations d'auteurs[1] pour regrouper les champs des sciences de l'information.

L'ACA peut se coupler à différentes méthodes d'analyse de données, par exemple la méthode de mise à l'échelle multidimensionnelle MDS (multidimensional scaling). Un algorithme utilisant la méthode MDS a été proposé par [Larson 1996] pour classer un ensemble de documents sur les sciences de la terre. Cet algorithme commence par la construction d'une matrice $M = \left(\{M_{ij}\}\right)$ de co-citations où M_{ij} représente le nombre de documents qui sont liés à la fois au document *i* et au document *j*. Après un calcul sur cette

[1] Les données utilisées sont principalement l'index SCI (*Science Citation Index*) et l'index SSCI (*Social Science Citation Index*) provenant tous les deux de l'ISI (*Institute for Scientific Information*)

matrice pour obtenir des mesures de proximité entre documents, la méthode MDS est appliquée pour visualiser les classes de documents regroupés par thèmes.

3.8.3 Algorithme « *trawling* »

L'algorithme de « *trawling* » décrit par [Kumar et al., 1999] combine à la fois l'analyse de co-citations et l'analyse de graphes. Le but de cet algorithme est d'identifier un ensemble de communautés sur le Web. Une communauté peut être définie comme un ensemble de personnes qui ont un centre d'intérêt en commun, comme les voitures Porsche de type Boxter par exemple.

3.9 Classification hybride

Les deux sections précédentes présentent des méthodes de classification de documents basées sur l'analyse textuelle pour l'une et fondées sur l'analyse de liens hypertextes pour l'autre. Ces deux approches ont évidemment leurs propres avantages et inconvénients. L'idée principale de la classification hybride est de combiner les deux types de classification énoncés précédemment en utilisant le « meilleur » ou les avantages des deux approches. L'approche hybride suppose ainsi que les lacunes d'un type de méthode de classification (textuel ou topologique) sont compensables ou fortement atténuées par une méthode de l'autre type de classification. Pour montrer que cette hypothèse n'est pas dénuée de sens, on peut rapidement récapituler les caractéristiques des deux approches. Dans la classification de type textuel, une distance ou une mesure de similarité est utilisée pour déterminer la proximité entre les documents. Dans une classification de type topologique, les citations, i.e. les liens hypertextes, sont utilisées pour déterminer une relation puis une proximité entre les documents. Bien que la classification de type topologique se fonde sur l'hypothèse que le lien hypertexte donne une valeur du lien sémantique entre deux documents (du fait que ce soit le concepteur d'une page qui insère le lien), les liens créés ne donnent aucune information quant au « degré de similarité » entre les documents. Cette information peut être utile pour le choix des liens à éliminer dans le cas où la similarité est trop faible entre les documents (ou à conserver dans le cas d'une similarité élevée). Cette information peut être apportée grâce aux caractéristiques des méthodes de classification de type textuel. De même, la classification de type textuel peut donner des meilleurs résultats en ajoutant des informations aux documents.

Les premières méthodes hybrides mêlant liens hypertextes et contenu textuel sont apparues en 1996 avec les travaux de Pirolli et indépendamment ceux de Weiss. Les travaux de Pirolli [Pirolli et al., 1996] sont fondés sur l'utilisation d'une combinaison de données de nature différente pour la représentation de chaque document. Chaque document est représenté par un vecteur unique contenant plusieurs informations : des liens topologiques, des informations méta de la page (les informations méta pour une page Web sont par exemple le titre, la taille du fichier ou l'URL.), des similarités textuelles et d'autres données. Les

similarités textuelles sont obtenues par une technique classique de la recherche d'informations. Chaque document est représenté par un vecteur de fréquences de mots. Le produit scalaire est ensuite appliqué entre les vecteurs des documents. Ces informations ne sont pas fusionnées mais utilisées indépendamment à travers différents graphes. Les classes sont obtenues après un algorithme d'activation de nœuds appliqué sur les différents graphes. L'algorithme d'activation de nœuds commence par initialiser un nœud en lui donnant un poids puis active les autres en parcourant le graphe et modifie le poids des arcs.

3.9.1 HyPursuit

Les travaux de Weiss [Weiss et al., 1996] sont fondés sur une classification hiérarchique de type saut maximum avec une mesure de similarité « hybride » entre les classes. L'algorithme de classification « *content-link* » utilise les termes des documents et leurs structures topologiques. La classification du type lien complet (décrite dans la section 3.7.7) est une classification hiérarchique ascendante qui regroupe à chaque étape les deux classes les plus proches suivant une mesure de similarité définie. Le choix de l'algorithme de classification (les différents algorithmes hiérarchiques sont décrits dans la section 3.7) est lié à la simplicité de la mise en œuvre : « *Although faster clustering algorithms exist, we chose the complete link method because it was easy to implement* ». Dans les travaux de Weiss, la mesure de similarité $S_{ij}^{hybride}$ n'est pas la mesure usuelle mais une combinaison $F\left(S_{ij}^{termes}, S_{ij}^{liens}\right)$ d'une mesure de similarité « textuelle » classique S_{ij}^{termes} et d'une mesure de similarité topologique S_{ij}^{liens}. La fonction F utilisée dans HyPursuit est la fonction `max`. La mesure de similarité topologique S_{ij}^{liens} est elle-même une combinaison linéaire de trois paramètres : le lien direct entre le document *i* et le document *j*, les ancêtres communs, les descendants communs. La valeur du lien direct entre deux documents *i* et *j* varie inversement avec la longueur du plus court chemin entre ces deux documents. Ce lien direct part de l'hypothèse que deux documents ont une relation sémantique entre eux (créée par les liens topologiques), et que cette relation est transitive. Ainsi la valeur de la relation sémantique entre deux documents *i* et *j* diminue lorsque le nombre de liens hypertextes nécessaires pour aller du document *i* au document *j* augmente. La valeur des ancêtres communs est proportionnelle au nombre de liens entrants que *i* et *j* ont en commun. De même, la valeur des descendants communs est proportionnelle au nombre de liens sortants. La mesure de similarité textuelle utilisée est le produit scalaire entre les deux vecteurs de mots. La fonction de pondération des mots utilisée est similaire à celle de [Salton et Buckley, 1988]. Les expérimentations[1] menées sur un ensemble de documents de 195 pages Web provenant du site de CNN[2] montrent que l'algorithme « *content-link* » génère des classes qui se rapprochent le plus du classement de CNN. L'algorithme décrit ci-dessus est un des aspects de HyPursuit, qui est un moteur de recherche sur réseau hiérarchique (*hierachical network*

[1] Les expérimentations sont effectuées avec trois méthodes : lien complet classique, lien complet avec liens hypertextes seul et content-link.
[2] www.cnn.com

search engine). Ce moteur de recherche inclut différentes fonctionnalités pour aider l'utilisateur dans sa recherche d'informations. Une de ces fonctionnalités est par exemple la génération automatique de résumés de classes basée sur les pondérations de mots.

L'algorithme est fondé sur une méthode de classification hiérarchique du type lien complet rendant la complexité en temps de calcul quadratique pour l'algorithme.

3.9.2 L'algorithme Toric k-means

Modha et Spangler ont proposé un nouvel algorithme de classification nommé « Toric k-means » [Modha et Spangler, 2000], qui est une extension de l'algorithme de k-means. Pour cet algorithme, les auteurs définissent une nouvelle mesure de similarité entre documents fondée sur les trois données usuelles pour les méthodes de classification hybride : les mots, les liens entrants et les liens sortants. Cette mesure de similarité est une combinaison linéaire de ces trois types de données où chaque donnée a une valeur quantitative. Dans l'algorithme content-link de Weiss [Weiss et al., 1996] décrit ci-dessus, la similarité entre deux documents est la valeur maximale entre la similarité textuelle et celle topologique. En d'autres termes la similarité est représentative d'un seul type de données. Néanmoins, les deux similarités peuvent avoir des valeurs élevées toutes les deux. Modha et Spangler représentent chaque document par un triplet de vecteurs (*D*,*F*,*B*) où *D* représente un vecteur de fréquences de mots, *F* représente un vecteur de liens entrants et *B* représente un vecteurs de liens sortants.

Pour chaque vecteur, une liste d'éléments est initialement créée puis filtrée. Par exemple, pour le vecteur *D*, une liste de tous les mots apparaissant dans tous les documents est créée. De cette liste, les mots présents dans moins de deux documents sont éliminés ainsi que des mots vides et les balises `html`. La mesure hybride de similarité $S(d_1, d_2)$ –équation (3.14)– entre un document $d_1 = (D_1, F_1, B_1)$ et un document $d_2 = (D_2, F_2, B_2)$ peut être exprimée comme la somme pondérée des produits des vecteurs correspondants :

$$S(d_1, d_2) = a_1 D_1^T D_2 + a_2 F_1^T F_2 + a_3 B_1^T B_2 \qquad (3.14)$$

avec $a_1 + a_3 + a_3 = 1$.

Modha et Spangler définissent également un vecteur (D_j^*, B_j^*, F_j^*) représentatif d'une classe C_j. Ce vecteur représentatif appelé « *concept triplet* » est également un triplet de vecteurs –équation (3.15)– dont chacun est calculé comme la somme des vecteurs normalisés correspondants de C_j :

$$D_j^* = \frac{\sum_{d_i \in C_j} D_i}{\left\| \sum_{d_i \in C_j} D_i \right\|},\ F_j^* = \frac{\sum_{d_i \in C_j} F_i}{\left\| \sum_{d_i \in C_j} F_i \right\|},\ B_j^* = \frac{\sum_{d_i \in C_j} B_i}{\left\| \sum_{d_i \in C_j} B_i \right\|}. \quad (3.15)$$

Un vecteur représentatif d'une classe C_j est le triplet de vecteurs le plus proche de tout triplet de la classe C_j, comme le centroïde pour la méthode k-means classique. Cette notion de concept est à la base de deux autres notions connues dans les méthodes de classification à savoir :

la cohérence des classes : $\sum_{d_i \in C_j} S\left(d_i, C_j\right)$

et la fonction d'objectivité : $\sum_j \sum_{d_i \in C_j} S\left(d_i, C_j\right)$.

L'algorithme Toric k-means utilise un moteur de recherche classique[1] pour récupérer un ensemble initial de pages Web en réponse à une requête. Cet ensemble initial sera ensuite enrichi de pages Web provenant d'une partie (sélection puis filtrage) des liens entrants et des liens sortants pour cet ensemble initial. L'algorithme débute avec une partition arbitraire de documents. Cette partition est ensuite itérativement modifiée en réallouant chaque document au vecteur représentatif le plus « proche » et en calculant ces nouveaux vecteurs représentatifs. A noter que le nombre de classes *K* est un paramètre donné de l'algorithme. L'algorithme s'arrête sous certains critères : par exemple, si la différence des valeurs de la fonction d'objectivité entre deux itérations est en dessous d'un seuil, alors l'algorithme s'arrête.

Enfin, les auteurs proposent une façon de représenter chaque classe en utilisant six critères : trois critères descriptifs et trois critères discriminants. Les critères descriptifs (représentatifs des trois vecteurs *D*, *B*, *F*) sont : le résumé, le « *breakthrough* » et le « *review* » qui sont, pour une classe donnée, les trois documents qui se rapprochent le plus du vecteur représentatif de la classe. La notion de proximité est vue à travers la fonction de similarité cosinus. Pour le critère résumé par exemple, le document choisi pour une classe C_j sera celui dont le vecteur *D* sera le plus similaire au vecteur D_j^*. Les trois critères discriminants pour une classe donnée sont les mots-clefs, les citations et les références. Les mots-clefs représentent les mots pour lesquels les poids sont plus importants dans une classe donnée que dans n'importe quelle autre classe (ces mots peuvent faciliter l'interprétation thématique du document). De même, les citations et les références représentent respectivement les liens entrants et sortants les plus courants pour une classe donnée.

[1] Le moteur de recherche utilisé dans les expérimentations par Modha et Spangler est Altavista (www.altavista.com)

L'algorithme Toric k-means est une variante de l'algorithme k-means classique où la représentation et la nature des données diffèrent ; la mesure de similarité et la fonction d'objectivité sont également différentes. En revanche, la complexité en temps de calcul reste linéaire avec le nombre de documents.

3.10 Conclusion

Ce chapitre nous a permis d'évoquer le principe général et les différentes approches de la classification non-supervisée. Bien que ces différentes approches soient fondées sur un schéma identique, c'est-à-dire un processus de classification se déroulant en trois étapes, elles différent entre elles notamment par l'algorithme, mais aussi par la représentation des classes, les hypothèses et le contexte d'application.

En résumant ces approches, on constate que les deux grandes catégories sont, dans la littérature, les méthodes de partitionnement et les méthodes de classification hiérarchique. Les premières sont principalement représentées par la méthode k-means ou X-means et les secondes par la méthode de Ward. Les méthodes de partitionnement souffrent principalement de la difficulté à déterminer la valeur de K, le nombre de classes, mais aussi à déterminer une partition initiale adéquate. En effet, la partition finale de ces méthodes est dépendante de la partition initiale. Les méthodes de classification hiérarchiques souffrent du passage à l'échelle, du fait de leur complexité en temps de calcul.

Chapitre 4

Méthodologie

Résumé

Un corpus peut être vu comme un ensemble de textes homogènes, c'est-à-dire que les textes partagent des caractéristiques communes, permettant l'application d'outils et des techniques d'extraction de connaissances dans le but d'acquérir des informations. Un corpus est également un support pour évaluer et comparer des méthodes ou des systèmes de recherche d'information.

Des corpus composés de thèmes variés (il s'agit généralement de textes de sources journalistiques et donc traitant de sujets divers) sont disponibles pour évaluer ces méthodes et systèmes. Notre approche d'aide à la recherche d'information sur le domaine juridique impose un choix de corpus adéquat dans ce domaine précis. Bien qu'aucun corpus ne soit défini et établi dans le domaine du droit, à des fins d'évaluation, des corpus existent et conduisent à certaines considérations quant au choix de l'un d'eux.

Le futur corpus n'étant pas reconnu en tant que tel dans la communauté de la RI, il est alors utile de définir la notion de corpus de référence (définition des caractéristiques nécessaires) et de motiver notre choix à travers des considérations à la fois juridiques (nécessitant l'avis d'experts du domaine) et pratiques (le corpus doit permettre l'évaluation de la méthode, si possible sans l'avis d'experts).

4.1 Introduction

Afin de pouvoir comparer plusieurs systèmes de recherche ou plusieurs méthodes, des corpus d'évaluation sont, depuis quelques années, constitués et mis à disposition dans ce but. Le corpus le plus répandu pour l'évaluation des systèmes de recherche documentaire est celui constitué dans le projet TREC (*text retrieval conference*). Ce projet a pour objectif d'organiser des campagnes internationales d'évaluation dont le premier congrès s'est déroulé en 1992. Des campagnes francophones d'évaluation sont également disponibles dans ce cadre ; c'est le cas des campagnes Amaryllis qui ont débuté en 1996. Ces corpus et campagnes d'évaluation sont évoqués plus en détail dans [Harman and Smeaton, 1997], [Bellot, 2000].

Dans notre démarche d'aide à la recherche d'information dans le domaine juridique, ces corpus sont dans un premier temps peu intéressants dû à la nature même des domaines abordés par les textes les constituant. Toutefois, notre méthodologie, décrite dans la section 4.4.2, est fondée en partie sur une méthode de classification non-supervisée qui peut, dans un second temps, faire l'objet d'une évaluation sur un corpus composé de thèmes hétérogènes tel que TREC, cette évaluation permettant ainsi de déterminer la précision des résultats de notre méthode de classification indépendamment d'un domaine spécifique.

La sélection objective d'un corpus d'évaluation dans un domaine spécifique tel le monde juridique ne peut se faire sans quelques connaissances spécifiques de ce domaine. Ainsi, le choix d'un tel corpus repose sur les travaux menés, précisément sur ce domaine, par Lame [Lame, 2002]. Ces travaux ont consisté à élaborer une ontologie du droit consacrée à la recherche d'information. Un axe majeur des travaux de l'auteur est l'obtention d'une liste de termes juridiques.

Ainsi, dans ce chapitre, nous nous attachons au choix du corpus juridique. Ce corpus doit être, pour la finalité de nos travaux, représentatif du domaine. Toutefois, il doit également satisfaire à nos besoins en ce qui concerne l'évaluation de notre méthode. En effet, un aspect de notre méthode, outre la construction de classes, est l'*étiquetage thématique* de ces classes.

Dans un premier temps, nous développons dans ce chapitre la définition d'un corpus de référence. Dans un second temps, nous déterminons notre corpus de référence qui servira, par la suite, de base à notre démarche de classification de documents. Enfin, nous présentons globalement notre méthode à travers des hypothèses et des choix algorithmiques.

4.2 Définition du terme : corpus de référence

Le corpus de référence est un ensemble de textes rassemblés dans un objectif précis, celui d'être la base d'une démarche d'acquisition de connaissances. Cette acquisition peut se

traduire dans certains cas par une identification de termes : nous donnons en exemple le projet CoRRecT[1].

Sur ce corpus seront utilisés des outils et techniques de traitement automatique de la langue avec, pour objectif, l'obtention des éléments importants de notre méthode de classification non-supervisée (décrite dans le Chapitre 6) : les termes du domaine, à travers une méthode d'étiquetage des classes. Cela s'apparente à un système d'identification de termes.

La notion de corpus de référence peut s'avérer confuse sans fixation de buts : on parle également de *corpus dédié* ou *special purpose corpus* [Moreno, 2001]. Certains distinguent le corpus de référence du corpus dédié [Pearson, 1998] inférant que la caractéristique principale du premier est la représentativité, tandis que le corpus dédié est un corpus dont la composition est déterminée par un objet précis pour lequel il est élaboré[2].
Un corpus de référence, selon Sinclair [1996], est « conçu pour fournir une information en profondeur sur une langue. Il vise à être suffisamment grand pour représenter toutes les variétés pertinentes de cette langue et son vocabulaire caractéristique, de manière à pouvoir servir de base à des grammaires, des dictionnaires et d'autres usuels fiables. ».
Dans notre expérience, le corpus dédié et le corpus de référence se confondent dans un seul et même corpus ; nous parlons alors de corpus de référence.

Dans [Lame, 2002], trois types d'élaboration de corpus de référence sont évoqués. La caractéristique principale, requise par l'auteur pour le corpus de référence, est la plus grande couverture possible du domaine, essence même du corpus de référence.

Une des caractéristiques voulues pour notre méthode de classification est la possibilité de classer un grand nombre de documents, et si nécessaire, sur un grand nombre de classes. Ainsi, un corpus de référence, recouvrant donc très largement le domaine, va permettre d'expérimenter notre méthode sur un grand nombre de documents dans le but d'identifier les termes du domaine.

Les trois types d'élaboration de corpus de référence sont présentés ci-dessous :

1) Soit ce corpus est spécialement élaboré à cet effet et les textes qui le composent n'existaient pas auparavant. Ce cas regroupe les démarches d'acquisition de connaissances à partir de textes qui consistent à récolter préalablement des avis d'experts sur les questions diverses. Les spécifications sur les connaissances d'un domaine sont alors réunies dans des documents qui, rassemblés, constituent le corpus de référence. Ainsi, une méthode de gestion des connaissances impliquant une phase d'acquisition prévoit un tel recueil des savoirs des experts dans des documents textuels par des procédures d'interviews. Les savoirs des experts sont en effet souvent opposés aux connaissances explicites exprimées en tant que telles et stockées sur un support

[1] http://www.sciences.univ-nantes.fr/info/perso/permanents/enguehard/recherche/CoRRecT/CoRRecT.htm
[2] *A corpus whose composition is determined by the precise purpose for which it is to be used*.

physique. Les connaissances des experts sont, elles, dites tacites, dans le sens où elles sont difficilement exprimables. Ces connaissances des experts, et surtout les connaissances organisationnelles, sont ainsi difficilement communicables donc accessibles. L'enjeu des systèmes de gestion des connaissances organisationnelles est alors de faire passer ces connaissances du tacite vers l'explicite. La constitution de corpus de référence incluant des interviews d'experts est fonction du domaine en question, d'une organisation en particulier, et de l'application visée.

2) Un deuxième cas consiste à élaborer ce corpus de référence en rassemblant des documents épars préexistants. Ainsi, des travaux s'attachent à la constitution de corpus de référence en déterminant des mots clés du domaine, en cherchant des documents correspondants sur le Web et en sélectionnant parmi eux les documents les plus représentatifs [Grabar, 2001]. Un corpus de référence est alors élaboré, réunissant un ensemble de documents représentatifs du domaine. En effet, l'enjeu est de sélectionner des documents représentatifs du domaine, ce qui peut se faire soit en lisant ces documents, soit en décidant de ne garder que ceux présentant un nombre de termes du domaine suffisamment élevé. L'utilisation de tels documents préexistants, comparée aux interviews, pour acquérir des connaissances à partir de textes, comporte l'avantage indéniable de la rapidité et de la facilité. En effet, des interviews d'experts ne sont ainsi pas nécessaires. Pour adopter cette méthode, il faut bien sûr que le domaine s'y prête et que les documents existent. Notre cas du domaine juridique français est typique de ce point de vue. Le droit est ainsi un domaine présidé par un ensemble de normes, ces normes étant l'objet même du domaine. Le droit français n'étant par ailleurs pas un droit coutumier, les normes qui le composent, les connaissances qu'il exprime, le sont sous la forme de textes. Ainsi, l'un des fondements du droit français est le Code civil, document créé sous Napoléon. Les sources des normes du droit français sont donc bien textuelles ; l'essence même du droit français transparaît dans des textes.

3) Enfin, dernier cas, le nôtre, la définition du corpus de référence consiste à identifier un corpus préexistant et apte à servir de corpus de référence. La totalité du droit français faisant l'objet de textes, à quelques exceptions près de quelques pratiques coutumières, notre corpus de référence est constitué de ces textes. Nous ne pouvions cependant envisager de considérer tous les documents du droit français, de tous les inclure dans le corpus de référence. Les sources du droit français, bien que toutes textuelles, sont de divers types : les textes constitutionnels, les lois organiques, les lois ordinaires, les normes réglementaires, la jurisprudence etc. A cet ensemble déjà vaste s'ajoutent les analyses d'experts qui constituent la doctrine. Cette doctrine est difficilement quantifiable, même à considérer toutes les revues autorisées traitant du droit français. Ainsi, les sources textuelles du droit français sont-elles quasiment infinies donc indéfinissables.

Au regard des documents dont nous disposions et des expériences effectuées par Lame [2002] que nous explicitons et commentons ci-dessous, nous en sommes arrivés à envisager l'ensemble des codes du droit français comme notre corpus de référence. Cet ensemble constitue bien un corpus de référence du fait qu'il rassemble tous les codes qui sont eux-mêmes une des sources du domaine, celle de droit codifié.

4.3 Choix du corpus de référence

Nous disposons au Centre de recherche en informatique de l'Ecole des mines de Paris d'un ensemble de documents juridiques du droit français ainsi que d'un ensemble de documents de droit communautaire. Ces documents sont ceux qui se trouvent diffusés par les sites officiels tels Légifrance[1] ou Europa[2]. Nous disposons ainsi de documents issus du Journal Officiel de la République française édition lois et décrets (ci-après JO), des codes du droit français et des directives et règlements européens.

4.3.1 Le Journal Officiel de la République française

Le corpus du JO dont nous disposons est celui diffusé par le site officiel Légifrance. Le Journal Officiel de la République française édition lois et décrets rassemble un ensemble de documents propres au droit français : les lois, les décrets, les arrêtés émanant des différents ministères du gouvernement et des avis d'autorités telles l'Autorité de Régulation des Télécoms ou le Conseil Supérieur de l'Audiovisuel. Nous disposons de l'ensemble de ces documents depuis le 1er janvier 1998 et des textes relativement importants pour les dates antérieures. Ce corpus contenait, en mai 2003, cent vingt mille documents, chaque loi, décret, arrêté ou avis constituant un seul document.

Ce corpus représente un ensemble d'environ 600 000 mots différents et trois à quatre millions de groupes de mots différents. Ces chiffres élevés sont expliqués, ou du moins explicables, par le fait que de nombreux noms propres apparaissent dans ces documents, des noms de personnes dans le cas des décrets de nomination par exemple, ou des noms de localités. Pour donner un ordre d'idées, l'ensemble des documents du JO correspondant à l'année 2000, soit un total de 24 178 documents, comporte 139 410 mots et 447 324 groupes de mots différents[3]. Ce sous-corpus du JO représente 172 Mo de données (dont 40 Mo sont octroyés au balisage html).

Dans les travaux récents de Lame [Lame, 2000], l'auteur a tenté d'exploiter les documents du Journal Officiel : « La démarche ne s'est cependant pas révélée fructueuse » [Lame, 2001a], [Lame, 2001b]. « Il s'avère en effet que le corpus du Journal Officiel se révèle

[1] http://www.legifrance.gouv.fr/jo
[2] http://www.europa.eu.int
[3] données fournies par le moteur Pertimm

inadapté à une démarche d'acquisition de connaissances à partir de textes, ou à tout le moins, il est moins adapté à cette démarche que le corpus des codes. »

Dans [Lame, 2002], l'auteur ne pouvait pas envisager de traiter l'ensemble des quelques cent mille documents du JO (nombre de documents correspondant à l'année 2002). Ses tentatives ont donc été menées sur un échantillon de ce corpus à savoir les lois publiées au JO durant l'année 2000, ainsi que celles publiées en 2001 (jusqu'en février uniquement) dont l'ensemble représente 8700 documents. Dans notre cas, cet argument ne peut être pris en considération dans la justification du choix du corpus. En effet, l'un de nos objectifs concerne la méthode de classification : pouvoir classer un grand nombre de documents.

Les travaux du même auteur ont également montré que, sur cet échantillon de corpus, un ensemble de plus de 500 000 termes différents a été détecté avec, parmi ceux-ci, de nombreux noms propres, et un constat : celui que les termes propres du droit se trouvaient "noyés" dans un ensemble de termes de la langue naturelle ; la densité terminologique juridique est trop faible.

Le corpus de référence doit en effet être suffisamment homogène pour constituer un ensemble cohérent, ce que ne constituent pas, finalement, les lois du Journal Officiel dont nous disposons. De préférence, ce corpus doit également avoir la particularité de contenir une majorité de termes du domaine.

L'application de notre méthode de classification, sur ce corpus de JO, pouvait être envisagée techniquement. Cependant, cette solution n'a pas été retenue pour deux raisons. La première est liée au problème d'évaluation et de validation des classes. Le corpus du JO n'étant pas initialement regroupé thématiquement, il est difficile de valider ou de calculer automatiquement la précision des résultats de la classification. La phase de validation des classes nécessite, dans ce cas, l'intervention d'un expert du domaine. La seconde raison concerne la validation, cette fois-ci, de l'étiquetage thématique des classes, qui nécessite également l'intervention d'un expert. Cet étiquetage est d'autant plus difficile à évaluer que certains textes regroupent plusieurs thématiques au sein d'un même document.

Ce corpus n'a pas été retenu pour les expérimentations et évaluations de notre méthode pour ces deux aspects principaux. Le premier est lié à la nature même des textes évoquée par Lame (faible densité terminologique juridique), et le second lié à la structure thématique inexistante du corpus qui nécessite l'intervention d'un expert pour l'évaluation.

Cependant, ce corpus offrait l'avantage (par rapport aux caractéristiques de notre méthode) de regrouper un bien plus grand nombre de documents que celui des codes.

4.3.2 Les codes du droit français

Nous utilisons comme base de l'application de notre méthode de classification les codes du droit français tels que présents sur le site officiel Légifrance au courant de l'année 2001 (mai 2001). En réalisant quelques regroupements, par exemple rassembler le Code rural

ancien et le Code rural ou bien le Code général des impôts avec ses quatre annexes, nous obtenons un ensemble de cinquante-sept codes. Ce nombre ne prend pas en compte le fait que certains codes comportent des parties législatives, décrets ou arrêtés. La liste des codes est consultable en Annexe A.

Nous notons que chacun de ces codes a une importance juridique relative au thème qu'il traite. Il va ainsi de soi que le Code civil ou le Code pénal ont une importance majeure au regard de codes beaucoup plus particuliers tels le Code de déontologie des sages-femmes ou le Code de la médaille militaire et de la légion d'honneur.

Les codes ont, par ailleurs, cette particularité d'avoir un vocabulaire relativement ramassé sur le domaine juridique. Ainsi, la fonction première des codes est de rassembler thématiquement les normes. Ce sont justement ces thématiques que nous tentons de reconnaître à travers notre méthode. Les codes sont actuellement créés par la Commission supérieure de codification dont l'objet est de rassembler thématiquement les normes, de les rationaliser et d'organiser un code avec ces normes. Toutes les normes n'ont pas encore fait l'objet d'une codification. Les codes ne couvrent donc pas l'intégralité des sources normatives du droit français. Ils en couvrent cependant une bonne partie et, à tout le moins, les éléments les plus importants comme le code civil. Recueils de normes, les codes utilisent donc les termes fondamentaux du droit normatif. L'expérience a également montré que, globalement, le vocabulaire utilisé dans les codes est relativement concentré autour du droit. Ce fait est expliqué par l'objet même des codes : rassembler les normes de façon rationnelle. Ainsi, nous ne disposons dans les codes ni d'indications sur les mesures d'applications de la norme en question, ni des noms des ministres chargés de son application comme c'est le cas dans les lois publiées au JO. Le corpus des codes s'avère donc, pour toutes ces raisons, être un corpus adapté pour constituer en tant que tel un corpus de référence. De plus, le corpus des codes a l'avantage de constituer par lui-même un corpus divisé thématiquement, ce qui est exploité dans notre classification.

Pratiquement, nous disposons de ces codes sous la forme d'un document html par article de codes tels que spécifiés dans les codes eux-mêmes, par exemple, les articles L.123-7 ou R.613-46 du Code de la propriété intellectuelle. Les 57 codes rassemblent un total de 64 184 articles. L'ensemble de ces fichiers des articles des codes correspond à 194 Mo de données. La totalité des codes rassemble un total de 6 403 216 mots.

4.4 Problématique

Dans le Chapitre 2, nous avons évoqué les principes de la recherche d’information ainsi que les inconvénients liés aux systèmes de RI. En synthétisant, les systèmes de recherche d’information donnent une liste de documents en réponse à une requête. Cette liste est généralement longue et nécessite ainsi, de la part de l’utilisateur, un effort pour déterminer les documents pertinents. Cette tâche est d’autant plus difficile si les documents pertinents sont

mal positionnés dans la liste, c'est-à-dire s'ils ne se trouvent pas dans la première page de résultats. Dans le but de trouver les documents voulus dans les premières positions, l'utilisateur entre alors dans un processus itératif de recherche (*cf.* section 1.9.1). Ce processus consiste à déterminer, à partir des documents retournés en réponse, les mots et/ou les expressions à ajouter/enlever/exclure de la requête pour atteindre les documents voulus.

La finalité de nos travaux est de mettre à disposition de l'utilisateur les moyens qui lui permettront d'accéder facilement et rapidement à l'information voulue. L'objectif est de faire intervenir l'utilisateur dans un processus de recherche semi-automatique qui nécessite peu d'efforts supplémentaires par rapport à un système traditionnel. Les deux notions évoquées précédemment, facilité et rapidité, semblent superflues au regard des systèmes de recherche d'information tels que les moteurs de recherche sur le Web. Cependant, celles-ci, et en particulier celle de facilité, ne sont pas pleinement considérées en ce qui concerne les systèmes d'aide à la recherche d'information : le but est d'aider l'utilisateur et non de le submerger dans un flux trop important d'informations.

Dans la suite de ce chapitre, nous décrivons dans un premier temps notre démarche et l'approche adoptée sur l'aide à la recherche d'information. Puis, dans un second temps, nous présentons les étapes de notre méthodologie fondée sur une méthode de classification non-supervisée. Finalement, la liste des hypothèses ainsi que l'originalité de notre approche sont présentées.

4.4.1 Démarche et approche

Notre but est d'établir une méthode d'aide à la recherche d'information faisant intervenir l'utilisateur, dans un processus d'expansion de requête. Dans le Chapitre 2, un système classique de RI et le processus de recherche correspondant ont été évoqués. Nous remarquons que le processus s'avère itératif tant que l'utilisateur n'est pas satisfait des résultats de sa requête. Cette phase itérative se traduit alors, pour l'utilisateur, à une quête de la combinaison de mots et d'expressions qui lui permettront d'aboutir à l'information voulue. Cette phase est d'autant plus délicate que les mots et les expressions doivent être déterminés par l'utilisateur, à travers une phase de lecture des documents trouvés non pertinents par exemple.

Ainsi, notre approche intervient lorsque l'utilisateur entre dans cette boucle, et qu'il doit déterminer par lui-même les termes pour une nouvelle requête. En effet, notre approche est de proposer à l'utilisateur un ensemble de termes définissant la requête. Dans ce cadre, l'utilisateur choisira simplement le terme qui correspondra le plus à son besoin pour affiner sa requête.

De cet abord de l'aide à la recherche d'information, nous avons distingué, dans le Chapitre 2, les principales approches existantes. Elles sont au nombre de deux, dont l'une découle de calculs statistiques (co-occurrences par exemple) et l'autre procède à une

classification des documents (ex : Scatter/Gather). Ces deux approches différentes le sont également dans la présentation des résultats. En effet, les approches fondées sur une classification proposent en résultat les classes trouvées sous forme de listes de mots dans la plupart des cas, tandis que les approches statistiques proposent une liste de termes.

Notre démarche est de regrouper l'essentiel de ces deux méthodes, suivant ce qui nous semble le plus approprié pour chacune d'elles. Du point de vue de l'interaction, l'utilisation des syntagmes nominaux nous semble plus appropriée pour l'utilisateur. En effet, une liste de mots nécessite un temps de réflexion pour cerner le contexte. Du point de vue de la méthode, de nombreux travaux [Cutting et al., 1992], [Bellot, 1999] ont montré une amélioration de la pertinence des résultats en utilisant des classes. Ces classes sont alors présentées dans un ordre suivant un critère de pertinence.

La difficulté de cette approche réside dans le choix des termes à proposer à l'utilisateur et la quantité de termes à proposer. En effet, il ne faut pas submerger l'utilisateur sous un nombre trop important de termes, amenuisant ainsi de façon considérable le bénéfice de la méthode. La difficulté, au niveau du choix des termes, réside, d'une part, dans la représentation terminologique d'une classe et, d'autre part, dans le choix des classes à représenter. Faut-il représenter toutes les classes ou bien une partie d'entre elles, et dans ce cas, selon quel critère ?

4.4.2 Etapes de notre méthodologie

Nous venons de décrire notre approche de l'aide à la recherche d'information. Dans ce paragraphe, nous présentons les différentes étapes de notre méthodologie, c'est-à-dire du texte intégral au système d'aide.

Notre méthodologie consiste, à partir d'un corpus de documents, à déterminer un ensemble de termes représentatifs du corpus qui permettront à l'utilisateur d'affiner sa requête. Elle est constituée de deux modules principaux : un module d'acquisition de connaissances et un module d'exploitation des connaissances.

Le module d'acquisition des connaissances regroupe les différents traitements appliqués sur le corpus afin d'extraire les termes représentatifs du domaine. Ce module contient deux phases principales. La première phase est une extraction de connaissances de premier niveau à l'aide d'un outil d'extraction terminologique. La seconde phase est un regroupement de connaissances de second niveau fondé sur une méthode de classification non-supervisée.

Le module d'exploitation des connaissances est un ensemble de fonctions qui permet, à partir des connaissances extraites par le module d'extraction des connaissances, de déterminer un ensemble de termes décrivant une requête utilisateur.

Globalement, notre méthodologie est ainsi une succession d'étapes que nous décrivons ci-dessous.

4.4.2.1 Module d'extraction de connaissances

1. Choix et utilisation d'un corpus représentatif d'un domaine spécifique sur lequel on applique des traitements statistiques.
2. Représentation des documents : extraction des termes susceptibles de représenter le domaine.
3. Application d'une méthode de classification non-supervisée sur un tableau de contingence.
4. Etiquetage des classes par méthodes statistiques : extraction des termes clés.

4.4.2.2 Module d'exploitation des connaissances

1. Mise en place d'une méthode de recherche basée sur une classification.
2. Sélection des termes statiques de la classification.
3. Sélection des termes dynamiques liés à la requête.

Le schéma global de notre méthodologie est représenté par la Figure 4.1. Sur cette figure, le module d'exploitation est symbolisé par l'outil SearchXQ, développé dans le cadre de cette thèse, qui permet de fournir l'ensemble des termes pour une requête donnée.

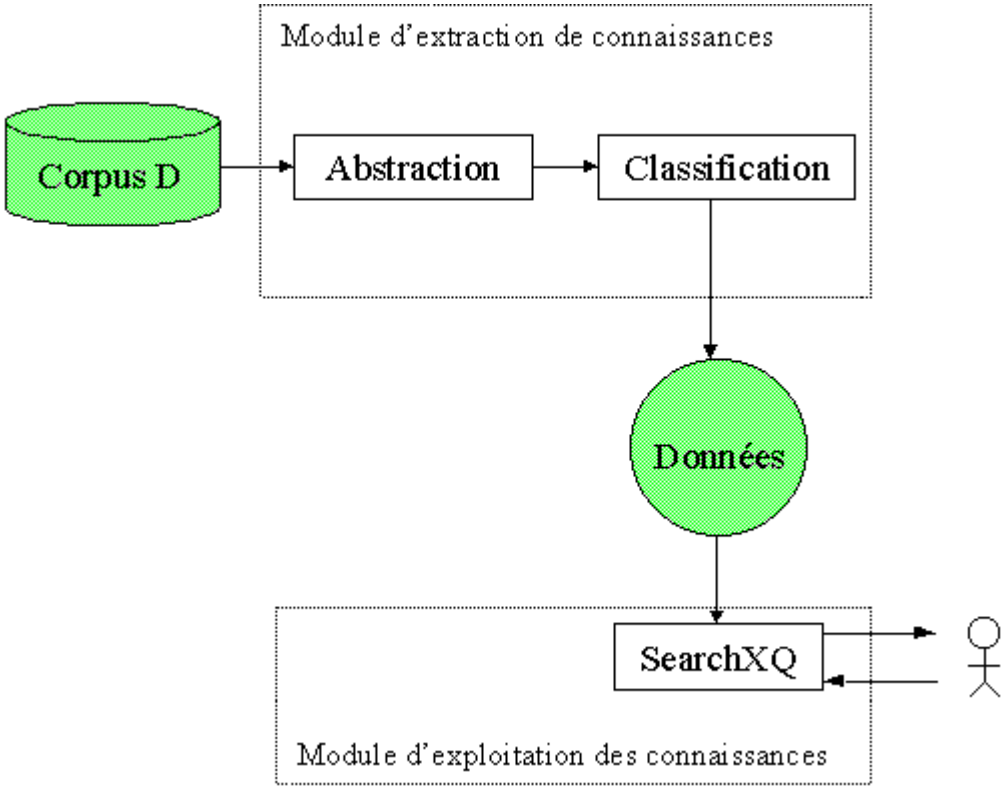

Figure 4.1 – Schéma de la méthodologie retenue

La phase d'abstraction est scindée en deux fonctions principales : l'extraction et le filtrage. La fonction d'extraction permet de représenter chaque document de *D* par un ensemble de termes. La fonction de filtrage regroupe un ensemble de techniques connues

telles que la réduction des termes par fonction de pondération, ou bien la réduction par lemmatisation. Cette phase est décrite dans le Chapitre 5.

La phase de classification permet, *in fine*, d'extraire, à partir de *D*, un ensemble de thématiques. Ces thématiques seront celles proposées à l'utilisateur dans le processus de recherche (*cf*. Figure 4.3). Toutefois, ce processus, décrit dans la section suivante, est itératif. Cette phase de classification est donc, en réalité, appliquée de façon itérative afin d'obtenir une hiérarchie de thématiques (*cf*. Figure 4.2). Cette phase est décrite dans le Chapitre 6.

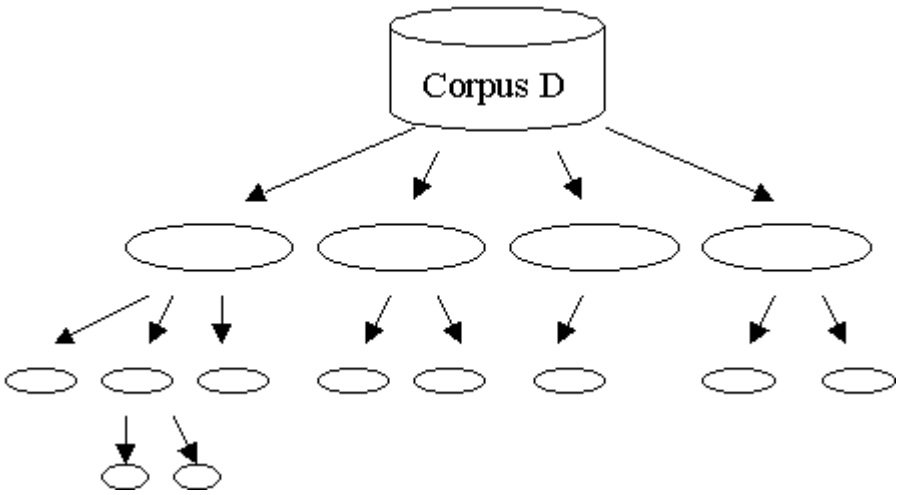

Figure 4.2 – Décomposition en hiérarchie de classes, et par conséquent, en thématiques

4.4.2.3 Processus de recherche

La Figure 4.3 expose le processus de recherche d'information incluant le module d'exploitation des connaissances, représenté par le module dénommé SearchXQ. Celui-ci diffère du processus classique (*cf*. Figure 2.7) par les moyens disponibles pour affiner la requête. En effet, dans ce processus, on constate que trois choix s'offrent à l'utilisateur, si la page de réponses ne convient pas, après une requête initiale. Il peut alors choisir un terme proposé par le module SearchXQ, ou bien un terme de la requête initiale ou expansée, ou bien reformuler une requête. Ces choix sont décrits dans l'ordre inverse de l'énumération ci-dessous.

- Le dernier cas est présent dans le processus classique et révèle, par son utilisation, un manque de pertinence de notre module. L'utilisateur, à partir des résultats d'une requête *R*, reformule de lui-même une nouvelle requête *R'*.
  R ∩ R' peut être l'ensemble vide si l'utilisateur change complètement de requête.
- Le second cas permet de faire un retour en arrière, c'est-à-dire que si la requête *R* est de la forme $M \,\&\, T_1 \,\&\, T_2 \,\&\, T_3$, alors l'utilisateur peut revenir à tout moment à l'une des formes antérieures de *R*, à savoir : $\{M \,\&\, T_1 \,\&\, T_2, M \,\&\, T_1, M\}$. Ce cas permet d'appliquer l'un des principes évoqués à la section 2.9 du Chapitre 2 sur les interfaces. La nouvelle requête *R'* a la propriété suivante :
  $R' \subset R$

- Le premier cas permet à l'utilisateur d'affiner sa requête par l'un des termes proposés par le module. La nouvelle requête *R'* sera de la forme : $R' = T_i \,\&\, R$, si T_i est le terme choisi. Pour cette nouvelle requête, une liste de termes correspondants sera à nouveau proposée ainsi qu'une nouvelle liste de documents. L'utilisateur peut alors choisir un nouveau terme T_j pour affiner de nouveau et si nécessaire la requête, et ainsi de suite. L'objectif est de converger en quelques itérations vers un petit ensemble de documents. Les termes ainsi proposés devront être discriminatifs. La nouvelle requête R', à chaque itération, possède la propriété suivante :

  $R \subset R'$

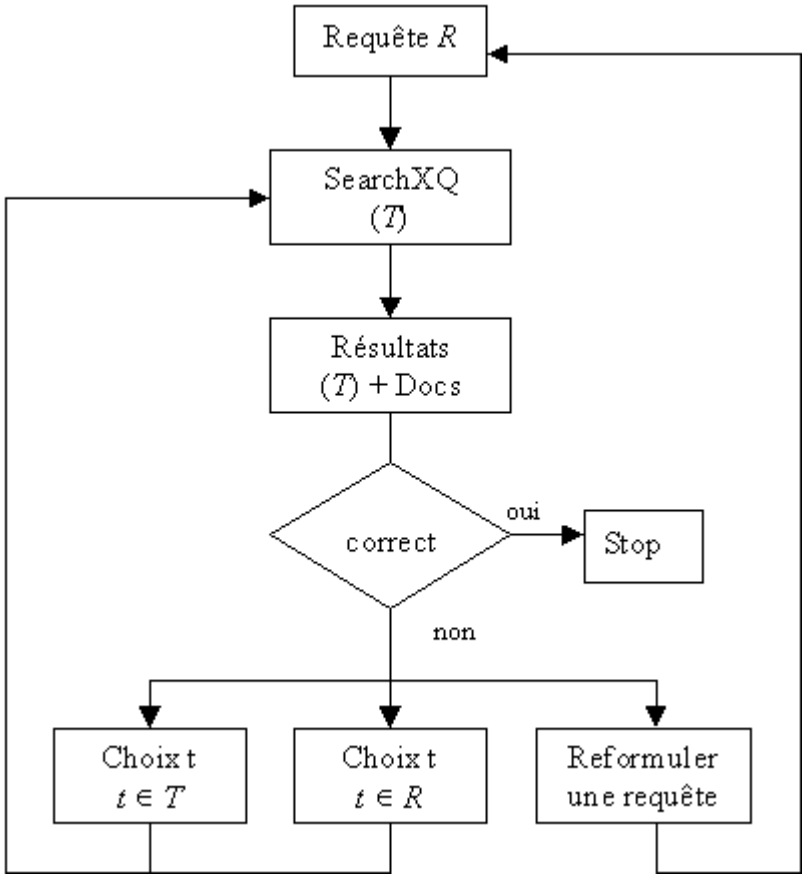

Figure 4.3 – Processus de recherche.

4.4.3 Hypothèses

Pour les systèmes de recherche d'information qui représentent les documents à l'aide du modèle vectoriel, certaines hypothèses sont posées [Bellot, 2000]. Les hypothèses utilisées pour notre méthode sont examinées dans cette section.

Hypothèse 1 : *À toute thématique correspond une distribution singulière des termes, c'est-à-dire que des termes sont plus fréquents dans les documents d'une thématique que dans les documents des autres thématiques.*

Cette hypothèse est utilisée de différentes façons dans les systèmes de recherche d'information. Généralement, elle l'est pour mesurer une distance entre deux *individus* différents où un individu est assimilable à un document, un ensemble de documents ou une requête : distance entre deux documents par exemple.

Cette distribution singulière des termes est également utilisée comme fondement à la détermination automatique de thèmes des classes : étiquetage thématique et automatique d'un corpus par des méthodes statistiques.

Hypothèse 2 : *Deux documents proches*[1] *abordent la même thématique.*

Cette hypothèse n'est évidemment pas toujours vérifiée. Un tuple de documents ayant une certaine quantité de vocabulaire en commun n'abordera pas nécessairement la même thématique : un vocabulaire similaire mais utilisé dans des contextes différents, ou utilisation d'un sens différent des mots, c'est-à-dire la problématique de la polysémie des mots. Dans un modèle vectoriel, cet effet est non négligeable : pendant la phase d'indexation, le contexte du document n'est généralement pas pris en compte. Notons toutefois que l'effet est d'autant plus atténué que la quantité de vocabulaire en commun est importante entre deux documents. Par conséquent, cet effet est surtout discernable pour des documents de petite taille. Dans le cadre de nos expérimentations sur le corpus de référence, décrit dans la section 4.3, les documents sont enclins à ce phénomène. En effet, la moyenne des documents n'excède pas 10 Ko.

Hypothèse 3 : *Les documents pertinents d'une requête sont proches entre eux.*

Dans [Bellot, 2000], cette hypothèse est scindée en deux hypothèses, réciproque l'une de l'autre. La première concerne l'hypothèse de classification [van Rijsbergen, 1979] : « *Closely associated documents tend to be relevant to the same requests* »[2]. Cette hypothèse est une extension de l'hypothèse 2 énoncée ci-dessus. Cette extension est relative à l'introduction de la notion de pertinence et de la considération de la requête utilisateur comme un document. La seconde hypothèse concerne la proximité (en terme de distance) des documents pertinents pour une requête donnée : les documents pertinents, pour une requête donnée, sont proches entre eux et éloignés des documents non pertinents. Ces deux hypothèses ne sont pas toujours vérifiées. Dans le cas où les deux hypothèses seraient invariablement vérifiées, les conséquences en seraient les suivantes :

- Les performances pour les moteurs de recherche, en terme de temps de réponse, seraient améliorées car il suffirait de confronter la requête uniquement avec les classes d'une pré-classification.

[1] La notion de proximité est liée au modèle de représentation des documents. Dans le cas du modèle vectoriel, celui que nous avons choisi, la proximité est représentée par la distance qui sépare les deux documents considérés.
[2] Les documents proches entre-eux tendent à être pertinents pour les mêmes requêtes.

- Les documents pertinents pour une requête se situeraient dans une même zone de l'espace vectoriel : les documents proches d'un document pertinent sont pertinents.
- Dans le cas d'une classification locale, une classification des documents en deux classes (une classe de documents pertinents et une de documents non pertinents) permettrait de retourner uniquement les documents pertinents.

Les travaux de van Rijsbergen [1979] tentent de confirmer une répartition distincte des documents pertinents et des documents non pertinents. Cette répartition se fait à partir d'un calcul de distances entre documents pertinents et documents non pertinents pour des requêtes. Selon [Salton, 1994], l'hypothèse de la proximité des documents pertinents ne peut être vérifiée puisqu'il n'existe aucun lien entre le concept de pertinence et le modèle vectoriel : durant la phase d'indexation, la notion de pertinence de document n'intervient pas. Les documents pertinents et les documents non pertinents peuvent donc se trouver dans la même zone de l'espace vectoriel. Les expérimentations récentes [Cutting et al., 1992], [Hearst et Pederson, 1996] valident, au contraire, le fait qu'il existe en partie une séparation entre les deux ensembles de documents (pertinents et non pertinents). Les travaux de [Bellot, 2000] montrent que les documents pertinents sont situés dans plusieurs classes à l'intérieur desquelles les documents sont proches entre eux. Les raisons d'une telle répartition se justifient par le fait qu'une thématique peut se définir de différentes façons ; les notions de synonymes [Hamon et Nazarenko, 2001], d'hyponymes [Morin, 1999] et d'hyperonymes doivent être prisent en compte. Ainsi, l'ambiguïté sémantique que l'on retrouve dans les documents ne permet pas de regrouper les documents pertinents entre eux pour une requête donnée.

Conséquences

Ces deux hypothèses sur la proximité des documents pertinents, pour une requête donnée, s'inscrivent dans le schéma question/ « réponse *directe* ». Bien que ce schéma ne soit pas celui que nous adoptons, il permet de nous interroger sur certains points de notre démarche.

La première interrogation concerne le vocabulaire du corpus de référence. Les travaux de Lame [Lame, 2002] montrent que le langage juridique est concerné principalement par des phénomènes de polysémie. Dans ce cas, l'une des principales difficultés liées à ce corpus est la détection des différents contextes pour un terme juridique.

La seconde interrogation concerne l'une des étapes de notre module d'exploitation des connaissances et plus précisément la sélection des termes statiques de la classification. Cette sélection de termes doit faire face aux phénomènes évoqués ci-dessus : la synonymie par exemple. Il est, en effet, inconcevable de proposer à l'utilisateur des termes différents mais évoquant la même sous-thématique de la requête. Ainsi, les documents pertinents, pour une requête donnée, qui se trouvent dans des classes différentes le seront en raison de contextes différents et non en raison du vocabulaire différent employé.

En résumé, dans nos expérimentations, tous les documents pertinents ne se retrouveront pas dans la même zone de l'espace vectoriel, pour des raisons de polysémie principalement. En conséquence de ce phénomène de polysémie, le concept évoqué par Salton [Salton, 1994], concernant la proximité des documents pertinents et des documents non pertinents, est vérifié dans notre cas.

Les hypothèses citées précédemment font principalement référence aux systèmes de recherche d'information. Nous posons, ci-dessous, une liste d'hypothèses de notre méthodologie.

Hypothèse 4 : *Le corpus de référence est adapté aux traitements statistiques.*

Le corpus de référence, décrit dans la section 4.3, est adéquat pour les traitements statistiques suivants :

- extraction de syntagmes nominaux représentatifs du domaine ;
- classification de documents : méthodes de partitionnement et méthodes hiérarchiques ;
- étiquetage des classes.

Notre module d'extraction de connaissances ainsi que la validation de nos expérimentations pour ces différents traitements confirment que notre corpus est adéquat aux traitements. Une validation faisant intervenir des utilisateurs est, cependant, utile pour le module d'exploitation des connaissances.

Hypothèse 5 : *Les thématiques des classes peuvent être déterminées par des méthodes statistiques.*

4.5 Conclusion

Dans ce chapitre, nous avons évoqué, dans un premier temps, les motivations concernant le choix du corpus de référence. La première est relative à l'essence même du corpus. Le corpus des codes offre un vocabulaire juridiquement plus intéressant que le corpus du Journal Officiel. Dans notre approche d'aide à la recherche d'information, cette caractéristique sera primordiale. La seconde motivation ne concerne plus l'aspect juridique du corpus mais son aspect intrinsèque. La classification thématique du corpus des codes sera la base de l'évaluation de notre méthode de classification. Bien que nos travaux soient essentiellement fondés sur le corpus des codes pour les raisons évoquées ci-dessus, l'utilisation d'autres corpus juridiques est, toutefois, envisageable.

Dans un second temps, nous avons évoqué notre méthodologie. Le cœur de celle-ci repose sur l'application d'une méthode de classification de façon hiérarchique afin d'obtenir une hiérarchie de thématiques. Dans le chapitre suivant, nous exposons cette méthode de classification et les expérimentations menées sur le corpus de référence.

Chapitre 5

Extraction de termes

Résumé

La première phase d'une méthode de classification de documents est l'abstraction des documents composant le corpus. Durant cette phase, on extrait pour chaque document un ensemble de termes. Ces derniers subissent a posteriori un filtrage dans le but de réduire la taille de l'espace du point de vue pratique, mais surtout de réduire certaines ambiguïtés.

Dans ce chapitre, nous nous intéressons, dans un premier temps, aux différentes méthodes d'extraction de termes. Puis, dans un second temps, nous essayons de mettre en place une méthode appropriée au domaine juridique dans le but d'abstraire les documents uniquement avec des termes juridiques.

Ce chapitre permet de mettre en évidence que notre domaine d'application ne permet pas d'appliquer des méthodes classiques de réduction de termes, telles que la réduction par synonymie, sans perte d'informations utiles.

5.1 Introduction

Nous avons mentionné laconiquement, dans la section 2.4 du Chapitre 2, des méthodes et des outils d'extraction de termes ; parmi les outils disponibles, deux analyseurs syntaxiques ont été décrits succinctement.

Nous présentons, plus en détail, les différentes approches de l'extraction de termes, et plus précisément de l'identification de termes. Cette identification est fondée sur deux grandes approches : linguistique et statistique. De plus, des approches hybrides associant linguistique et statistique sont également présentées. Ces méthodes hybrides sont au cœur de ce chapitre. En effet, dans le présent chapitre, nous proposons de mettre en place une méthode d'extraction de termes appropriée au domaine juridique. Cette méthode utilise, dans un premier temps, un extracteur terminologique et, dans un second temps, une méthode de filtrage de termes. A travers cette méthode de filtrage, nous tentons d'exploiter les deux approches d'identification de termes pour s'approcher le plus possible d'une liste de termes juridiques. Des travaux avec la seule approche statistique ont déjà été menés sur un corpus juridique. Nous tentons d'ajouter une composante linguistique de base, à travers la détection des suites de catégories grammaticales des termes du domaine juridique.

Dans ce chapitre, nous abordons également l'aspect de « l'extraction de termes », tout d'abord à travers son contexte, c'est-à-dire les présupposés liés à l'extraction ainsi que les principales structures grammaticales des termes d'un corpus de langue française. Puis, nous présentons les principales approches citées ci-dessus (statistique et linguistique), et certaines approches hybrides représentées notamment par ACABIT, l'outil d'extraction de Daille [1994]. Dans une seconde partie, nous présentons les caractéristiques du domaine juridique ainsi que des travaux de nature statistique. Puis, nous examinons les structures grammaticales des termes de ce domaine pour essayer d'en déterminer les principaux représentants. Nous présentons, ensuite, des travaux de filtrage fondés sur des notions classiques de la linguistique. Enfin, nous présentons notre méthode d'extraction de termes qui prend en compte les considérations et les résultats des expérimentations sur ce domaine.

5.2 Extraction de termes

Dans cette section, nous développons les différents aspects de l'extraction de termes dont une partie a été vue dans le Chapitre 2, sous forme de quelques exemples d'outils.

5.2.1 Contexte

L'extraction de termes est une étape fondamentale dans un processus de recherche d'information ; une partie de la qualité des résultats en dépend. Les différentes approches d'extraction de termes, énumérées dans le paragraphe suivant, s'appuient sur les présupposés

suivants (ou sur une partie d'entre eux, selon la stratégie d'extraction adoptée) [L'Homme, 2001] :

1. les textes d'un corpus, dans le cas d'un domaine spécialisé, comportent un ensemble de termes qui caractérisent les connaissances spécialisées du domaine : les termes du domaine ;

2. un terme du domaine sera utilisé à plusieurs reprises dans un texte spécialisé : le terme possède une fréquence plus importante dans ce texte que les autres termes, et/ou le terme est plus fréquent dans ce texte spécialisé que dans un texte *général* ;

3. la plupart des termes sont des syntagmes nominaux ;

4. la plupart de ces termes sont dits complexes, c'est-à-dire qu'ils sont composés de plusieurs mots. Ces mots sont par ailleurs utilisés isolément (ex. *bien immobilier, révision des traités*, *contrat de travail*) ;

5. les termes complexes se construisent au moyen d'un nombre fini et restreint de séquences de catégories grammaticales. En effet, généralement, les termes complexes français se composent d'un nom (conséquence du présupposé 3) modifié par un :
   - adjectif (NA) : ex. *loi nationale, usure normale* ;
   - un syntagme prépositionnel contenant un nom : ex. *prix de rachat, sirop d'inuline* ;
   - un syntagme prépositionnel contenant un verbe (NppV) : ex. *terrain à boiser* ;
   - un autre nom (NN) : ex. *conseil de famille* ;
   - n'importe quelle combinaison des séquences ci-dessus : ex. *méningite cérébro-spinale à méningocoques.*

   Il s'agit donc d'un sous-ensemble de syntagmes nominaux.

Dans le Tableau 5.1, on présente, sous forme de séquence de catégories grammaticales, le sous-ensemble de syntagmes nominaux que l'on rencontre le plus souvent dans les documents français (après une phase d'extraction de candidats termes) :

Abréviation	Séquence de catégories grammaticales	Exemple
NA	Nom + Adjectif	Marine nationale
NppN	Nom + Préposition + Nom	Contrat de travail
NppV	Nom + Préposition + Verbe	Machine à timbrer
NN	Nom + Nom	Bien meuble
NppdN	Nom + Préposition + Déterminant + Nom	Loi sur l'enfance

Tableau 5.1 – Les séquences de catégories grammaticales les plus fréquentes dans un corpus français

Les séquences présentées dans le Tableau 5.1 peuvent être considérées comme des éléments de base. En effet, parmi les séquences les plus courantes, on retrouve également toute combinaison plus longue de ces séquences, dont des exemples sont donnés dans le Tableau 5.2.

Abréviation	Séquence de catégories grammaticales
NNA	Nom + (Nom + Adjectif)
NppNppN	Nom + Prép. + Nom + Prép. + Nom

Tableau 5.2 – Autres séquences fondées sur des séquences de base.

Dans notre méthode d'extraction décrite dans la section 5.4, une des fonctions de filtrage est la suppression des mots vides tels que les pronoms et les déterminants. Ainsi, les séquences les plus courantes sont résumées dans le Tableau 5.3.

Abréviation	Séquence de catégories grammaticales	Exemple
NA	Nom + Adjectif	Marine nationale
NN	Nom + Nom	Contrat travail
NV	Nom + Verbe	Machine timbrer
NN	Nom + Nom	Bien meuble
NNA	Nom + Nom + Adjectif	Montant taxe exigible
NNN	Nom + Nom + Nom	Application code procédure

Tableau 5.3 – Les séquences les plus courantes après réduction

Bien que les termes soient réduits, les pointeurs des termes réduits vers les formes dérivées sont stockés.

5.2.2 Identification

L'identification (ou l'extraction) automatique des termes d'un document fait appel à des techniques regroupées généralement dans deux catégories :

- les approches *linguistiques* ;
- les approches *statistiques*.

D'autres approches d'identification appelées hybrides existent également. Ces approches hybrides, combinaison de traitements linguistiques et statistiques, sont également décrites brièvement dans cette section.

- Les approches linguistiques associent des informations linguistiques à des chaînes de caractères ou font appel à des connaissances sur la langue traitée (au moins minimales) et recherchent, le plus souvent, des suites de catégories grammaticales.

- Les approches statistiques effectuent des calculs sur les chaînes de caractères et s'appuient sur le fait que des termes significatifs sont employés assurément plus d'une fois dans un texte. Toutefois, dans la plupart des outils existants, les connaissances linguistiques et les calculs statistiques sont combinés selon différentes modalités. Cette section est un complément de la section 2.4 du Chapitre 2, dans laquelle certains de ces outils sont notamment présentés.

5.2.2.1 Approches linguistiques

Les approches linguistiques se fondent avant tout sur le fait que les termes complexes sont des syntagmes nominaux (présupposé 3 de la section 5.2.1), composés de suites de catégories grammaticales régulières. Ils repèrent donc des séquences de mots correspondant à des patrons préalablement définis. Par exemple, si on demande à un extracteur de termes de retenir les suites « nom + adjectif » et « nom + prép. + nom », il s'arrêtera sur deux séquences dans la phrase suivante et les placera dans une liste proposée à l'utilisateur.

Exemple :
Phrase : *L'ordinateur portable est en général plus coûteux que l'ordinateur de bureau.*

Séquences : *ordinateur de bureau*
ordinateur portable

Une variante à cette première stratégie linguistique consiste à envisager le problème à l'envers. Plutôt que d'exploiter des connaissances « en positif », c'est-à-dire chercher des syntagmes nominaux correspondant à des suites de catégories grammaticales précises, on exploite des connaissances « en négatif ». Cette technique, mise au point par Bourigault [1993], consiste à appliquer au texte une série de règles de découpage. Les coupes sont pratiquées là où le programme rencontre une unité qui ne peut pas faire partie de termes complexes, comme un signe de ponctuation fort, un pronom ou un verbe conjugué. Pour certains mots, le logiciel recourt à d'autres éléments de la phrase. Par exemple, certaines prépositions sont éliminées si elles sont suivies d'un possessif ; les participes passés sont éliminés s'ils sont suivis d'une préposition. L'exemple ci-dessous montre comment les syntagmes nominaux sont délimités à partir de frontières.

Exemple :
Le circuit d'aspersion de l'enceinte de confinement / assure le / maintien / de sa / température nominale de fonctionnement / après une / augmentation de pression.
[Bourigault, 1993]

Les deux techniques linguistiques décrites ci-dessus s'appuient sur une reconnaissance des catégories grammaticales des mots dans le texte. Cette reconnaissance se fait de deux manières :

- par la consultation d'un dictionnaire dans lequel on associe, à chacun des mots simples, sa catégorie grammaticale (certains mécanismes de « désambiguïsation » sont parfois mis en œuvre pour régler les problèmes posés par des mots ambigus) ;
- par la consultation d'un texte étiqueté renfermant les mentions explicites des catégories grammaticales (la désambiguïsation est alors faite par l'étiqueteur).

La Figure 5.1 illustre ces deux méthodes d'explicitation des catégories grammaticales.

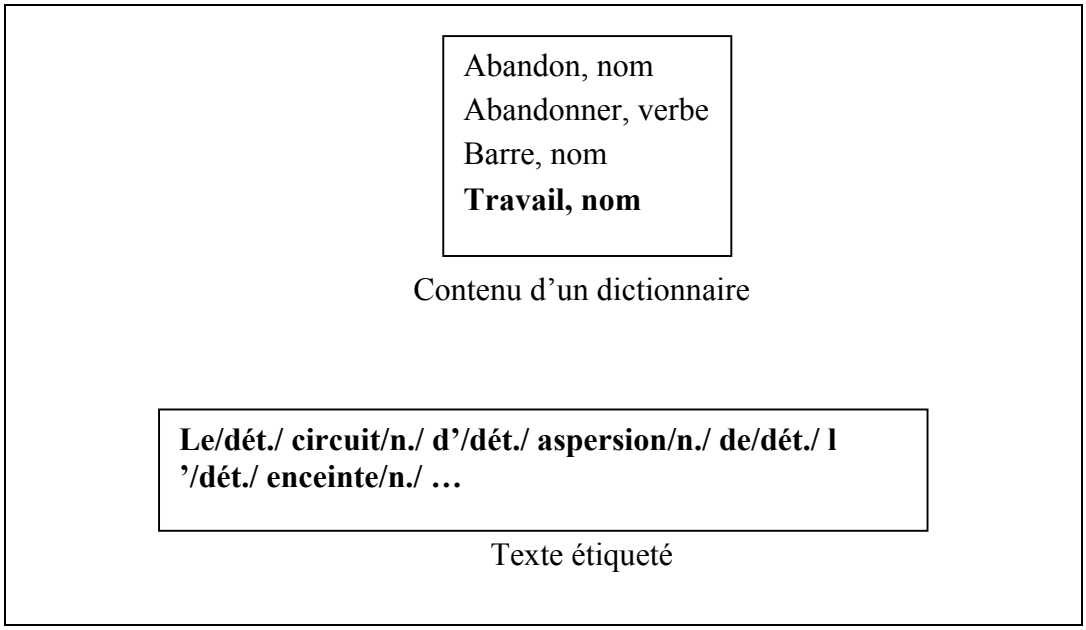

Figure 5.1 – Reconnaissance des catégories grammaticales

5.2.2.2 Approches statistiques

Les stratégies statistiques (ou probabilistes) s'appuient principalement sur le principe suivant : un terme significatif est employé plus d'une fois dans un texte spécialisé. Il existe plusieurs méthodes statistiques appliquées à l'extraction terminologique qui sont pour la plupart fondées sur un principe central, appelé *information mutuelle*. *Grosso modo*, ce principe veut que l'association récurrente de deux mots est assurément significative et ne peut être considérée comme « le fruit du hasard ».

Concrètement, les occurrences des mots d'un texte sont examinées de la manière suivante : si un mot *X* apparaît plus fréquemment dans l'entourage d'un mot *Y* qu'ailleurs dans le texte, alors *X* et *Y* forment une combinaison significative.

5.2.2.3 Approches hybrides

À quelques rares exceptions, les outils d'extraction de termes combinent les deux stratégies. On parle alors d'extraction faisant appel à des stratégies hybrides ou mixtes : les extracteurs génèrent une liste de termes à partir d'informations linguistiques et épurent la liste au moyen de calculs statistiques ; soit, au contraire, ils établissent une première liste de termes

au moyen de calculs sur des chaînes de caractères et exploitent par la suite de l'information linguistique. Par exemple, ACABIT [Daille, 1994] recherche des suites de catégories grammaticales dans des textes étiquetés (connaissances linguistiques) et fait des calculs statistiques sur les termes préalablement extraits (stratégie probabiliste).

Selon [L'homme, 2001], les listes produites par les extracteurs contiennent des anomalies, quelle que soit la stratégie adoptée. Elles comportent des suites qui ne sont pas celles qu'on recherche (ces erreurs sont regroupées sous le générique *bruit*). Par ailleurs, les extracteurs passent outre des termes pourtant corrects. Ce deuxième groupe d'erreurs est appelé *silence*. Les concepteurs ne peuvent corriger toutes les erreurs, compte tenu de la difficulté liée à la tâche d'extraction, mais tentent de trouver des méthodes pour réduire le silence ou le bruit en fonction d'une application donnée.

Deux mesures, empruntées au domaine de la recherche d'information et décrites dans le Chapitre 2, servent à calculer la performance des extracteurs de termes. La *précision* évalue la proportion de termes corrects présents dans la liste produite par l'extracteur ; le *rappel* mesure la proportion de termes relevés par rapport aux termes présents dans le texte traité. Plusieurs travaux tentent de raffiner l'extraction de termes qui se limite souvent à la production d'une liste de termes complexes. Voici un aperçu de quelques aspects :

1. Extraction de termes simples

Les termes simples sont extrêmement difficiles à repérer puisque rien dans leur forme ne permet de les distinguer des autres unités lexicales présentes dans les textes spécialisés. Toutefois, on croit que leur fréquence d'utilisation peut servir d'indicateur fiable. Une technique simple consiste à présenter les mots simples par ordre décroissant de fréquence (avec exclusion des mots grammaticaux) en tenant pour acquis que les mots en tête de liste sont révélateurs du contenu d'un texte.

Une seconde stratégie consiste à comparer la fréquence des mots simples présents dans un corpus de référence (par exemple un corpus de textes informatiques) à celle de mots apparaissant dans une collection (par exemple, un collection de textes journalistiques). Les mots dont la fréquence est nettement plus élevée dans le corpus de référence risquent fort d'être des termes significatifs (ou des termes du domaine).

2. Regroupements de termes préalablement extraits

Différentes techniques tentent de rassembler des termes extraits en fonction de leur parenté formelle ou sémantique. Il est parfois difficile de retrouver manuellement les termes apparentés dans une liste qui les présente par fréquence décroissante ou par ordre alphabétique.

Une technique consiste à retrouver des termes qui ont des composantes communes. La Figure 5.2 (dans [L'homme, 2002]) montre comment les termes apparentés à *logiciel d'application* sont proposés à l'utilisateur. L'outil recherche tous les autres termes comportant

logiciel (comme tête ou modificateur) et fait de même pour *application*. Cette liste peut être produite par simple comparaison graphique des termes entre eux.

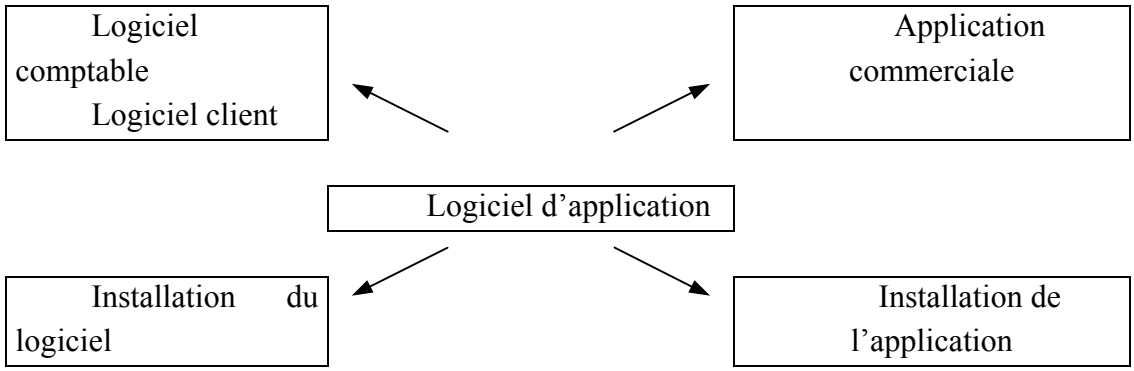

Figure 5.2 – Termes associés à « logiciel d'application »

D'autres travaux prennent en compte la variation terminologique, c'est-à-dire les multiples réalisations du terme qui peuvent aller d'un changement graphique (*système expert* et *système-expert*, par exemple) ou de transformations morpho-syntaxiques (ex. *ulcère de cornée*, *ulcère cornéen* ; *réseau pour données*, *réseau à données* [Daille, 1995]) à des modifications syntaxiques beaucoup plus importantes, comme la coordination (ex. *artère rénale* et *artère coronaire* dans *artères fémorales, rénales et coronaires*) ou les transformations syntaxiques (ex. *concentrate measurement* dans le terme *measured COHb concentration, measured the concentration* ou *measuring the concentration* [Jacquemin, 2001]). L'appariement des variantes est nettement plus ardue à réaliser que le regroupement des termes ayant des composantes communes et repose sur un appareillage linguistique élaboré [Habert et al., 1997].

5.3 Les termes du domaine

Dans cette section, nous appliquons les approches d'identification de termes, décrites dans la section précédente, au domaine juridique tout en respectant les contraintes du vocabulaire. Ce vocabulaire est précis et certains traitements bien qu'applicables peuvent conduire à des résultats hasardeux.

Notre objectif est donc de déterminer si ce vocabulaire renferme des caractéristiques statistiques et linguistiques qui lui sont propres.

5.3.1 Identification statistique

L'ensemble initial des termes, généré par des outils d'extraction de termes, doit être filtré à plusieurs niveaux (voir section 5.2.2) : statistique et linguistique. L'objectif de l'application de ces techniques de filtrage sur l'ensemble initial est d'éliminer les termes qui ne sont pas intéressants du point de vue juridique, c'est-à-dire de s'approcher le plus possible de la liste des termes du domaine. Les travaux de Lame [2002] sur le filtrage des candidats

termes sont de nature statistique principalement. Toutefois, ces calculs statistiques ont été réalisés uniquement sur des syntagmes nominaux. L'objectif de l'auteur était d'identifier statistiquement l'ensemble des termes du domaine par l'application de différentes fonctions de pondération sur les candidats termes, afin de déterminer l'existence éventuelle d'un seuil au-delà duquel il n'existe plus de termes juridiques. L'auteur aboutit à la conclusion qu'il n'est pas possible de déterminer statistiquement les termes juridiques de l'ensemble initial : « Nous ne pouvons distinguer dans notre corpus de référence les termes juridiques des autres ; peut-être est-ce dû au fait que tous les termes extraits de ce corpus par les outils extracteurs peuvent être considérés comme des termes juridiques. »

5.3.2 Caractéristiques des termes juridiques

Selon [Lame, 2002], « les termes du domaine sont des termes qui représentent des artefacts juridiques ». Les termes du domaine concernent les termes spécifiques à la matière juridique, tels que *licéité*, *stellionataire*, *colon partiaire*, *répétition de l'indu* ou *tontine*. D'autres termes du domaine sont également des termes ayant un sens juridique et sont utilisés dans ce sens dans le langage commun : *contrat*, *mariage*, *testament*, *société à responsabilité limitée* ou *avocat* etc. Enfin, les termes du domaine rassemblent également des termes polysèmes ayant un sens propre et un sens autre dans le vocabulaire juridique, par exemple *absence*, *minute* ou *grosse* [Cornu, 2000]. Par conséquent, les termes du domaine réunissent un ensemble de termes qui présentent au minimum un sens juridique.

L'auteur ajoute : « Par ailleurs, le droit est une discipline qui interprète le monde. Les objets du monde sont à ce titre appréhendés par le droit et sont exprimés, de notre point de vue, par des termes du domaine, ainsi *machine*, *médicament*, *passager* ou *clôture*. Nous entendons donc comme termes du domaine à la fois les termes représentant des artefacts juridiques et des objets du monde appréhendés par le droit. »

Selon Schmidt [1997], « le vocabulaire juridique, dont la complexité résulte de sa technicité se doit automatiquement d'être précis. En vertu de cette précision, le juriste n'a pas le droit d'employer un mot pour un autre, erreur qui pourrait être pour lui catastrophique ». Elle ajoute que « la précision des termes juridiques permet aux juristes d'employer un terme sans en expliquer sa portée, sa signification et parfois même son contexte ». L'exemple donné pour illustrer ces propos, est le terme « mis en examen ». En effet, ce terme indique que la personne concernée est soupçonnée d'une infraction de droit pénal.

Le vocabulaire juridique est précis, et l'absence de contexte d'un terme peut conduire à une méprise pour le novice. Certains termes juridiques ont également une autre définition en droit que dans le vocabulaire courant. Un exemple classique de cette difficulté est le terme « meuble » qui dans le vocabulaire courant représente aussi bien une chaise, une table, etc. Ce terme couvre une notion plus vaste dans le vocabulaire juridique. Il représente une chaise (meuble meublant), un animal (bien corporel pouvant être déplacé), une récolte sur pied, etc.

Outre que le vocabulaire est précis et technique, où chaque terme a une signification juridique particulière et fait appel à des notions bien précises du droit, le vocabulaire juridique est de plus en constante évolution. Par exemple, le terme de « personne mise en examen » vient aujourd'hui remplacer celui d' « inculpé ».

Ces points de vue ne sont guère exhaustifs, mais ils tentent de converger sur certaines caractéristiques du vocabulaire. Les termes du domaine ont un minimum de sens juridique. Ils sont précis et techniques mais sont sujets à la polysémie avec le vocabulaire courant.

Ces considérations ont des répercussions relativement faibles sur notre méthode, tant que l'on reste dans ce domaine. Toutefois, les répercussions sur l'interface homme-machine sont réelles pour le novice.

5.3.3 Importance des termes du domaine dans une méthode de classification

Le choix des termes a une certaine importance dans une méthode de classification classique. En effet, pour la plupart des méthodes, la réduction du nombre de termes permet de réduire la dimension de l'espace vectoriel. Le temps de calcul et le stockage s'en trouvent améliorés. Cette réduction concerne généralement les termes vides (*cf.* section 2.5.1 du Chapitre 2), et la notion de termes du domaine intervient peu.

Dans notre corpus, les termes ont plus d'importance par la nature même de la méthode de classification. En effet, nous devons passer du tableau de données X à une matrice de distances D creuse (*cf.* section 3.2 du Chapitre 3). Cette transformation est réalisée à l'aide d'une réduction des variables, c'est-à-dire des termes. Toutefois, cette réduction ne doit pas altérer la finalité de notre méthode : retrouver une hiérarchie de thématiques en utilisant les termes du domaine.

Nous devons donc déterminer les moyens les plus adéquats pour obtenir une matrice de distances D creuse, en utilisant les approches décrites ci-dessus. Cette réduction ne doit pas éliminer des termes du domaine, ce qui serait préjudiciable à la précision du résultat final.

5.3.4 Filtrage linguistique

Nous étudions dans ce paragraphe les caractéristiques morpho-syntaxiques du domaine du droit afin d'identifier les principales séquences de catégories grammaticales, c'est-à-dire les séquences les plus fréquentes dans le domaine juridique français.

N'ayant pas de ressource terminologique couvrant tout le domaine juridique à disposition, nous avons, dans un premier temps, pris en compte le lexique constitué par G. Lame [Lame, 2002]. Ce lexique, comportant 1489 termes juridiques, est un regroupement de différents lexiques juridiques disponibles sur le Web. Certains termes ne peuvent pas recevoir une étiquette syntaxique non ambiguë (« associé » est à la fois un nom et un participe). Pour déterminer l'étiquetage syntaxique des termes, nous avons utilisé les lexiques de *Pertimm*.

Cette analyse par les lexiques a permis de caractériser la plupart des termes du lexique juridique. En effet, dans le Tableau 5.4, on remarque que 3 % des termes ne sont pas reconnus (labellisé par la lettre I). Ces termes sont le plus souvent des acronymes : *pacs*, *sci*, etc. A noter que certaines ressources linguistiques sur les acronymes existent.

Classe morpho-syntaxique	Corpus_1489		Corpus_1704	
	#occ	%	#occ	%
Nom	592	39.76	630	36.97
Nom Nom	329	22.10	406	23.83
Nom Adj	129	8.66	168	9.86
Adj	92	6.18	95	5.58
Nom Nom Nom	61	4.10	76	4.46
I	48	3.22	51	2.99
Mot Composé	38	2.55	38	2.23
Nom Nom Adj	27	1.81	31	1.82
Verbe	24	1.61	26	1.53
Nom Nom Nom Nom	14	0.94	17	1.00
Nom Adj Nom	11	0.74	14	0.82
Nom Verbe	9	0.60	11	0.65

Tableau 5.4 – Distribution des termes juridiques les plus fréquents suivant la classe morpho-syntaxique

Dans un second temps, nous avons enrichi le lexique de base, composé de 1489 termes, par plus de 200 termes juridiques provenant notamment de lexiques juridiques de domaines spécifiques du droit tel que l'agriculture[1]. Les termes juridiques sont énumérés dans l'annexe B.

On remarque dans le Tableau 5.4 que l'enrichissement de ces quelques centaines de termes n'a pas altéré l'ordre, par fréquence décroissante, des séquences les plus courantes. On peut donc constater que la terminologie du domaine juridique est composée des mêmes séquences d'un domaine général dont le noyau est composé des 3 séquences principales suivantes : NV, NA, NN.

A partir de cette constatation, nous pouvons ainsi appliquer un filtrage linguistique de base sur l'ensemble des candidats termes. Ce filtrage consistera, en partie, à éliminer de la liste des candidats termes, ceux dont la séquence n'est pas uniquement composée des 3 séquences principales citées ci-dessus. Par conséquent, ce filtrage éliminera les séquences suivantes : NAAN, VN, etc.

[1] http://www.agriculteursdefrance.com/template_lexic_conseil.php

Classe	#occ	%
Nom Nom	22636	19.65
Nom Nom Nom	11368	9.87
Nom Nom Nom Nom	3128	2.72
Nom Adj Nom	2399	2.08
Nom Adj	957	0.83
Nom Adj Nom Nom	727	0.63
Nom I	677	0.59
Nom Nom Adj Nom	471	0.41
Nom Nom Adj	461	0.40
Nom I Nom	382	0.33

Tableau 5.5 – Extrait de la distribution des termes juridiques

Le Tableau 5.5 regroupe les principales séquences des termes juridiques extraits de notre corpus de référence avec leur nombre d'occurrences et le pourcentage par rapport au nombre d'occurrences totales. Dans ce tableau, sont considérés comme termes juridiques tous les termes qui commencent par un mot pivot. Les mots pivots sont extraits du lexique précédemment cité : un mot pivot est le premier mot d'un terme du lexique.

Nous remarquons que la répartition des séquences diffère de celle du lexique et que la plupart des termes débutent avec la séquence NN : environ 33 % des termes extraits et environ 88 % des termes juridiques qui commencent par un mot pivot. Toutefois, les séquences Adj et Verbe ne peuvent être retrouvées avec notre processus d'extraction.

Classe	#occ	%
Nom Nom	27044	23.48
Nom Nom Nom	10896	9.46
Nom I	10080	8.75
Nom Nom Nom Nom	2553	2.22
Adj Nom	2106	1.83
Nom Adj Nom	1943	1.69
Nom Adj	832	0.72
Nom Nom I	802	0.70
Nom I Nom	727	0.63
Adj Nom Nom	710	0.62

Tableau 5.6 – Extrait de la distribution des termes non juridiques

Le Tableau 5.6 regroupe les principales séquences des termes considérés comme non juridiques sur le principe des mots pivots évoqués ci-dessus. Les deux premières séquences sont identiques à celles des termes juridiques : NN et NNN. Les séquences commençant par

NN sont également majoritaires : environ 36% des termes extraits et environ 71% des termes non juridiques.

Cette approche permet d'éliminer certaines séquences de termes non juridiques. Par exemple, toutes les séquences qui commencent par un adjectif sont éliminées. En effet, ces dernières ne commencent pas par un mot pivot.

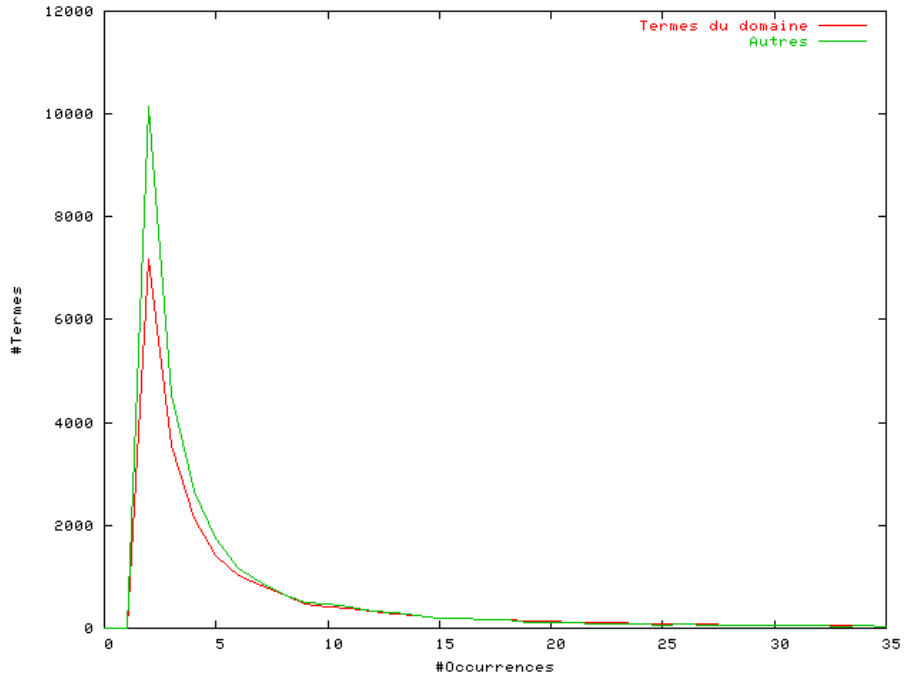

Figure 5.3 – Nombre de termes en fonction du nombre d'occurrences des séquences NN des termes du domaine et non juridiques.

Cependant, ce résultat est peu satisfaisant car la distribution des séquences prédominantes (NN, NNN, etc.) n'est pas discriminante : il existe quasiment autant de termes commençant par la séquence NN dans les deux catégories de termes. Le résultat est, toutefois, prévisible dans le sens où l'on retrouve les séquences les plus représentatives d'un corpus de langue française : ces séquences ne peuvent donc pas être attribuées aux seuls termes du domaine.

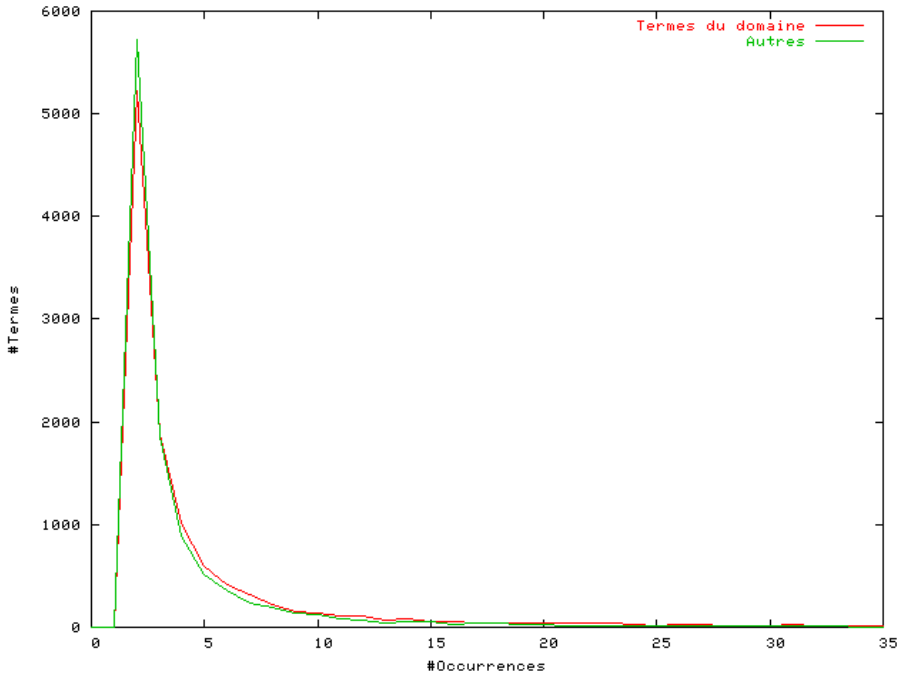

Figure 5.4 – Nombre de termes en fonction du nombre d'occurrences des séquences NNN des termes du domaine et les autres.

Le seul critère des séquences n'est pas discriminant pour déterminer l'ensemble des termes du domaine. Le seul critère statistique, c'est-à-dire avec une fonction de pondération, ne l'est pas non plus [Lame, 2002]. L'objectif est alors de combiner les deux critères afin de déterminer d'éventuelles caractéristiques propres aux termes du domaine.

La Figure 5.3 montre la répartition par fréquence des seules séquences NN des termes du domaine et des termes non juridiques. On remarque que les deux catégories de termes ont des répartitions de fréquences identiques, et qu'on ne peut donc les distinguer à l'aide d'un seuil par exemple. Ce constat se révèle identique pour les séquences NNN, par exemple (*cf.* Figure 5.4).

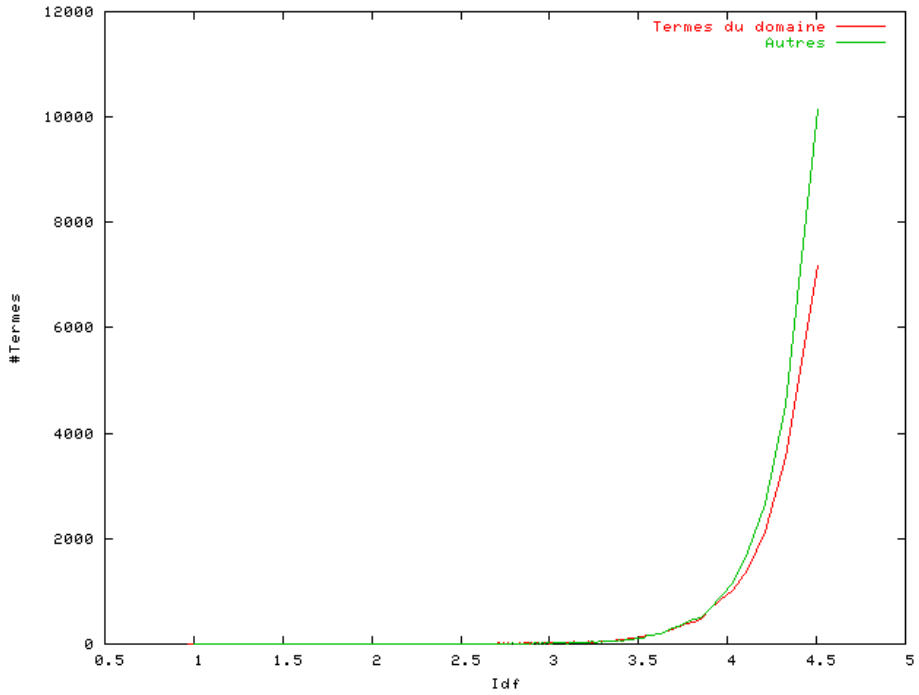

Figure 5.5 – Distribution des termes suivant la valeur de l'Idf des séquences NN pour les termes du domaine et les autres

La répartition des termes suivant une autre fonction de pondération ne permet pas de différencier les termes du domaine des autres (*cf.* Figure 5.5 avec, à titre d'illustration, la fonction de pondération `Idf`).

En conclusion, les termes du domaine sont difficilement identifiables, que ce soit par des méthodes statistiques, des méthodes linguistiques ou des méthodes hybrides.

Cette difficulté à les identifier peut s'expliquer par la nature même du vocabulaire juridique. Les termes sont techniques mais une partie est utilisée dans le vocabulaire commun avec des sens différents ou pas. Ces termes n'ont donc aucune structure grammaticale différente de celle des termes courants (non spécifiques du droit). De plus, la combinaison de cette structure avec une répartition statistique ne permet pas non plus de les identifier.

L'identification des termes techniques spécifiques d'un corpus juridique est beaucoup plus délicate que pour un corpus médical en raison de la fréquence et de la spécificité des termes. En outre, comme vu au § 5.3.2, « Il y a un langage du droit parce que le droit donne un sens particulier à certains termes » [Cornu, 2000].

Dès lors, nous présentons dans le paragraphe suivant, notre méthode d'extraction de termes. Cette méthode ne peut donc prendre en compte que des traitements statistiques et linguistiques de base (ex. une transformation morpho-syntaxique des termes)

5.4 Extraction des syntagmes nominaux

Dans les sections précédentes, nous avons montré que l'extraction des termes du domaine était une tâche délicate. De plus ce domaine ne permet pas d'appliquer certains traitements linguistiques tels que le traitement par synonymie.

Notre méthode d'extraction est de ce fait très limitée, si nous voulons conserver tous les termes du domaine. Toutefois, cette méthode doit permettre d'aboutir à une matrice de distances creuse. Dans cette section, nous présentons l'outil d'extraction terminologique. Le filtrage des termes est présenté dans la section suivante.

Nous utilisons l'extracteur de syntagmes nominaux de l'outil Pertimm qui prend en entrée un document et renvoie, pour ce document, la liste des syntagmes nominaux non lemmatisés. Cet extracteur, nommé *Tok* (référence au mot anglais *Tokenization*), est le « successeur » de Sylex, et plus précisément de l'extracteur intégré dans Sylex (Newpar). Il offre un panel d'options dont quelques unes sont citées ici : tri en sorties des SN par ordre d'apparition dans le document (ou par ordre alphabétique), choix de la langue, types d'informations en sortie (nombre d'occurrences, positions dans le texte), etc. Le fonctionnement est fondé, comme son « prédécesseur », sur un ensemble de dictionnaires représentés par des automates[1] et des règles syntaxiques. Cet ensemble peut être éventuellement configuré, par ajout ou suppression d'entrées dans les différents dictionnaires, pour des corpus spécifiques.

Le Tableau 5.7 représente le résultat de l'application de l'extracteur sur un fichier du corpus de référence représenté par la Figure 5.6.

Usage : Tok *fichier*

> Le domaine public routier comprend l'ensemble des biens du domaine public de l'Etat, des départements et des communes affectés aux besoins de la circulation terrestre, à l'exception des voies ferrées.

Figure 5.6 - Extrait du code de la voirie routière (Partie législative) : article L111-1.

[1] Les automates à états finis utilisés sont ceux décrits dans la littérature

Termes	#occurrence
besoins de la circulation	0001
biens du domaine public	0001
communes affectés aux besoins	0001
communes affectés aux besoins de la circulation	0001
domaine public	0001
domaine public de l'état	0001
domaine public routier	0001
ensemble des biens	0001
exception des voies	0001

Tableau 5.7 – Liste des syntagmes nominaux extraits par Tok.

L'extracteur Tok a été lancé sur le corpus de référence tel que nous l'avons préalablement défini. Une extraction de 250000 termes a été effectuée sur l'ensemble des 64184 articles de codes.

5.5 Filtrage des termes

Dans cette section, nous présentons l'ensemble des traitements statistiques et linguistiques appliqués sur l'ensemble des termes extraits dans l'étape précédente (*cf*. section 5.4).

5.5.1 Lemmatisation

Dans cette section, nous nous intéressons à la réduction des formes fléchies des termes, et plus précisément des syntagmes nominaux.

En sortie de l'extracteur, nous disposons d'une liste de syntagmes nominaux ainsi que d'un ensemble de caractéristiques pour chacun d'entre eux. Pour réduire les formes fléchies des syntagmes nominaux, leur étiquetage morpho-syntaxique attribué par l'extracteur s'avèrerait avantageux. Malheureusement, cette information n'est pas réutilisable car elle n'est pas fournie par l'extracteur.

Ce processus de lemmatisation s'appuie sur des ressources linguistiques : un lexique des formes fléchies et un lexique des mots vides.

Le lexique des formes fléchies regroupe un ensemble de mots français. Des données sont associées à chaque entrée (ou mot) de ce lexique : les flexions ainsi que la catégorie grammaticale pour chaque flexion.

Le lexique des mots vides regroupe l'ensemble des déterminants, articles, pronoms, c'est-à-dire l'ensemble des prépositions.

Pour chaque syntagme nominal, nous testons si les mots le composant appartiennent ou non au lexique des formes fléchies.

5.5.2 Le lexique des formes fléchies

Le lexique des formes fléchies est un fichier texte dont les lignes correspondent aux entrées du lexique. Il contient environ 44000 mots de la langue française courante. Dans nos expériences sur le corpus de référence, seule une partie de ce lexique est utilisée. En effet, le processus de lemmatisation est en aval du processus d'extraction des syntagmes nominaux. Ces derniers ne représentent qu'une partie de la composition syntaxique des documents. Le vocabulaire utilisé par l'ensemble des syntagmes nominaux représente environ 18000 mots.

Epurateur : épurateur, i
Equipement : équipement, n
Equipements : équipement, n
Equipes : équipe, n
Equivalence : équivalence, n
Ergonomie : ergonomie, n
Ergonomiques : ergonomique, a
Ergot : ergot, n
Ergothérapeute : ergothérapeute, n
Ergothérapeutes : ergothérapeute, n

Tableau 5.8 – Extrait du lexique des formes fléchies

Une entrée du lexique est de la forme suivante :

```
<entrée>:<f_canonique>,<catégorie>
```

où `<entrée>` représente le mot que l'on cherche à reconnaître ;
`<f_canonique>` est la forme canonique de l'`<entrée>` ;
`<catégorie>` représente la catégorie syntaxique de la forme canonique.

Dans le cas où un mot serait substituable à différentes formes canoniques, c'est-à-dire à différentes catégories syntaxiques, la forme de l'entrée n'en serait pas moins différente :

```
<entrée>:<f_canonique1>,<catégorie1>:
         <f_canonique2>,<catégorie2>:…
```

Les catégories syntaxiques sont notées dans le lexique en suivant la convention suivante [Constant, 1995] :

v : verbe	(22000 occurrences pour Tok incluant les formes fléchies)
n : nom	(8000 occurrences)
np : nom propre	(530 occurrences)
mc : mot composé	(870 occurrences)
a : adjectif	(12800 occurrences)
b : adverbe	(800 occurrences)
i : inconnu	
p : préposition	(140 occurrences)

d : déterminant (100 occurrences)
c : conjonction (70 occurrences)

Le lexique a été mis à jour spécifiquement pour le corpus de référence. Après la phase d'extraction des syntagmes nominaux, nous réduisons ces derniers en testant chaque mot les composant avec le lexique. Nous avons énuméré les mots n'existant pas dans le lexique. Cette liste contient environ 1000 mots. Les raisons de l'absence de ces mots dans le lexique sont variées. Ces différentes raisons sont résumées ci-dessous :

- Le mot est technique.
  Exemple : cérébro-spinale, aéroportuaire
- Le mot est un acronyme.
  Exemple : ADN, Afnor
- Le mot n'existe pas dû à une erreur dans le document :
  - Il est coupé dans le document par des espaces.
  - Il n'est pas orthographié correctement.
    Exemple : nommmé

Cet aspect des mots inconnus dans un lexique nous oblige à nous interroger sur l'importance de ces mots dans les différentes étapes de notre processus :

- Ces mots sont-ils des termes du domaine ?
- Ces mots ont-ils une importance dans l'étape de classification ; permettent-ils de connecter des documents qui ne l'étaient pas ?

En partant du principe que tous les termes du domaine doivent être conservés, et que le vocabulaire du droit évolue au fil du temps, il est donc impossible de statuer, à long terme, sur ces mots inconnus. Les mots inconnus sont donc conservés malgré la possibilité qu'ils soient des termes non spécifiques.

5.5.3 Les termes vides

La dernière étape de la réduction des termes est l'élimination des termes vides. La notion de terme vide est énoncée dans le Chapitre 2. En résumé, elle peut être vue comme une liste prédéfinie de déterminants, articles, etc. Elle peut être également vue comme une liste prédéfinie des mots trop fréquents du corpus.

Dans un premier temps, nous éliminons des termes tous les articles, déterminants, prépositions, etc.

Dans un second temps, nous éliminons les termes trop fréquents et les hapax. L'élimination des hapax se justifie par le fait qu'ils n'ont aucune incidence sur le résultat d'une méthode de classification. En effet, l'hapax est, par définition, présent dans un seul document et ne peut donc faire l'objet d'un regroupement entre deux documents. Toutefois, l'hapax a une importance négligeable dans le calcul d'une distance. L'élimination des termes trop fréquents va permettre de construire une matrice creuse. Nous définissons comme terme

trop fréquent tout terme t dont $\text{Idf}(t) \geq \alpha |C|$ avec $\alpha = 0.1$. Cette valeur empirique est fréquemment utilisée dans la littérature.

Ce filtrage par valeur de l'Idf d'un terme permet d'éliminer près de la moitié des termes. Sur un total de 343000 termes, nous disposons d'environ 115000 termes après filtrage (dont 8000 sont éliminés par lemmatisation). Parmi ces quelque 220000 termes éliminés, la plupart sont éliminés en tant qu'hapax. Seule une vingtaine de termes sont éliminés en tant que termes trop fréquents, dont les termes « parti législatif », « décret conseil », « conseil état ».

Bien que l'objectif principal ait été d'adapter les méthodes traditionnelles d'identification de termes au domaine juridique, nous constatons qu'il n'en peut être ainsi. Le vocabulaire juridique n'a pas de caractéristique spécifique. En contrepartie, nous avons montré que notre méthode d'extraction est générique, et peut s'adapter à tout corpus.

5.6 Conclusion

Les méthodes d'extraction de termes s'axent sur des approches statistiques, linguistiques et hybrides. Nous avons montré que ces trois méthodes ne permettent pas d'isoler le vocabulaire juridique du vocabulaire courant pour un corpus du domaine. Ce constat converge avec l'idée qu'une partie du vocabulaire juridique s'apparente au vocabulaire courant dans la forme, toutefois avec des différences au niveau de la définition ou du concept. Ainsi, nous pouvons nous trouver face à des problèmes d'homonymie dans l'interface homme-machine. Ces expériences sur le corpus mettent en évidence deux idées.

La première est que le vocabulaire est précis et technique, et certains termes ne nécessitent aucun contexte pour en cerner le sens, tout du moins pour l'initié. Ce qui n'est pas toujours le cas pour le novice. De plus, ce vocabulaire, par ces caractéristiques, ne peut subir n'importe quel traitement linguistique sans l'avis d'experts, par exemple, les traitements synonymiques ou les termes associés. L'extraction de termes et les méthodes de filtrage se limitent donc, dans notre cas, à des méthodes classiques.

Si ce vocabulaire ne s'adapte pas à tout traitement linguistique, il permet toutefois de représenter un concept par un seul terme, et par extrapolation une thématique par un seul terme. L'idée de représenter une classe par un seul terme est donc envisageable dans ce cas (nous ne faisons pas référence au centre d'une classe). Par conséquent, la présentation d'une liste de termes et non pas d'une liste d'ensemble de mots est cohérente avec ce corpus.

La seconde idée concerne la distribution des termes du domaine. Nous remarquons que les termes juridiques ont des propriétés statistiques faibles, et la distinction avec le vocabulaire courant est difficile. Dans le chapitre précédent, l'hypothèse 1 soutient l'idée qu'à toute thématique correspond une distribution singulière des termes. Cette hypothèse reste valable, avec toutefois une orientation différente : on supposera que le vocabulaire d'une thématique sera singulier de la thématique, c'est-à-dire que le vocabulaire associé y est plus

fréquent dans cette thématique que dans les autres. Cependant, rien ne montre que ce vocabulaire sera fortement présent au sein de chaque document d'une classe.

Chapitre 6

Ω-means : un algorithme de classification globale non-supervisée

Résumé

Dans ce chapitre, nous présentons une nouvelle méthode de classification. De type partitionnement, elle est inspirée, plus précisément, de l'algorithme K-means. Utilisant une matrice de similarité creuse, elle permet de classer un grand nombre de documents sur un grand nombre de critères.

La plupart des algorithmes de type partitionnement ont l'inconvénient majeur de ne pouvoir déterminer un nombre initial de classes. Dans ce chapitre, nous proposons une méthode pour déterminer automatiquement la valeur de K. Nous supposons, toutefois, que cette méthode ne semble pouvoir s'appliquer que sur des matrices de similarité creuses.

6.1 Introduction

Un système de recherche documentaire donne, en réponse à une requête donnée, une liste de documents. Les documents ainsi proposés sont généralement ordonnés suivant une valeur de pertinence calculée automatiquement. Cette liste est souvent longue et permet rarement à l'utilisateur de la parcourir dans sa totalité afin de trouver d'éventuels documents pertinents mal positionnés. L'une des alternatives à cette liste de documents est de classer les documents trouvés et de présenter les résultats [Hearst & Pedersen, 1996]. Toutefois, il est possible de garder une structure de type liste de documents en parcourant les classes et en réordonnant les documents pour constituer une nouvelle liste. Une présentation structurée des résultats permet d'améliorer la qualité de la recherche, car l'utilisateur choisit les classes qui contiennent le plus de documents pertinents [Evans et al., 1999] ou bien le plus de mots pertinents si les classes se présentent sous forme de liste de mots. Néanmoins, cette approche nécessite de la part de l'utilisateur des efforts supplémentaires par rapport à un système classique. En effet, celui-ci doit, dans un premier temps, parcourir toutes les classes et, dans un second, choisir les classes les plus pertinentes. Il est possible de faire intervenir l'utilisateur avec un effort moindre, en lui proposant un ensemble d'informations qu'il pourra parcourir rapidement, où il choisira l'information pertinente tout aussi rapidement.

Dans ce chapitre, nous proposons dans un premier temps un algorithme de classification non-supervisée naïf et, dans un second temps, sa version définitive. Puis, nous évaluons ces deux algorithmes sur notre corpus de référence, détaillé dans le Chapitre 4. L'algorithme Ω-means est évalué dans le chapitre suivant.

Pour évaluer les différentes partitions trouvées, nous utilisons globalement les critères usuels de la littérature, à savoir le taux de précision, le taux de rappel, etc. Pour évaluer les étiquettes trouvées, il existe peu de critères dans la littérature. De plus, ces derniers peuvent donner une vision erronée suivant la composition du corpus. Nous proposons donc de nouveaux critères, tout en utilisant ceux proposés dans la littérature.

6.2 Paramètres

Dans cette section, nous présentons les différents paramètres de l'algorithme, à savoir la fonction de pondération des termes, la mesure de ressemblance utilisée et la représentation des classes adoptée. Les pré-traitements relatifs à l'abstraction des documents tels que l'extraction des termes, le filtrage des termes, etc., ont été décrits dans le Chapitre 5 et ne seront donc pas abordés dans ce chapitre.

6.2.1 Pondération des termes

La pondération utilisée dans notre méthode est classique dans le domaine de la recherche d'informations puisqu'il s'agit de la fonction `Tf.Idf`, dont l'une des formes a été présentée dans la section 2.6 du Chapitre 2.

Pour nos expérimentations, nous utilisons différentes formes de la fonction `Tf.Idf`, c'est-à-dire avec des normalisations différentes.

6.2.2 Mesure de ressemblance

Les mesures de ressemblance utilisées dans les méthodes de classification sont variées : distances, similarités, dissimilarités ou modèles probabilistes. Chaque mesure vérifie une ou plusieurs conditions [Michelet, 1988]. Certaines méthodes de classification, telles que la méthode des nuées dynamiques par exemple, requièrent l'utilisation d'une distance pour assurer la convergence. En effet, la convergence nécessite une mesure d'association vérifiant l'inégalité triangulaire, que seules les distances respectent. Dans le cas où la mesure d'association n'est pas une distance, la convergence n'est pas assurée. Toutefois, des travaux [Bellot, 2000] montrent que la classification devient relativement stable au bout d'une dizaine d'itérations[1]. Cela permet de stopper le processus de classification arbitrairement, sans être pénalisé en ce qui concerne la partition finale.

Dans le cadre de notre méthode de classification, la mesure d'association n'est pas une distance : nous avons choisi l'indice de similarité `Cosine` (*cf.* Chapitre 2). Bien qu'il ne s'agisse pas d'une distance au sens strict des mathématiques, ce choix n'a pas d'incidence pratique sur la convergence de l'algorithme. Des expérimentations ont malgré tout été menées avec d'autres mesures : la distance euclidienne, par exemple (voir § 7.7). Ces expérimentations tentent de montrer que le choix de la mesure d'association n'a qu'une influence modérée sur les résultats de la classification.

La plupart des méthodes de partitionnement nécessitent une valeur prédéfinie du nombre de classes. De même, pour les systèmes de recherche, un nombre de classes identiques est utilisé pour chaque requête [Hearst & Pedersen, 1996], [Silverstein & Pedersen, 1997]. Ce nombre est défini empiriquement. Dans le § 6.5.1, nous proposons une méthode de détection automatique du nombre de classes, fondée essentiellement sur les liens entre documents et ne faisant pas intervenir la notion de mesure de ressemblance. Nous verrons, dans la section 7.4 du Chapitre 7, la relation entre le nombre de classes et la quantité de liens produits entre les documents. Cette méthode peut s'avérer une alternative, de façon hypothétique, au nombre de liens empiriques, si nous utilisons notre algorithme en mode système de recherche dynamique.

[1] La partition trouvée évolue peu d'une itération à l'autre.

6.2.3 Représentation des classes

Les classes sont généralement représentées par un centroïde ou par un médoïde (ces deux modes de représentation ont été définis dans le § 3.5.2 du Chapitre 3). Pour résumer, l'avantage du médoïde est qu'il permet de retrouver des classes de *forme quelconque* (voir Figure 6.1 extraite de [Govaert, 2003])[1]. La représentation d'une classe par plusieurs individus permet, en effet, de se rapprocher bien plus de la forme de la classe qu'un centroïde. Quant à ce dernier, l'avantage est qu'il limite les effets des attributs *perturbateurs*.

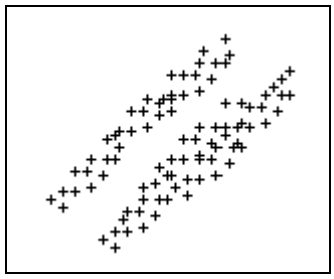

Figure 6.1 - Exemple de formes de classes

Nous suivons l'approche de [Diday et al., 1982], [Bellot, 2000] qui consiste à prendre un médoïde (ou axe médian) afin d'assurer la convergence plus rapidement. Le noyau est généralement composé de trois individus. Nous verrons dans nos expériences l'influence de ce nombre empirique sur le corpus de référence.

6.3 Evaluation

Dans cette section, nous abordons les notions d'évaluation à la fois des partitions trouvées et des thématiques.

6.3.1 Evaluation des partitions

Dans la section 2.8 du Chapitre 2, nous avons présenté les critères classiques pour évaluer les performances d'un système de recherche d'information : précision, rappel, précision/rappel, etc.

Ces critères permettent d'évaluer les performances d'un système à retrouver les documents pertinents pour une requête donnée. Ils peuvent toutefois s'adapter pour évaluer une partition trouvée. Ainsi, la précision est définie de la façon suivante :

Soit $P = \{P_1,...,P_i,...,P_n\}$ une partition finale de *n* classes,

$Q = \{Q_1,...,Q_i,...,Q_n\}$ une partition de référence de *n* classes,

[1] Le choix de la distance joue un rôle important dans le cas d'une représentation par centroïde.

$DPR(P_i, Q_i)$ le nombre de documents présents dans la classe P_i et pertinents pour la classe Q_i correspondante,

$DPNR(P_i, Q_i)$ le nombre de documents présents dans la classe P_i et non pertinents pour la classe Q_i correspondante.

$$\begin{aligned} \text{Pr}\,écision &= \frac{1}{|P|}\sum_{i=1}^{|P|}\frac{DPR(P_i,Q_i)}{DPR(P_i,Q_i)+DPNR(P_i,Q_i)} \\ &= \frac{1}{|P|}\sum_{i=1}^{|P|}\frac{DPR(P_i,Q_i)}{|P_i|} \end{aligned} \tag{6.1}$$

Le rappel est défini comme suit :

$$Rappel = \frac{1}{|P|}\sum_{i=1}^{|P|}\frac{DPR(P_i,Q_i)}{|Q_i|} \tag{6.2}$$

Dans cette notation, on suppose qu'à la classe P_i correspond la classe Q_i de référence. Ainsi, à chaque classe trouvée, on associe une classe de référence pour laquelle on retrouve le plus grand nombre de documents.

Ces critères sont intéressants pour comparer deux partitions ayant un cardinal identique. Dans le cas où $\text{card}(P) \neq \text{card}(Q)$, il est délicat de prendre comme classe de référence, pour une classe trouvée, celle qui regroupe le plus grand nombre de documents présents dans la classe trouvée. En effet, suivant la différence entre le nombre de classes de référence et celui déterminé automatiquement, les classes de référence seront divisées ou regroupées dans une ou plusieurs classes. Ainsi, l'association d'une classe trouvée à une classe de référence devient une tâche difficile. Dans le cas où $\text{card}(P) > \text{card}(Q)$, il est facilement imaginable que deux classes de référence (voire plus) soient regroupées au sein d'une même classe. Et inversement, une classe de référence peut être majoritaire dans plusieurs classes trouvées même si $\text{card}(P) < \text{card}(Q)$.
Ainsi, nous devons adapter ces deux indices pour les cas cités ci-dessus. Dans le cas du taux de rappel, l'indice est alors défini de la façon suivante :

$$PQ = \frac{1}{|P|}\sum_{i=1}^{|P|}\frac{DPR(P_i, rep(P_i))}{|rep(P_i)|} \tag{6.3}$$

où $rep(P_i)$ est la fonction qui permet de retrouver la classe Q_i correspondant à P_i.

D'autres critères ont été définis pour évaluer les systèmes de recherche tels que le taux de corrélation [Voorhees & Harman, 1999] ou le test de Wilcoxon-Mann-Whitney [Saporta, 1990]. Ces critères permettent de déterminer, par exemple, la quantité de documents

pertinents parmi les *n* documents proposés. Ils permettent également de comparer deux échantillons de réponses, etc.

6.3.2 Evaluation des thématiques

L'évaluation d'un système de recherche est principalement fondée sur les capacités qu'il possède à retrouver les documents pertinents, ce qui constitue le principal objectif. Dans notre cas, nous devons non seulement évaluer les partitions trouvées en terme de précision, rappel, etc. (voir section ci-dessus), mais aussi évaluer les thématiques (étiquettes) des classes de la partition. En effet ces thématiques sont utilisées par la suite dans notre système de recherche.

Pour évaluer la performance d'un système à retrouver les thématiques, nous pouvons nous inspirer du concept de la métrique (AC) définie par [Xu et al., 2002] qui évalue pour chaque document d_i du corpus C la correspondance entre la thématique trouvée par la méthode et celle définie dans C. Pour un document d_i donné, l_i et α_i sont respectivement la thématique trouvée pour le document et la thématique définie dans le corpus (C). Cette métrique est définie comme suit :

$$AC = \frac{\sum_{i=1}^{|C|} \delta(\alpha_i, map(l_i))}{|C|} \tag{6.4}$$

où $\delta(x, y)$ est la fonction delta qui vaut 1 si $x = y$ et vaut 0 dans le cas contraire, et $map(l_i)$ est la fonction de correspondance qui permet d'associer la thématique l_i avec celle correspondante dans (C).

Cette métrique est essentiellement adaptée pour des corpus composés de classes de taille homogène. Pour les corpus dont les classes sont de tailles variées, cette mesure n'est pas adéquate car elle donne une valeur globale des documents correctement étiquetés. En d'autres termes, les classes prépondérantes dans un corpus et retrouvées en grande partie dans la partition finale dissimuleront les lacunes à retrouver les thématiques des plus petites classes. La métrique (LAC) que nous définissons ci-après, donne une vue globale par rapport aux étiquettes des classes. Ainsi, nous avons une mesure précise sur la quantité d'étiquettes retrouvées et ceci indépendamment de la taille des classes. Par contre, nous ne pouvons pas quantifier le nombre de documents trouvés pour chaque étiquette. La métrique (LAC) est définie comme suit :

$$LAC = \frac{\sum_{i=1}^{|P|} \delta\left(l(P_i), l(Q_i)\right)}{|P|} \tag{6.5}$$

où $\delta(x, y)$ est la fonction delta définie précédemment et $l(P_i)$ la thématique de la classe P_i.

Les mesures de AC et de LAC permettent ainsi de mesurer de façon adéquate la capacité à retrouver les étiquettes des classes. A noter que l'on peut combiner ces deux métriques comme suit :

$$ACC = \frac{\sum_{i=1}^{|P|} \left(\frac{1}{|P_i|} \sum_{j \in P_i} \delta\left(\alpha_j, map(l_j)\right) \right)}{|P|} \tag{6.6}$$

où les fonctions $\delta(x, y)$ et $map(l_i)$ sont les mêmes que décrites précédemment.

Cependant, dans nos expériences, nous utiliserons principalement les critères AC et LAC car notre objectif principal est de retrouver les thématiques et, par ailleurs de déterminer s'il sera possible de les retrouver au niveau inférieur ; le critère ACC nous donne juste une information globale.

6.4 Algorithme naïf

Dans cette section, nous décrivons un algorithme naïf de type K-Means pour lequel seules les fonctions inhérentes à cette catégorie d'algorithmes sont appliquées. Les objectifs de l'algorithme sont fortement liés à la capacité de celui-ci à classer un grand nombre de documents en un temps *raisonnable*, c'est-à-dire de converger le plus rapidement possible (en quelques itérations). Les expérimentations menées sur notre corpus de référence montrent que ces objectifs sont atteignables avec une qualité de résultats encourageante.

6.4.1 Approche de la méthode

L'approche classique des méthodes de type K-means est d'utiliser les distances entre documents, d'une part, pour agréger chaque document à un centre suivant un critère défini et d'autre part, pour recentrer chaque classe avec l'élément (centroïde ou médoïde) qui se rapproche le plus du centre de la classe. Le critère d'agrégation est généralement d'associer le document au centre le plus proche en terme de distance.

Nous avons montré que les classes peuvent avoir une forme quelconque et que l'approche de la distance minimum entre individus peut s'avérer insuffisante pour certaines données (voir Figure 6.2).

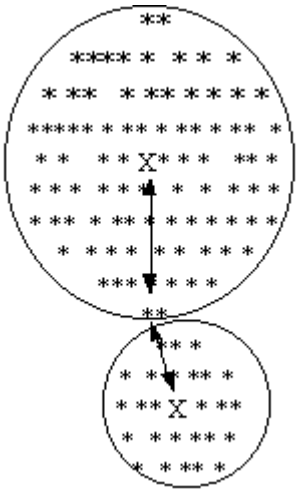

Figure 6.2 – Problème de la distance minimum : exemple sur deux classes

Notre approche est d'utiliser à la fois les distances entre documents et les liens entre documents. Dans la littérature, les liens se rapportent habituellement aux liens hypertextes des pages Web ; ils sont généralement mis en place par les concepteurs des sites Web, et sont utilisés dans différentes méthodes de classification dans le but de déterminer des communautés (ensemble de documents partageant des liens sémantiques -*cf*. Chapitre 3-) [Gibson et al., 1998]. Dans notre méthode, la notion de lien est simplement liée au fait que deux documents partagent ou non des informations communes, c'est-à-dire des termes.

Le lien entre deux documents est ainsi affranchi de toute distance ; on s'intéresse uniquement à un sous-ensemble des informations, l'ensemble des informations partagées par les deux documents.

6.4.2 Condition initiale

L'algorithme est une itération successive de deux fonctions décrites ultérieurement : une fonction d'allocation des documents et une de recentrage des classes. La fonction de recentrage repose sur une caractéristique forte de la matrice. En effet, cette dernière doit être suffisamment creuse. La convergence de l'algorithme naïf est liée à cette caractéristique de la matrice.

La construction d'une matrice de distances résulte directement de la phase d'extraction de termes. Pour « s'assurer » d'obtenir une matrice creuse, nous avons extrait uniquement les syntagmes nominaux des documents (*cf*. Chapitre 5).

6.4.3 Description de l'algorithme

Dans cette section, nous décrivons notre algorithme dans son ensemble.

- *K* est donné ;
- choix des *K* centres : prendre les *K* documents ayant le plus grand nombre de liens ;
- faire :
    1. affecter chaque document au centre le plus proche ;
    2. calculer le nouveau centre de chaque classe ;

  tant que des documents migrent d'une classe vers une autre.

Algorithme 6.1 – Algorithme naïf

En considérant notre ensemble de documents *D* comme un graphe $G(V,E)$ où *V* est l'ensemble des nœuds (*i.e.* l'ensemble *D*) et *E* l'ensemble des arcs[1] ; le but de notre méthode de classification est de détecter les sous-ensembles fortement connectés. Ces sous-ensembles fortement connectés sont ainsi considérés comme des classes homogènes.

6.4.3.1 Choix des centres

Cette phase permet de déterminer l'ensemble des *K* centres initiaux pour une valeur de *K* donnée. La valeur de *K* n'est pas déterminée automatiquement dans l'algorithme naïf. La phase d'initialisation est une étape importante pour les algorithmes de partitionnement car elle conditionne la qualité de la partition finale : les résultats s'avèrent différents selon le choix de la partition initiale (ou le choix des centres initiaux). Notre algorithme de partitionnement est également dépendant du choix des centres initiaux dans la qualité de la partition finale.

En dehors de toute considération sur le choix du nombre de classes, la partition initiale est déterminée de différentes façons dans la littérature. La méthode aléatoire a été l'une des premières méthodes utilisées dans la phase d'initialisation. Cependant, elle ne peut être satisfaisante puisque le résultat de la partition finale est, dans ce cas, difficilement interprétable. Certains auteurs ont substitué des méthodes de classification classiques telles que celle de Forgy [Forgy, 65] comme moyen d'initialisation de leur méthode. Cette approche est coûteuse en temps de calcul puisqu'il faut, au final, appliquer deux méthodes de classification.

Le choix d'une méthode d'initialisation se résume donc à opter pour la méthode aléatoire, pour une méthode de classification classique ou pour une nouvelle méthode en adéquation avec l'algorithme de classification.

Le choix des centres est fortement lié à la fonction de recentrage décrite ultérieurement dans ce sous-paragraphe. Le principe de cette fonction est que chaque centre, à une itération *i*, possède un nombre de liens inférieur (ou égal) à celui du centre lui correspondant à l'itération $i-1$. Les centres ayant de moins en moins de liens à chaque itération, le risque est de tomber

[1] Soit V_1 et V_2 deux nœuds (correspondant à D_1 et D_2) ; si $D_1 \cap D_2 \neq \varnothing$ alors il existe un arc non pondéré entre les deux nœuds.

rapidement dans des minimas locaux indésirables, si les centres sont choisis aléatoirement par exemple. Les conséquences sont multiples :

- construction de petites classes uniquement ;
- convergence prématurée de l'algorithme ;
- nombre important de documents non-classés.

On suppose que ce phénomène peut être nuancé par un choix de documents (et donc de centres) ayant le plus grand nombre de liens.

Pour une valeur de *K* donnée, les *K* centres initiaux correspondent aux éléments d_I ayant le plus grand nombre de liens (en considérant le corpus comme un graphe) tel que :

$$I = \arg\max_{i \in N} \mathrm{N}_C(d_i) \tag{6.7}$$

La sélection des centres initiaux se fait en $O(KN)$.

6.4.3.2 Affectation des éléments

Les *K* centres étant choisis, l'étape suivante consiste à affecter chaque élément *c* de *C* à l'un des centres selon un critère. Soit $\{O_i\}$ l'ensemble des *K* centres, avec $i \in [1, K]$. Le critère le plus élémentaire est d'affecter chaque élément au centre qui lui est le plus proche en terme de distance (ou de mesure de similarité dans notre cas).

L'élément *c* est affecté à O_i si :

$$i = \arg\min_{i \in K} \cos(c, O_i) \tag{6.8}$$

Le coût en temps de calcul de cette phase est proportionnel à *K*|*C*|.

6.4.3.3 Recentrage des classes

La fonction de recentrage permet de déterminer les éléments les plus représentatifs pour chaque classe. Cette notion de représentation de classe diffère suivant le type de méthode de classification. Pour les modèles fondés sur le contenu des documents (par exemple k-means ou EM), le centre est généralement l'élément qui est le plus proche de tous les éléments de la classe (en terme de distance ou en mesure de similarité). Pour les modèles fondés sur les liens hypertextes, le centre est l'élément qui possède le plus grand nombre de liens avec les éléments de la classe.

Le but de notre méthode de recentrage est de détecter les sous-ensembles fortement connexes du point de vue d'un graphe. Pour retrouver ces sous-ensembles à partir d'un nœud unique pour chacun d'entre eux, nous choisissons le nœud qui possède le plus grand nombre

de liens dans le sous-ensemble (pour retrouver le maximum de nœuds) et un minimum de liens avec les autres nœuds (pour éviter de rattacher des nœuds indésirables).

Cette fonction est fondée sur l'hypothèse 2 du Chapitre 4 à propos d'un vocabulaire spécifique à chaque classe. Compte tenu de ce vocabulaire spécifique partagé par tous les éléments d'une classe, ces éléments seront, dans le cas idéal, tous connectés entre eux. Ce vocabulaire spécifique sera peu présent dans les autres éléments. Il y aura ainsi peu de liens avec les autres éléments et par conséquent avec les autres classes. Néanmoins, les éléments n'étant pas uniquement représentés par du vocabulaire spécifique, ces classes seront malgré tout connectées entre elles.

La sélection du nouveau centre pour chaque classe est fondée sur le critère suivant :
Soit $\Pi_i = \{\Pi_{i1},..., \Pi_{iK}\}$ la partition retrouvée des *K* classes à l'itération *i*,
Soit $\Pi_{ij} = \{c_{j1},..., c_{jm}\}$ l'ensemble des éléments de la classe *j* avec $m = |\Pi_{ij}|$
Le nouveau centre de la classe Π_{ij} est C_j ($j \in [1, m]$), si :

$$C_j = \arg\max_{c_k} \frac{N_{\Pi_{ij}}(c_{jk})}{N_C(c_{jk})} \tag{6.9}$$

Convergence

La fonction de recentrage permet une convergence rapide de l'algorithme. Cette convergence est d'autant plus rapide qu'il n'y a pas de fonction de fusion.

6.4.3.4 Expérimentations

Dans cette section, nous évaluons notre algorithme naïf uniquement à travers les paramètres définis par défaut en début de chapitre. En effet, l'objectif est de montrer la légitimité de l'approche et d'en définir les avantages et les inconvénients. Toutefois, les changements de paramètres pour quelques expérimentations sont indiqués.

Cette évaluation s'effectue sur le corpus de référence défini dans le Chapitre 4. Le nombre de classes *K* est prédéfini et vaut 57 (voir p. 88).

Dans le Tableau 6.1, nous présentons une partie des classes de la partition finale ainsi générée. En effet, les classes doublons, c'est-à-dire les classes représentatives d'un code identifié préalablement dans une autre classe, ne sont pas présentées dans ce tableau : elles présentent peu d'intérêt à travers les critères d'évaluation proposés dans ce tableau.
On remarque que plusieurs cas se présentent quant à la constitution des classes. Certains codes sont retrouvés entièrement au sein d'une classe mais sont englobés dans une multitude de sous-ensembles de codes. Par conséquent, le taux de rappel est élevé mais le taux de précision est faible. Les résultats montrent la faible capacité de l'algorithme à retrouver les codes de grandes tailles : taux de précision généralement élevé mais un taux de rappel faible. Ainsi, une classe représente essentiellement une partie d'un code. Les classes doublons sont

principalement représentatives de l'éclatement de ces codes de grande taille. Toutefois, les taux de rappel les plus faibles correspondent à des codes de taille moyenne.

Codes	|codes|	Précision	Rappel
CACTSOC	577	0.47	0.79
CARTISA	45	0.46	0.98
CASSURA	1333	0.92	0.70
CCIVILL	2605	0.85	0.98
CCONSTR	2295	0.94	0.71
CDMEDIC	114	0.62	0.81
CDYANES	477	0.87	0.65
CEDUCAT	757	0.96	0.61
CELECTO	857	0.79	0.83
CENVIRO	971	0.86	0.80
CEXPROP	237	0.27	**1.00**
CFOREST	1149	0.91	0.77
CGCTERR	3586	0.70	0.18
CGIMP	4901	0.97	0.47
CLEGHON	176	0.89	**1.00**
CMONFIN	1299	0.94	0.76
CMUTUAL	205	0.42	0.37
CORGJUD	951	0.93	0.93
CPENALL	964	0.18	0.04
CPENSIC	275	0.57	0.62
CPENSIM	1861	0.95	0.96
CPOSTES	708	0.93	0.71
CPROCIV	1554	0.90	0.96
CPROCPE	2404	0.22	0.10
CRURAL	4911	0.44	0.07
CSANPU	4720	0.82	0.10
CSECSOC	6500	0.96	0.41
CTRAVAI	4845	0.89	0.62
CURBANI	1493	0.90	0.72
CVOIRIE	244	0.77	0.86

Tableau 6.1 – Algorithme naïf avec les paramètres par défaut : taux de rappel et précision pour les classes de la partition finale (les classes doublons[1] ne sont pas représentées).

Le Tableau 6.2 résume les expérimentations menées sur l'utilisation de l'algorithme pour en déterminer non seulement l'influence sur les résultats mais également l'intérêt ou pas du critère d'initialisation choisi (*cf.* § 6.4.3.1). Pour déterminer la pertinence du critère d'initialisation des centres, nous l'avons simplement comparé avec un critère basique qui est le choix aléatoire des centres. Les expérimentations montrent que le choix aléatoire des centres donne globalement de meilleurs résultats que le critère par défaut. Toutefois, le taux de AC est plus important pour le critère par défaut, ce qui signifie qu'un plus grand nombre de documents est étiqueté correctement. Le taux de LAC montre que l'on retrouve quasiment autant de labels corrects quel que soit le critère choisi.
L'algorithme a l'avantage de converger rapidement grâce à la fonction de recentrage qui permet à chaque itération de déterminer des nouveaux centres c_i avec un nombre de liens $\mathrm{N_C}(c_i)$ relatifs de plus en plus faible.

[1] Une classe est dite doublon si elle est représentée par un code déjà représentatif d'une autre classe.

Initialisation	#codes_atteints	AC	LAC	PQ	#ité
Défaut	30	**0.40**	0.42	0.37	**4**
Rand	**40**	0.37	**0.43**	**0.49**	**4**

Tableau 6.2 – Algorithme naïf pour des initialisations différentes : caractéristiques de la partition finale

La Figure 6.3 montre l'évolution des différents indices AC, LAC et PQ à travers les différentes itérations. Nous constatons que ces indices augmentent de façon significative entre la première et la deuxième itération. Entre la troisième et la quatrième itération, ces indices évoluent très peu montrant ainsi de faibles migrations des documents d'une classe à une autre entre ces deux itérations.

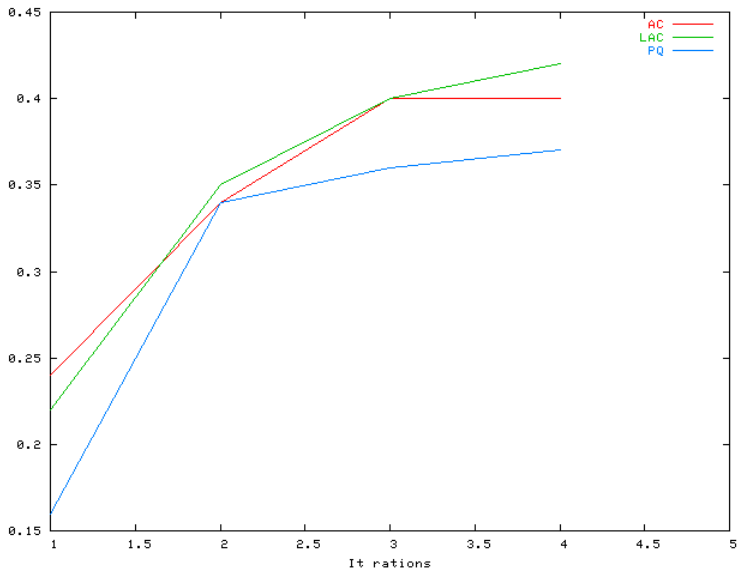

Figure 6.3 – Algorithme naïf : valeurs des indices AC, LAC et PQ pour les différentes itérations

Notre algorithme permet la non affectation de documents grâce à leur versement dans la classe résidu. Cette classe résidu devrait être quasiment vide, pour notre corpus de référence, à chaque fin de processus. Toutefois, à travers nos expérimentations, nous remarquons que celle-ci est composée d'un ensemble de documents de taille non négligeable. Elle comprend 15488 documents en fin de processus, en prenant pour notre algorithme les paramètres par défaut. Elle comprend 15787 documents avec l'initialisation aléatoire des centres. Cette classe représente donc, pour ces expérimentations, près d'un quart du corpus de référence, loin de la valeur escomptée. Nous supposons que ce comportement de l'algorithme est la conséquence d'un manque de fusion des classes proches les unes des autres qui permettrait d'éliminer les classes doublons tout en choisissant un centre proche de la classe ainsi créée en terme de liens. En effet, ce manque de fusion peut engendrer facilement des trous locaux de façon irréversible.

Termes	#occurrences
livre ier	2920
code santé	2560
code rural	2307
code général collectivité	1962
chapitre iii	1893
code procédure pénal	1548
code pénal	1548
code commerce	1399

Tableau 6.3 – Les principaux termes représentant la classe résidu

Dans le Tableau 6.3, nous avons énuméré les termes les plus présents dans la classe résidu et nous pouvons constater que la plupart de ces termes constituent des étiquettes représentant les principaux codes présents dans la classe résidu et énumérés dans le Tableau 6.4.

Codes	#Eléments	Précision (P)	Rappel (R)
CSANP	2501	0.16	0.53
CRURA	2260	0.15	0.46
CGCTERR	1935	0.12	0.54
CPROCPE	1548	0.10	0.64
CCOMMER	1391	0.09	0.73
CPENALL	751	0.05	0.78
CPROINT	709	0.05	0.67
CJURFIN	635	0.04	0.66

Tableau 6.4 – Les principaux codes de la classe résidu

Le taux de rappel moyen des codes présents dans la classe résidu (*cf.* Tableau 6.4) est de 0.63. Ce taux élevé est d'autant plus inquiétant qu'il concerne des codes de taille moyenne qui, pour la plupart, sont retrouvés malgré tout dans la partition finale. Ainsi, la classe résidu contient hypothétiquement des centres plus intéressants, en termes de liens, que ceux choisis dans la classe correspondante. Cette classe résidu doit donc être intégrée dans la fonction de recentrage comme vivier de nouveaux centres éventuels.

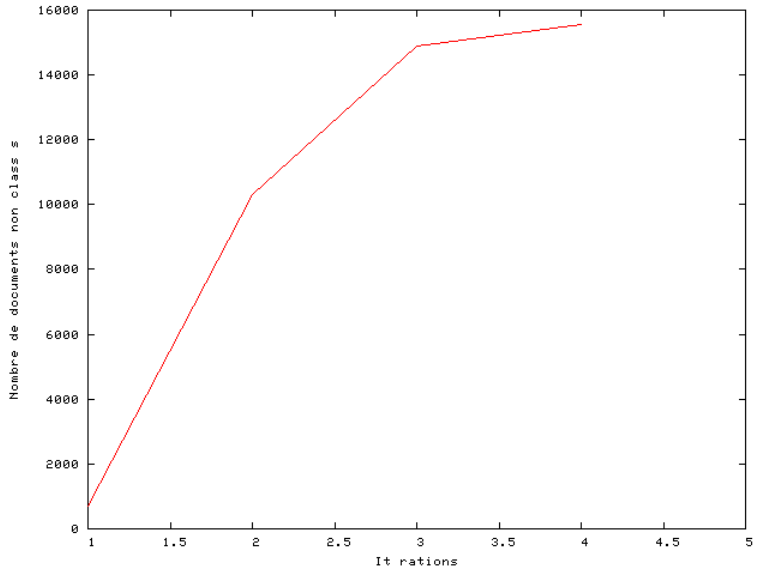

Figure 6.4 – Nombre de documents non classés en fonction des itérations

La Figure 6.4 représente l'évolution du cardinal de la classe résidu au fur et à mesure des itérations. Il est intéressant de constater que la première itération ne permet pas de classer tous les documents : environ 1.5 % des documents ne sont pas classés. Le cardinal de la classe augmente considérablement à la deuxième itération. Ce phénomène peut s'expliquer simplement par le fait suivant : pour la première itération, les centres choisis sont ceux ayant le plus grand nombre de liens ; pour les itérations suivantes, les centres sont choisis entre autres suivant leur faible nombre de liens avec l'extérieur. Une trop grande mixité des codes à la première itération peut engendrer la perte d'un ensemble de centres éventuellement intéressants pour les itérations suivantes.

Expérimentations sur les chapitres

Nous allons à présent classer notre corpus suivant un niveau différent de celui des codes, à savoir les chapitres. Ce niveau est le dernier avant la décomposition en articles ; nous avons recensé 5167 chapitres différents. Le recensement des différents chapitres a été effectué en fonction de leurs labels et des labels des niveaux supérieurs car des chapitres de codes différents peuvent avoir le même label. Un chapitre est, dans ce cas, défini comme la concaténation des labels des niveaux supérieurs, y compris celui du chapitre.

Le but de cette expérimentation est d'observer le comportement de l'algorithme sur un nombre plus important de centres sans pour autant modifier les liens entre les documents.

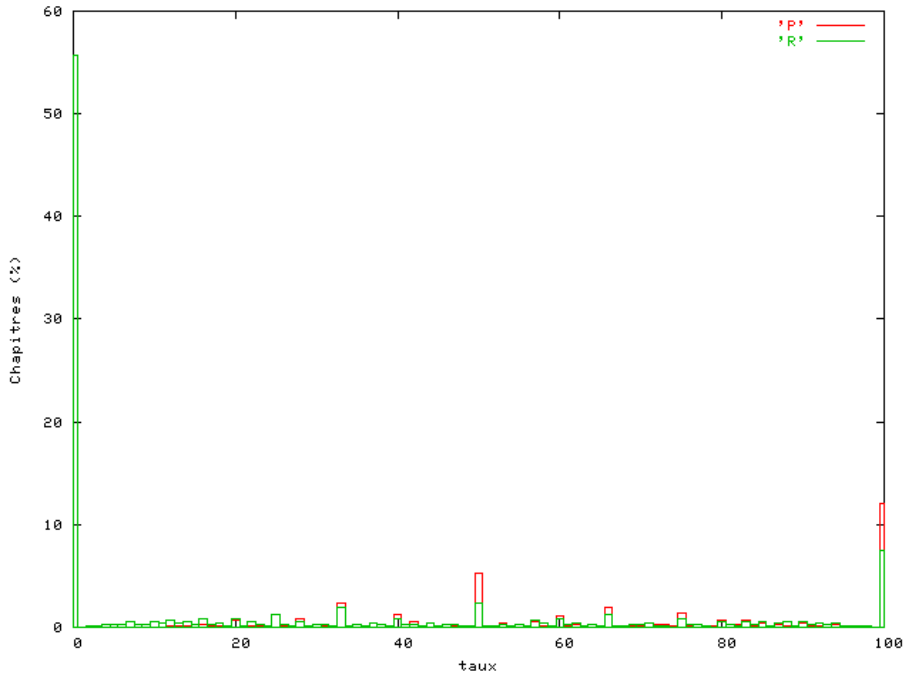

Figure 6.5 – Taux de précision (P) et de rappel (R) des classes de la partition finale sur les chapitres

La Figure 6.5 montre, pour un taux de précision donné, le nombre de classes trouvées avec la partition finale. Nous avons fait de même avec le taux de rappel. Ces deux taux ont été calculés sur le critère qui suit. Le chapitre associé à la classe est celui qui est le plus présent dans la classe en termes de documents et qui satisfait de façon ordonnée l'une des deux heuristiques suivantes :

- le chapitre n'a pas été détecté auparavant ;
- le taux de précision, ou de rappel, est supérieur à celui trouvé antérieurement.

Cette approche de l'évaluation de la partition est en adéquation avec l'objectif de représenter une classe par un thème. En effet, d'autres approches auraient pu être choisies comme, par exemple, favoriser en premier les chapitres non détectés.

L'un des premiers constats est qu'un nombre très important de chapitres n'a pas été détecté. La cause ne peut être liée aux documents non classés car tous les documents sont présents dans la partition finale. Le second constat est la répartition très large des taux de précision et de rappel avec des taux $P = 1$ ou $R = 1$ non négligeables : les classes dont $P = 1$ et $R = 1$ représentent 1 % des classes totales.

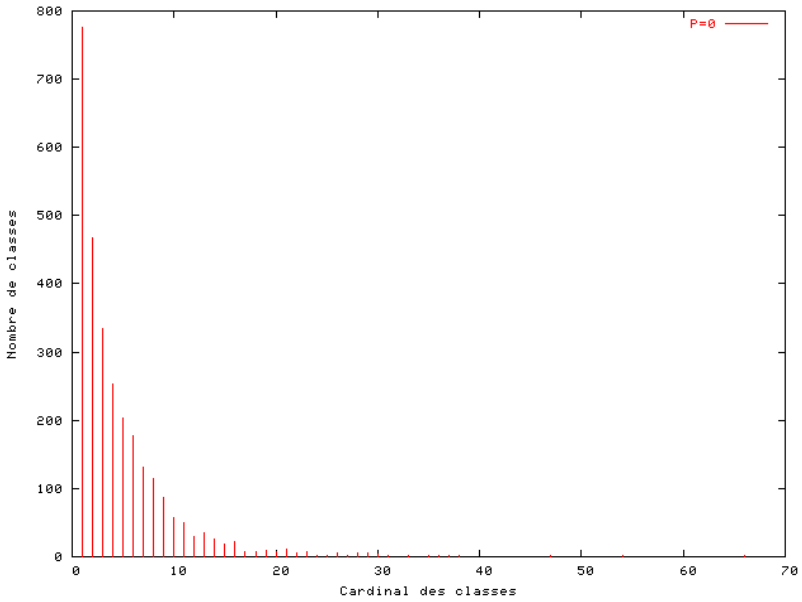

Figure 6.6 – Nombre de classes pour P=0 en fonction du cardinal

Nous avons vu que le nombre de chapitres non trouvés est important. A travers la Figure 6.6, nous essayons de détecter une caractéristique commune et nous pouvons constater que les classes non trouvées sont principalement de petite taille. La taille moyenne d'un chapitre est de 12 articles et près de 72 % des chapitres ont une taille inférieure ou égale à 12.

6.4.3.5 Conclusion

Les expérimentations ont montré que l'approche donnait des résultats encourageants. Cependant, l'algorithme souffre d'un manque de fonctionnalités propres aux algorithmes de type k-means, et notamment une fonction de fusion des classes.

L'algorithme possède quelques avantages tels que la rapidité d'exécution et une convergence rapide en moins de cinq itérations.

Les inconvénients sont d'une part, la création de classes dites doublons essentiellement par manque de fonctionnalité de fusion des classes et d'autre part, une convergence vers des minima locaux dans certains cas. Nous avons également constaté que le critère d'initialisation des centres n'était pas optimal ; cependant le critère aléatoire ne peut être suffisant et le choix d'utiliser une autre méthode de classification n'est pas envisageable pour des raisons de temps d'exécution. Une amélioration envisageable est de choisir les centres en fonction de leurs distances respectives entre eux.

6.5 Algorithme Ω-means

Nous avons cité dans le Chapitre 3 des propriétés englobant tous les types d'algorithme de classification, mais également des propriétés propres aux algorithmes de partitionnement. Certaines propriétés, indispensables pour des corpus tel que le nôtre, ne sont pas intégrées

dans les algorithmes classiques et limitent donc le champ d'utilisation de ces derniers. Nous donnons en exemple la limitation du nombre de documents à classer.
Dans cette section, nous présentons un nouvel algorithme de partitionnement, appelé Ω-means, qui intègre les principales propriétés évoquées, à savoir :

- détection automatique du nombre de classes (inconvénient majeur des algorithmes de partitionnement) ;
- classification sur un grand nombre de documents ;
- classification sur un grand nombre de paramètres ;
- classification sur un grand nombre de classes.

Cet algorithme s'inspire fortement de l'algorithme naïf décrit précédemment dans le sens où les traitements antérieurs à l'algorithme (abstraction des documents, fonction de filtrage et création de la matrice) et la fonction d'objectivité (retrouver des sous-graphes fortement connectés) sont les mêmes.

6.5.1 *K*-CDL : Estimation de *K*

Une nouvelle méthode a été conçue, implémentée et testée dans le cadre de cette thèse. Nous l'avons nommée *K*-CDL (*K* Clusters Detection using Link). Elle permet de déterminer la valeur de *K* indépendamment de l'algorithme de classification ; nous l'utilisons sur le graphe $G(V,E)$ créé durant la phase de pré-traitement.

Cette méthode repose sur les hypothèses énoncées ci-dessous :

Hypothèse 1 : les classes composant un même corpus sont de taille variée. Dans notre corpus de référence, nous constatons que le rapport entre la taille de la plus grande classe et celle de la plus petite est supérieur à 300.

Hypothèse 2 : les classes de petite taille ont un nombre de liens vers l'extérieur (i.e. avec les autres classes) inférieur par rapport aux classes de grande taille. Toutefois, cette hypothèse ne peut pas être étendue au fait que toutes les classes ont un nombre de liens vers l'extérieur proportionnellement identique.

Hypothèse 3 : les éléments d'une même classe ont des nombres totaux de liens proches les uns des autres.

A partir des hypothèses 1 et 2, on peut en déduire qu'il est plus facile d'estimer la taille réelle d'une classe ayant un nombre total de liens faible, car celle-ci a probablement un nombre de liens vers l'extérieur également faible. A l'inverse, cette tâche est alors d'autant plus difficile que le nombre total de liens d'une classe est grand.

L'hypothèse 3 permet de déduire que, si l'on trie les documents par ordre décroissant (ou croissant) du nombre total de liens, les documents d'une même classe seront proches les uns des autres sur l'axe.

Notre algorithme d'évaluation de la valeur de K est le suivant :

Soit $G(V,E)$ le graphe de notre corpus C ;

$K \leftarrow 0$;

faire :

1. sélectionner le nœud V_i non atteint de V qui possède le plus petit nombre de liens dans G ;
2. marquer comme « atteints » tous les nœuds connectés à V_i; éliminer V_i ainsi que tous les liens dans G entre V_i et les nœuds qui lui sont connectés ;
3. incrémenter K de 1 ;

tant qu'il reste des nœuds non « atteints » de G.

Algorithme 6.2 – *K*-CDL : Estimation de la valeur de *K*

Dans l'étape 2 de cet algorithme, on élimine tous les liens connectés au nœud choisi à l'étape précédente, ce qui a un double impact sur le graphe. D'une part, on élimine le sous-graphe fortement connecté. D'autre part, on élimine les autres liens connectés au nœud considéré, c'est-à-dire les liens de cette classe avec les autres classes. L'intérêt est la diminution du nombre de liens extérieurs des classes volumineuses afin de se rapprocher de leur taille réelle avec leur nombre de liens total.

On suppose qu'au moment de *casser* les liens à l'étape 2, la taille de la classe est proche de sa taille réelle c'est-à-dire que le nœud possède un nombre de liens total proche du nombre de liens du sous graphe fortement connecté (la classe contenant le nœud).

Cette démarche est réaliste en partant des petites classes et en cassant au fur et à mesure les liens des classes volumineuses avec les autres classes plus petites.

Au niveau du temps de calcul, cette méthode ne nécessite pas de faire un tri décroissant sur le nombre de liens des N documents. Ainsi, le temps de calcul sur la recherche des éléments se fait en $O(KN)$. En revanche, cette méthode impose de stocker tous les liens existant entre les différents nœuds ce qui représente pour notre corpus de référence un total de 400 Mo segmentés en 160 automates. Ces liens sont enregistrés dans un ensemble d'automates à états finis [Constant, 1995]. Chaque automate contient l'ensemble des liens de 400 documents. Les liens réciproques ne sont pas stockés. L'ensemble de ces données, de taille acceptable, peut être chargé en mémoire pour accélérer davantage l'accès en lecture. Cet accès aux liens pour un nœud donné se fait donc en temps linéaire.

6.5.2 Initialisation des centres *vs.* partition initiale

Dans l'algorithme naïf, la valeur de *K* est un paramètre de l'algorithme qui doit être fourni. La valeur de *K* connue, nous avons utilisé une heuristique pour déterminer l'ensemble des centres initiaux (voir sous § 6.4.3.1). Cette heuristique, fortement liée à la fonction de recentrage de l'algorithme naïf, est également applicable pour cet algorithme, pour les raisons évoquées précédemment : cet algorithme intègre dans sa fonction de recentrage celle de l'algorithme naïf. Cette heuristique est d'autant plus applicable qu'elle est peu coûteuse en temps de calcul.

D'un autre point de vue, durant la phase d'estimation de la valeur de *K*, nous obtenons, hormis *K*, une partition qui peut être considérée comme partition initiale de l'algorithme. En effet, cette phase doit retrouver non seulement une valeur adéquate de *K* mais aussi les classes correspondantes (en grande partie). L'algorithme peut alors être appliqué sur cette partition initiale.

Les deux méthodes sont envisageables, car elles sont toutes les deux en temps linéaire. Dans le cas d'une initialisation avec la partition initiale, on ajoute une étape à l'algorithme, même si celle-ci s'effectue en temps linéaire. Cependant, le cœur du problème est l'impact sur le déroulement de l'algorithme de classification. En effet, la question est de savoir si l'une des méthodes permet de faire converger plus rapidement l'algorithme. Dans les expérimentations, nous avons donc appliqué les deux approches pour déterminer si, d'une part, l'une des méthodes permettait de faire converger plus rapidement l'algorithme et, d'autre part, si la différence sur la qualité des résultats était significative.

6.5.3 Affectation des documents

Cette partie reste identique à celle de l'algorithme naïf, c'est-à-dire que l'on affecte chaque document au centre qui lui est le plus proche en terme de mesure de similarité.

La matrice peut éventuellement engendrer des « problèmes » lors de l'affectation des documents de par sa particularité d'être creuse. En effet, les documents[1] pour lesquels il n'existe pas de liens avec au moins l'un des centres sont inclassables. Ce cas est prévu dans l'algorithme à travers une classe dite *résidu* qui regroupe tous les documents qui ne sont pas classés après la phase d'affectation. Bien que l'idée de documents non classables semble naturelle, même si on peut toutefois considérer qu'à tout document correspond au moins une thématique, il n'en subsiste pas moins des interrogations sur la nature de ces rejets et de leurs éventuelles conséquences sur la partition finale.

Le rejet d'un document peut provenir :

1. de la phase d'initialisation (itération 0) : les centres initiaux n'atteignent pas toutes les classes. Ce problème rencontré dans l'algorithme naïf est amélioré dans cet algorithme

[1] Les documents non classés ont généralement un nombre de liens assez faible.

avec la fonction de recentrage (*cf.* section suivante). Les classes non atteintes à l'itération *i* peuvent ainsi l'être à l'itération *i+1*.

2. du faible nombre de liens du document. Cette situation est alors améliorable à l'itération suivante avec le nouveau centre.
3. d'une valeur de *K* inadéquate. Si la valeur de *K* est sous-estimée, plusieurs classes, c'est-à-dire $K_{réel} - K_{estimé}$ classes en théorie, se retrouveront dans la classe résidu.
4. de par sa nature inclassable, ce qui n'est pas concevable avec notre corpus de référence.

Les conséquences sur la partition finale dépendent bien évidemment de la nature des rejets. Une des conséquences envisageables sur la partition finale peut se concrétiser par une incapacité à atteindre la totalité des classes dans le cas où une, voire plusieurs classes, se retrouvent dans la classe résidu. Une autre conséquence concerne la précision sur chaque classe (la capacité à retrouver tous les documents de chaque classe). Cette conséquence est toutefois négligeable dans le processus d'étiquetage de la classe.

Pour résumer, un document se retrouvant dans la classe résidu, pour différentes raisons, n'est pas, malgré tout, définitivement exclu de la classification du fait de la fonction de recentrage. En revanche, les conséquences sont non négligeables dans le cas où la classe résidu contient, au final, une ou plusieurs classes (ou du moins la majorité des éléments des différentes classes).

6.5.4 Recentrage des classes

Cette méthode, qui permet de réorganiser la partition donnée, est composée de 3 étapes :

1. détection des classes *homogènes* ;
2. fusion des classes ;
3. sélection de nouveaux centres.

Une méthode de recentrage permet de représenter chaque classe de la partition trouvée par un meilleur centre (centroïde ou médoïde). La fonction appliquée repose généralement sur les distances entre les différents éléments de la classe considérée.

Notre méthode de classification est en amont d'une phase d'étiquetage de classes, afin d'alimenter un moteur de recherche. L'idée sous-jacente d'un étiquetage de classes, qui est de représenter une classe par un ensemble fini de termes les plus représentatifs, a été intégrée dans notre méthode.

6.5.4.1 Détection des classes homogènes

Une classe homogène est définie comme suit :

Soient

$P = \{P_1, ..., P_k\}$, une partition de *K* classes disjointes,

$P_i = \{c_j\}$ avec $1 \leq j \leq |P_i|$, une classe d'un ensemble fini d'éléments c_j.

Une classe homogène $\mathrm{H}(P_i)$ de P_i est un sous-ensemble de P_i dont chaque élément c_j vérifie :

$$\frac{\mathrm{N}_{P_i}(c_j)}{m} \geq \varepsilon \tag{6.10}$$

avec $m = |P_i|$ et $\mathrm{N}_{P_i}(c_j)$, qui représente le nombre de liens relatifs de c_j dans P_i, est défini comme suit :

$$\mathrm{N}_{P_i}(c_j) = \sum_{\substack{k \\ \cos(c_j, c_k) \neq 0}} 1 \tag{6.11}$$

et ε est le coefficient d'homogénéité dont la valeur par défaut vaut 0.6.

Pour une classe P_i donnée, la classe homogène correspondante regroupe les éléments qui possèdent un nombre de liens relatifs importants afin d'obtenir un ensemble d'éléments fortement connectés. Un lien relatif est un lien entre deux éléments d'une même classe, ce qui équivaut, l'orientation du lien en moins, à l'*inlink* en anglais. Cet ensemble est supposé partager une seule thématique globale.

Cette étape tend à éliminer les documents affectés à une classe par l'intermédiaire d'un terme perturbateur. Si l'on s'appuie sur l'hypothèse 2 (énoncée dans le Chapitre 4) concernant la thématique globale d'une classe, alors ces documents ne seront pas retenus dans la classe homogène, car ils auront peu de liens avec les autres éléments de la classe.
Les éléments ne faisant pas partie de la classe homogène sont, eux, affectés à la classe résidu.

Une classe dont la classe homogène correspond à l'ensemble vide est intégralement affectée à la classe résidu. On suppose qu'une telle classe ne possède pas de thématique la caractérisant.

Pour une classe P_i donnée, telle que $\mathrm{card}(P_i) = T_i$, la classe homogène correspondante peut être calculée en $O(T_i^2)$.

6.5.4.2 Fusion des classes

La fusion des classes est une étape qui, généralement, regroupe les classes proches ou similaires, c'est-à-dire dont la distance entre les centres (centroïdes, etc.) est faible : cette distance doit être inférieure à un seuil prédéfini.

Ce seuil, hormis la difficulté à le déterminer de façon efficace, permet de regrouper les classes les plus proches entre elles, c'est-à-dire que si deux classes sont fusionnées, c'est que les éléments partagent toutes ou du moins la plupart de leurs données.

Notre approche est de regrouper les classes qui partagent une même partie δ de leurs données et non la plupart ou la totalité de leurs données. Cette partie δ, déterminée pour chaque classe, doit caractériser au mieux la classe. Cette approche de la fusion de classes repose sur l'hypothèse 2 (du Chapitre 4) sur la thématique des classes et sur la composition d'un document de plusieurs thématiques : à une thématique peut correspondre plusieurs sous-thématiques différentes. Ainsi, dans cette approche, les sous-thématiques sont ignorées et nous considérons uniquement la thématique principale.

Ainsi, cette étape regroupe les classes, et plus précisément les classes homogènes, qui partagent la même thématique principale car, bien que l'on trouve une partition de classes disjointes, ces dernières peuvent partager des thématiques communes.
La thématique principale d'une classe est représentée par un ensemble de β termes, avec β une valeur déterminée expérimentalement.

6.5.4.3 Sélection des nouveaux centres

La sélection des *K* nouveaux centres est composée de deux étapes. La première étape consiste à déterminer les nouveaux centres des classes homogènes (y compris celles émanant d'une fusion). La seconde consiste à déterminer de nouveaux centres, à partir de la classe résidu, si les *K* nouveaux centres ne peuvent être déterminés à la première étape.

En considérant $\Pi = \{\Pi_i\}$, la partition en sortie de la fusion des classes, et *P* la partition initiale (en sortie de la fonction d'affectation), on a $\text{card}(\Pi) \leq \text{card}(P)$, c'est-à-dire que $\text{card}(\Pi) \leq K$. Cette inégalité est la conséquence à la fois de l'étape de détection de classes homogènes (si des classes homogènes sont vides) et de l'étape de la fusion des classes (si des thématiques communes entre des classes sont retrouvées). Par conséquent, la seconde étape doit déterminer au maximum $K - |\Pi|$ nouveaux centres. Si aucune fusion n'est réalisée et si aucune classe homogène n'est vide, alors cette étape n'est pas appliquée.

- **Centre d'une classe homogène**

  Le nouveau centre d'une classe homogène repose sur le critère suivant :

  Soit Π_i une classe et $\text{H}(\Pi_i)$ sa classe homogène correspondante, C_i est le nouveau centre de Π_i si :

$$C_i = \arg\max_{C_k} \frac{\text{N}_{\text{H}(P_i)}(C_k)}{\text{N}_C(C_k)} \tag{6.12}$$

Le nouveau centre sera donc celui qui possédera un grand nombre de liens à l'intérieur de la classe homogène et un faible nombre de liens vers l'extérieur (i.e. les autres classes). Ce critère est proche de celui de l'algorithme naïf qui, lui, s'applique sur les classes trouvées. En

appliquant ce critère sur les classes homogènes, on suppose que le risque de tomber dans des minima locaux est moindre.

En supposant que la valeur de $\mathrm{N}_{\mathrm{H}(P_i)}(C_k)$ est quasiment identique[1] pour tout élément C_k de la classe homogène $\mathrm{H}(\Pi_i)$, alors l'équation (6.12) est équivalente à :

$$C_i = \arg\min_{C_k} \ \mathrm{N}_C(C_k) \tag{6.13}$$

Le critère (6.12) ne permet pas de garantir le choix du centre le plus représentatif de la classe, mais il permet, malgré tout, de choisir un centre ayant un nombre de liens faible avec les autres classes. Ainsi, les « erreurs d'affectation »[2] diminuent avec les itérations.

– **Les autres centres**

Cette étape définit les hypothétiques nouveaux centres *manquants* (i.e. *m* centres avec $m = K - |\Pi|$). Après les phases de détection des classes homogènes et de fusion de classes, il est possible qu'un certain nombre de classes manquent. Dans ce cas, ces classes ont une particularité commune : elles se trouvent toutes, *a priori*, dans la classe résidu.

La tâche revient donc à déterminer les *m* centres parmi la classe résidu. Cette tâche est d'autant plus délicate qu'elle est équivalente à une phase de détermination de centres, avec les conséquences que cela entraîne sur la partition finale.

La classe résidu peut contenir, pour résumer, des documents de nature différente :

- les documents non affectés ;
- les documents rejetés d'une classe homogène ;
- par extension, les documents rejetés d'une classe ne possédant pas de classe homogène.

Plusieurs approches de sélection sont alors possibles :

- choisir les centres uniquement parmi les documents non affectés. On suppose ainsi que les centres manquants se trouvent exclusivement dans cette catégorie de documents rejetés (i.e. non affectés). Cette approche a l'avantage de ne pas traiter l'ensemble des documents rejetés ;
- choisir les documents parmi la classe résidu. On suppose que plusieurs classes ont pu être affectées à un même centre ;
- des approches mixtes, telles un ordonnancement suivant la nature du rejet par exemple.

Cette tâche étant équivalente à une phase d'initialisation, nous appliquons comme critère de sélection de centre celui évoqué dans la section 6.5.2.

[1] Cette hypothèse n'est pas vérifiée pour une valeur de ε faible (i.e. proche de 0). En revanche, si cette valeur tend vers 1, alors le nouveau centre sera probablement identique, qu'il soit choisi par l'un ou par l'autre critère.
[2] Les documents affectés à un centre, et donc à une classe, à laquelle ils n'appartiennent pas.

6.5.4.4 Détection et fusion

Ces deux étapes de la fonction de recentrage, bien que différentes, peuvent apparaître redondantes. En effet, la fusion des classes homogènes repose sur une partie δ de chaque classe qui peut se substituer, sous un certain angle, à la classe homogène.

La détection de classes homogènes a pour but d'éliminer les documents affectés à une classe par l'intermédiaire de termes indésirables. La fusion des classes homogènes se fait sur une représentation de la thématique principale de chaque classe (i.e. les termes les plus représentatifs d'une classe et, dans notre cas, ce sont plus précisément les termes les plus fréquents de la classe). En omettant la fonction de détection, on peut imaginer facilement retrouver la même thématique principale par la seule fonction de fusion.
Cependant, ce cas est uniquement valable lorsque la classe retrouvée est composée principalement de documents d'une classe de référence. En d'autres termes, si la classe retrouvée est composée de deux classes de référence par exemple, la thématique principale retrouvée par la seule fonction de fusion sera erronée : elle sera composée de deux thèmes principaux de référence. La fonction de détection de classe homogène éliminera une des classes de référence.

Ce phénomène est d'autant plus envisageable de par le critère de sélection des centres initiaux. La fonction de détection n'est pas à considérer, malgré tout, comme un contrepoids de la fonction d'initialisation. En effet, celle-ci est essentielle pour les raisons évoquées au paragraphe précédent. De plus, elle permet d'atténuer la convergence vers des minima locaux.

6.6 Conclusion

Dans ce chapitre, nous avons présenté deux nouveaux algorithmes de partionnement ainsi que des variantes pour l'un d'entre eux. Nous avons également apporté une réponse à la difficile question du choix du nombre de classes pour un algorithme de partitionnement. Enfin, nous avons introduit des nouveaux critères d'évaluations en adéquation avec notre objectif.

Chapitre 7

Expérimentations

Résumé

Nous évaluons l'algorithme Ω-means, décrit dans le chapitre précédent, sur notre corpus de référence (cf. Chapitre 4) et sur un échantillon de celui-ci afin de comparer avec d'autres méthodes connues. De plus, des évaluations sont effectuées avec une méthode de classification aléatoire. Nous constatons que la méthode donne des résultats suffisamment satisfaisants pour pouvoir appliquer l'algorithme récursivement.

7.1 Introduction

Dans ce chapitre, nous expérimentons et évaluons notre algorithme, Ω-means, sur le corpus de référence décrit dans le Chapitre 4. L'expérimentation se fait d'une part sur le corpus de référence dans sa totalité et d'autre part sur une partie de ce corpus (*cf.* § 7.3) : un échantillon de codes choisis aléatoirement tout en respectant des contraintes.

L'évaluation de l'algorithme consiste à faire varier certains de ses paramètres et à comparer les résultats obtenus sur notre corpus de référence. Nous proposons, dans les dernières sections, des améliorations de l'algorithme par rapport aux résultats obtenus. Cette évaluation ne correspond en rien à l'évaluation courante des algorithmes sur des corpus classiques tels que *Trec* ou *Reuters*. En effet, ces corpus, qui sont utilisés principalement au travers de campagnes, utilisent un ensemble de requêtes et, pour chacune d'elle, une liste de documents en réponse est déterminée. Ainsi, le but est de se rapprocher, pour une requête donnée, le plus possible de la liste de documents pré-déterminés et de les retrouver, de préférence, dans les premiers résultats de la liste proposée.

Notre objectif n'est pas de retrouver directement les documents dits pertinents pour une requête donnée. D'une part, nous partons du principe que, pour une même requête donnée par plusieurs utilisateurs, ces derniers n'attendent pas forcément le même type de renseignements et donc de documents. Il est dans ce cas impossible de satisfaire tous ces utilisateurs en proposant, dans les premiers résultats de la liste, les documents attendus par chacun d'entre eux. D'autre part, pour une requête donnée peu précise (générant ainsi divers contextes en réponse), il est difficile de retrouver dans les premiers résultats les documents attendus. C'est dans cette perspective que nous n'utilisons pas le jeu de questions-réponses des corpus usuels. Notre évaluation porte ainsi sur la capacité à retrouver globalement les différents codes composant notre corpus et à retrouver également les étiquettes correspondantes.

7.2 Notations

Les notations que nous allons décrire à présent sont utilisées pour définir le type d'initialisation choisie ou le nombre de centres choisis, etc.

- Init. : type d'initialisation choisi ;
- Déf. : initialisation par défaut fondée sur le critère de la section 6.4.3.1 du Chapitre 6 ;
- Init1 : Les K centres dont le nombre de liens se trouvent dans un intervalle centré sur $\frac{|C|}{K}$;
- Init2 : les K centres ayant le minimum de liens ;
- Rand : K centres choisis aléatoirement ;
- Part. init. : les centres des K classes trouvées par la méthode de l'estimation automatique de K ;
- $K57$: $K = 57$;
- $K58$: $K = 58$;

- #Itér. : nombre d'itérations de l'algorithme avant convergence ;
- Rés. : taille de la classe résidu par rapport à celle du corpus (en %) ;
- Cos : la mesure `cosine`.
- P : précision
- Q : rappel
- AC, LAC et PQ : *cf*. section 6.3.1 et 6.3.2

7.3 Corpus de référence

Le corpus de référence est composé de 57 codes et donc de 57 thématiques principales. Pour le corpus de référence, la thématique principale correspond au nom du code. La faible taille de l'échantillon de codes, au nombre de 7, permet d'appliquer des méthodes de classification qui ne supportent pas le passage à l'échelle telles que les méthodes hiérarchiques par exemple. Le corpus de référence et l'échantillon sont détaillés dans l'annexe A et des caractéristiques sont résumées dans le Tableau 7.1 ci-dessous.

Corpus	#documents	#classes	Plus petite classe	Plus grande classe
Codes	64184	57	20	6500
Sélection de codes	3153	**7**	59	966

Tableau 7.1 – Le cardinal du corpus et taille, de la plus petite et de la plus grande classe pour chaque corpus

La sélection de codes a été choisie en fonction du nombre maximum de documents que certains outils pouvaient prendre en compte tout en ayant un nombre de classes confortable. Ce dernier critère n'est pas conceptuellement indispensable, mais il est expérimentalement intéressant.

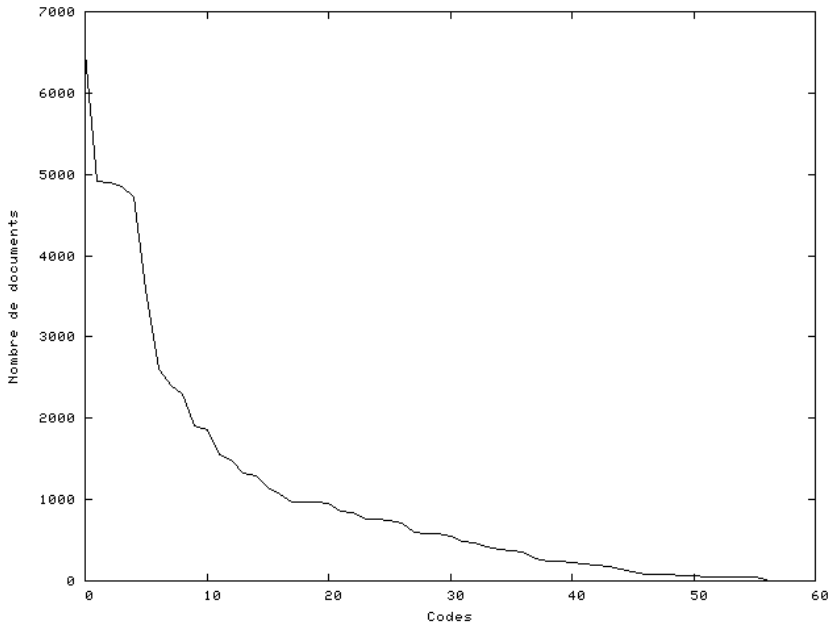

Figure 7.1 - Nombre d'articles pour chaque code du corpus de référence

La Figure 7.1 présente le cardinal de chaque code composant le corpus ; ce cardinal se situe dans un intervalle assez large. Cette forte hétérogénéité peut s'avérer être une difficulté pour retrouver les plus petites classes qui peuvent facilement se noyer dans de grands ensembles de documents.

7.4 Détection de la valeur de *K*

Nous avons appliqué notre méthode d'initialisation (détermination de *K*) sur le corpus de référence et une valeur de *K* égale à 58 a été retrouvée ; celle-ci est très proche de la valeur théorique à une classe près. Pour l'échantillon, une valeur de *K* égale à 7 a été détectée et correspond au nombre de classes de cet échantillon.

Cette méthode donne des résultats satisfaisants sur notre corpus. Toutefois, nous tentons de détecter si les hypothèses invoquées pour cette méthode sont légitimées ou pas. A noter que cette valeur correspond aux simplifications qui ont été réalisées sur l'ensemble des codes [Lame, 2002]. Ce corpus peut toutefois admettre, sous certaines considérations, d'autres solutions pour la valeur de K : par exemple, $K = 64$ qui correspond à l'ensemble des codes sans réunification.

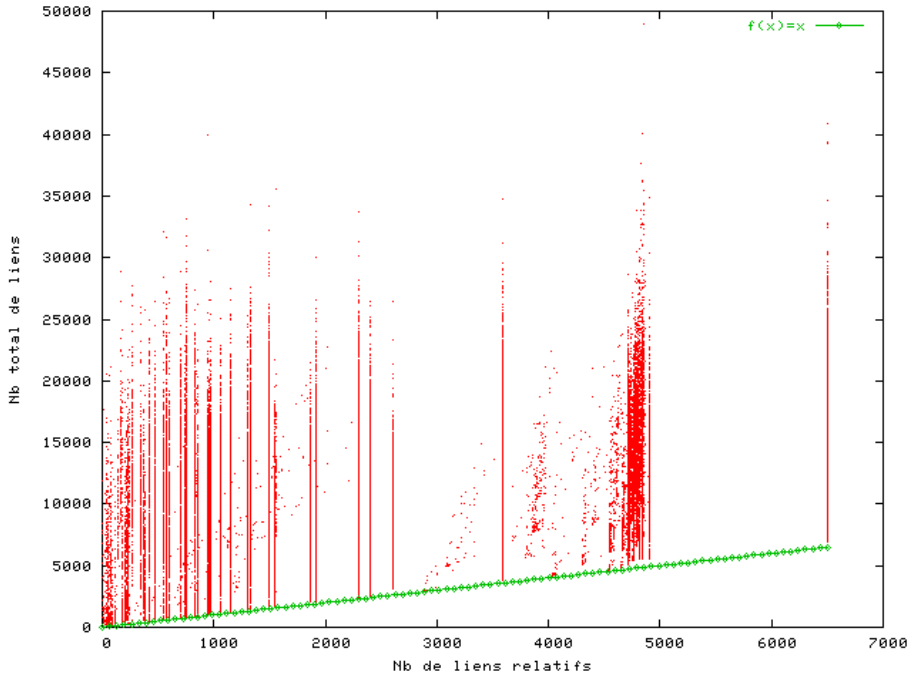

Figure 7.2 – Nombre total de liens en fonction du nombre de liens relatifs

La Figure 7.3 permet de remarquer que les classes sont des ensembles fortement connectés : pour la plupart des classes, le nombre de liens relatifs associés (*cf.* section 6.5.4.1) est relativement stable. Le nombre total de liens est quant à lui très variable et ce quelle que soit la taille de la classe. En effet, la Figure 7.2 montre que l'intervalle du nombre de liens total pour chaque classe est grand. Les hypothèses 2 et 3 ne sont donc pas vérifiées pour tous les documents : des documents appartenant à une classe de petite taille peuvent avoir un grand nombre total de liens. Cependant, cette même figure montre également que pour la plupart des classes, il existe au moins un document pour lequel le nombre total de liens est proche du nombre de liens relatifs. C'est sur ce type de document que l'initialisation s'opère, même si tous les documents ne vérifient pas les hypothèses 2 et 3.

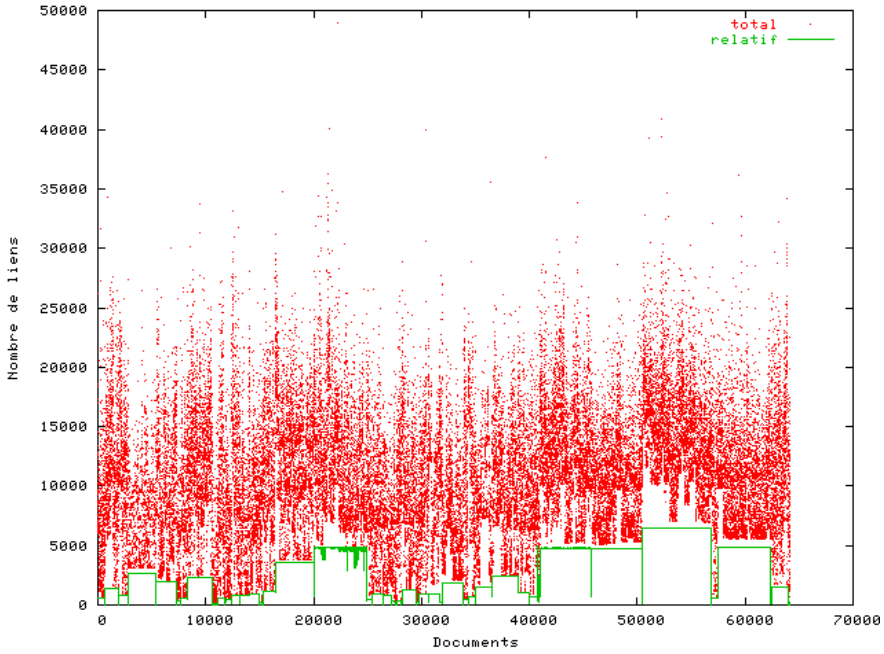

Figure 7.3 – Nombre de liens relatifs et nombre total de liens pour chaque document : tri des documents par code

7.4.1 Limitation

Cette méthode de détection, donnant de bons résultats sur notre corpus, ne peut cependant s'appliquer sur tout type de corpus. De par ses caractéristiques, cette méthode ne peut s'appliquer sur des corpus dont les éléments C_j vérifient :

$$\frac{N_{\Pi_i}(C_j)}{N_C(C_j)} << 1 \qquad (7.1)$$

La Figure 7.4 nous montre qu'il est difficile, à partir d'un document ayant un nombre total de liens élevé, de déterminer un sous-ensemble proche de la classe à laquelle il appartient. Par contre, cela devient envisageable pour les documents ayant un nombre total de liens faible.

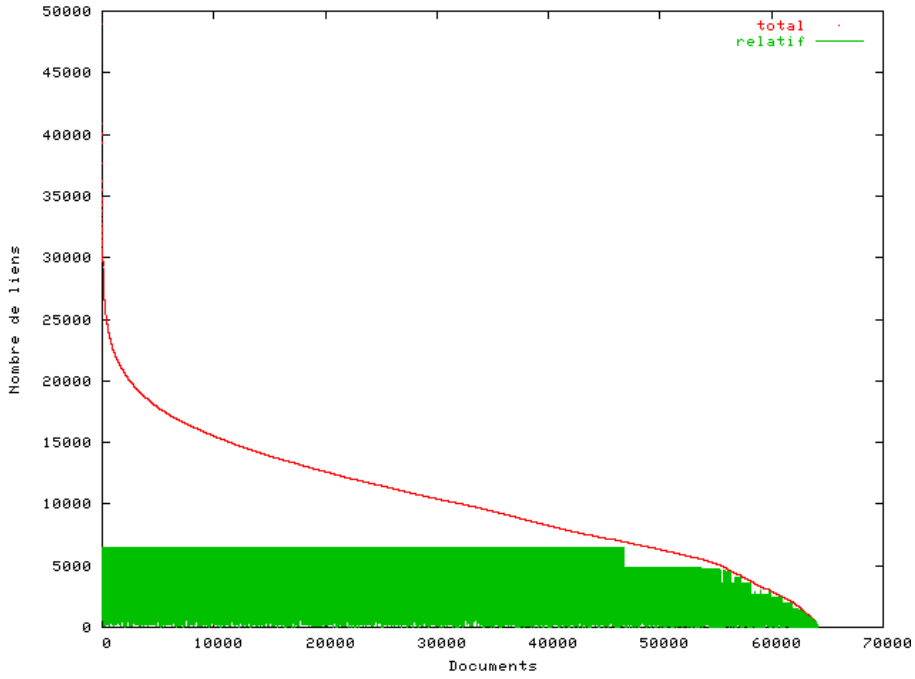

Figure 7.4 – Nombre de liens relatifs et nombre total de liens de chaque document : tri des documents par rapport au nombre total de liens

7.4.2 Partition initiale

Nous nous intéressons à la partition initiale trouvée à partir de l'étape précédente.

Dans le Tableau 7.2, nous présentons la liste des thématiques non valides, c'est-à-dire que ces thématiques ne correspondent pas aux thématiques des codes.

Termes	#occurrences	Rappel	Précision
ouvrage art	35	100.00	68.63
qualification professionnel	67	98.53	56.78
marin marchand	177	99.44	97.79
code déontologie	246	98.40	100.00
régime particulier	413	99.28	98.57
domaine public	500	87.87	100.00
domaine état	658	80.93	96.06
communauté européen	442	78.37	50.17
règle général	892	72.64	89.20
victime guerre	1814	92.13	98.16
alinéa article	1772	44.85	98.23
livre ier	127	2.19	100.00

Tableau 7.2 – thématiques non valides trouvées

La terme 'code déontologie' n'est pas considéré comme thématique car le code de déontologie n'existe pas : il existe plusieurs codes de déontologie faisant référence à des activités précises.

7.4.3 Influence de la partition initiale sur la partition finale

Nous avons appliqué l'algorithme avec des séquences d'initialisation différentes.

- La première séquence est obtenue à partir du critère d'initialisation de l'algorithme naïf -équation (6.7)- dont les caractéristiques sont résumées dans le Tableau 7.3 ci-dessous. On peut remarquer que les centres sont répartis dans 14 codes différents.

Codes	#Centres	Codes	#Centres
CASSURA	2	CPOSTES	2
CAVIACI	2	CPROCIV	1
CCONSTR	1	CPROCPE	4
CGIMP	28	CRURAL	2
CMARPUB	1	CSECSOC	7
CMONFIN	2	CTRAVAI	1
CORGJUD	2	CURBANI	2

Tableau 7.3 – Caractéristiques des centres avec l'initialisation du critère (6.1)

- La seconde séquence d'initialisation consiste à utiliser la partition initiale trouvée avec la détection de K pour déterminer les K centres initiaux. Les centres choisis permettent d'atteindre 51 codes.

7.5 Différentes valeurs de *K*

7.5.1 Valeur théorique

La valeur observée de K est 57, correspondant aux différents regroupements de codes effectués. Nous appliquons notre méthode en prenant cette valeur pour K et en utilisant, comme séquence d'initialisation, celle par défaut.

Nous avons énuméré, dans le Tableau 7.4, une partie des classes trouvées. En effet, ne sont pas présentes dans ce tableau les classes dont le code le plus représentatif est déjà présent dans une autre classe. Pour les classes ayant le code le plus représentatif en commun, celle qui possède le plus grand nombre de représentants du code est donnée dans le tableau. Ainsi dans ce tableau sont énumérées 47 classes. On remarque que les valeurs de rappel sont globalement élevées : les codes sont donc assez bien regroupés.

Parmi les codes représentatifs « doublons », on retrouve les codes suivants : CCOMMER, CGIMP, CRURAL, CSECSOC, CURBANI.

Code	#Eléments	Précision	Rappel
CACTSOC	577	0.89	0.82
CASSURA	1333	0.92	0.83
CAVIACI	838	0.99	0.72
CCIVILL	2605	0.86	0.99
CCOMMER	1915	0.87	0.79
CCOMMUN	380	0.78	0.98
CCONSOM	601	0.91	0.80
CCONSTR	2295	0.98	0.72
CDABBOI	55	0.86	**1.00**
CDCHIRD	87	0.34	**1.00**
CDVIMAR	84	0.49	**1.00**
CDWOMET	582	0.89	0.96
CDXFLUV	224	0.61	0.99
CDYANES	477	0.92	0.87
CEDUCAT	757	0.98	0.91
CELECTO	857	0.99	0.86
CENVIRO	971	0.96	0.87
CEXPROP	237	0.33	0.99
CFAMILL	41	0.34	0.98
CFOREST	1149	0.96	0.92
CGCTERR	3586	0.99	0.79
CGIMP	4901	0.92	0.64
CINDCIN	59	0.95	**1.00**
CJURFIN	966	0.92	0.98
CJUSADM	757	0.96	0.99
CJUSMIL	370	0.65	0.98
CLEGHON	176	0.93	**1.00**
CMARPUB	351	0.43	0.98
CMONFIN	1299	0.98	0.70
CMUTUAL	205	0.77	0.97
CORGJUD	951	0.93	0.90
CPENALL	964	0.74	0.79
CPENSIC	275	0.42	0.84
CPENSIM	1861	0.95	0.98
CPORMAR	421	0.95	0.90
CPOSTES	708	0.95	0.86
CPROCIV	1554	0.91	0.95
CPROCPE	2404	0.99	0.79
CPROINT	1061	0.96	0.97
CROUTE	744	0.90	0.68
CRURAL	4911	0.98	0.66
CSANPU	4720	0.95	0.91
CSECSOC	6500	0.97	0.63
CSERVNA	555	0.83	0.92
CTRAVAI	4845	0.93	0.71
CURBANI	1493	0.88	0.51
CVOIRIE	244	0.89	0.91

Tableau 7.4 - Liste des codes trouvés avec *K*57 et suivant le taux de précision et de rappel.

La Figure 7.5 indique l'évolution de PQ, AC et LAC pour les différentes itérations. On remarque que l'accroissement s'effectue principalement sur l'intervalle [0, 9]. A l'itération 8, une baisse importante de la valeur des 3 critères indique que la phase de recentrage a engendré des mouvements de documents entre classes de manière significative.

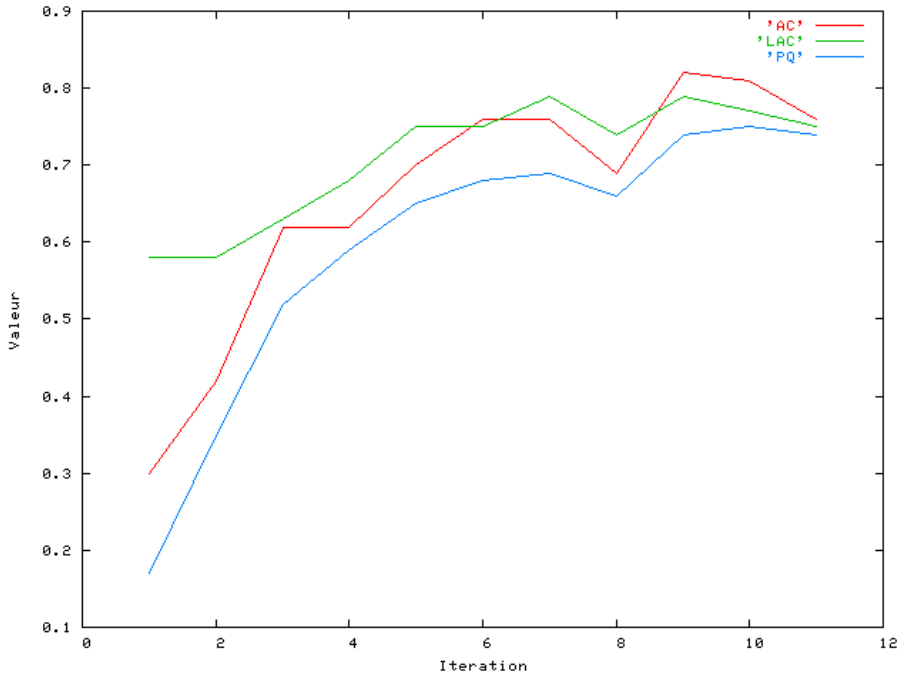

Figure 7.5 - Valeur de PQ, AC et LAC pour les différentes itérations (avec *K*57)

7.5.2 Valeur déterminée

A présent, nous appliquons l'algorithme avec la valeur de *K* déterminée automatiquement, c'est-à-dire avec une valeur de 58. Cette expérimentation permet de découvrir si la partition finale est plus ou moins identique de celle trouvée à la section précédente (avec *K*57), ou au contraire si la partition finale est en grande partie différente. On suppose que la différence doit être insignifiante car la différence entre les deux valeurs de *K* est faible.

Nous avons énuméré, dans le Tableau 7.5, 48 classes parmi lesquelles deux représentent exactement deux codes (précision=1 et rappel=1) : le code de déontologie médicale et celui des vétérinaires.

Parmi les codes « doublons », on retrouve les codes suivants : CEDUCAT, CGIMP, CRURAL, CSECSOC, CTRAVAI.

Code	#Eléments	Précision	Rappel
CACTSOC	577	0.88	0.74
CASSURA	1333	0.93	0.63
CAVIACI	838	0.96	0.89
CCIVILL	2605	0.86	0.98
CCOMMER	1915	0.91	0.76
CCOMMUN	380	0.76	0.96
CCONSOM	601	0.85	0.87
CCONSTR	2295	0.97	0.65
CDABBOI	55	0.86	**1.00**
CDCHIRD	87	0.36	**1.00**
CDMEDIC	114	**1.00**	**1.00**
CDVETER	54	**1.00**	**1.00**
CDWOMET	582	0.71	0.96
CDXFLUV	224	0.63	0.99
CDYANES	477	0.52	0.95
CEDUCAT	757	0.97	0.91
CELECTO	857	**1.00**	0.93
CENVIRO	971	0.95	0.81
CEXPROP	237	0.34	0.99
CFOREST	1149	0.85	0.70
CGCTERR	3586	0.99	0.73
CGIMP	4901	**1.00**	0.53
CGLIVPF	474	0.59	0.93
CINDCIN	59	0.95	**1.00**
CJURFIN	966	0.89	0.95
CJUSADM	757	0.95	0.97
CJUSMIL	370	0.75	0.98
CLEGHON	176	0.88	**1.00**
CMARPUB	351	0.28	0.74
CMONFIN	1299	**1.00**	0.82
CMUTUAL	205	0.67	0.48
CORGJUD	951	0.94	0.91
CPENALL	964	0.74	0.85
CPENSIC	275	0.40	0.64
CPENSIM	1861	0.95	0.98
CPORMAR	421	0.94	0.86
CPOSTES	708	0.95	0.82
CPROCIV	1554	0.90	0.93
CPROCPE	2404	0.99	0.93
CPROINT	1061	0.96	0.98
CROUTE	744	0.90	0.71
CRURAL	4911	0.95	0.63
CSANPU	4720	0.95	0.87
CSECSOC	6500	0.99	0.46
CSERVNA	555	0.86	0.99
CTRAVAI	4845	0.92	0.70
CURBANI	1493	0.89	0.65
CVOIRIE	244	0.76	0.81

Tableau 7.5 – Liste des codes trouvés avec *K*58 et suivant le taux de précision et de rappel

En comparant les résultats trouvés pour les deux valeurs de *K*, on remarque globalement que des différences existent mais des ensembles stables ressortent, ou du moins ils sont peu influencés par la valeur de *K* quand celle-ci est comprise dans un intervalle réduit.

Les deux partitions ont 45 codes en commun avec des valeurs de précision et de rappel assez proches pour certains d'entre eux, comme par exemple le code de la propriété intellectuelle.

Les codes dits doublons ont globalement un nombre de documents élevé. Cela dit, ces codes sont retrouvés avec une précision accrue et une valeur de rappel convenable. Par

exemple le code du travail pour *K*58 est retrouvé avec une précision égale à 0.92 et un rappel égal à 0.70.

Les codes non retrouvés, c'est-à-dire ceux qui ne sont représentatifs d'aucune des classes trouvées, sont pour la plupart identiques et donc indépendants de la valeur de *K*. Ils ont la particularité d'être des codes de petite taille (résumé dans le Tableau 7.6). Toutefois, notre méthode permet de trouver des codes de petite taille : par exemple le code des débits de boisson et des mesures contre l'alcoolisme qui contient 59 documents.

En se focalisant sur les codes doublons, on remarque qu'ils sont indépendants des regroupements effectués même si le code rural ancien a tendance à être regroupé au sein d'une même classe. Néanmoins quelques documents du code rural nouveau lui sont attachés. L'union du code général des impôts avec l'ensemble des annexes est justifiée par les nombreux regroupements au sein de différentes classes.

Code	#Eléments
CMINIER	187
CARTISA	45
CTRAVMA	145
CDARCHI	49
CPENSIR	81
CDSAGES	69
CDPOLIC	20

Tableau 7.6 – Liste et taille des codes non retrouvés pour les deux valeurs de *K*

Les codes non retrouvés sont tout de même classés. Le Tableau 7.7 résume les positions de ces codes dans la partition trouvée. Pour chaque code non retrouvé du Tableau 7.6, nous avons indiqué la classe où sa présence est la plus significative, et dans certains cas le code y est présent en totalité. Dans ce tableau, nous avons également indiqué le nombre de documents du code en question et la taille de la classe dans laquelle il se trouve et enfin le terme représentatif de celle-ci.

Nous constatons que ces codes sont globalement « noyés » dans des classes dont la taille excède très largement celle du code concerné. Cela s'explique en partie par le terme représentatif de la classe. En prenant l'exemple des différents codes de déontologie non retrouvés, nous remarquons que la distinction entre les différents corps de métiers abordés par ces codes (architectes, sages-femmes, etc.) n'a pas été détectée, et seul le « concept » de déontologie a été retenu dans cette classe.

Toutefois, cette séparation peut s'opérer à l'étape suivante, c'est-à-dire lors de l'application de la méthode de classification sur cette classe. Le résultat dépendra de la valeur de *K* obtenue pour cette classe. En effet, si cette valeur est plus ou moins proche du nombre de codes regroupés, il est alors imaginable de les discerner. Dans le cas contraire, il est possible de passer directement à un niveau plus bas.

Code non retrouvé	#Eléments	Code représentatif	|Classe|	Etiquette
CMINIER	124	CSECSOC	2110	Régime général
CARTISA	9	CSECSOC	2110	Régime général
CTRAVMA	123	CTRAVAI	3705	Code travail
CDARCHI	49	CDCHIRD	257	Code déontologie
CPENSIR	81	CPENSIC	552	Pension retraite
CDSAGES	69	CDCHIRD	257	Code déontologie
CDPOLIC	20	CRURAL	486	Police nationale

Tableau 7.7 – Répartition dans les classes des codes non retrouvés (pour *K*57)

Bien que le Tableau 7.7 présente les résultats pour $K57$, ceux pour la valeur de K égale à 58 ne diffèrent pas au niveau des codes représentatifs, même si les termes représentatifs, quant à eux, sont différents pour certains codes.

La Figure 7.6 montre l'évolution des critères PQ, AC et LAC pour les différentes itérations. Nous constatons qu'une forte progression du critère PQ s'opère sur l'intervalle [0, 8]. En effet, sur les 8 premières itérations, l'accroissement est de 0.54 %, tandis que sur l'intervalle [8, 11] il est seulement de 0.02 %. L'accroissement du critère AC est du même ordre malgré des phases de décroissance dont l'origine est probablement liée au choix du médoïde, et plus particulièrement au nombre total de liens de celui-ci.

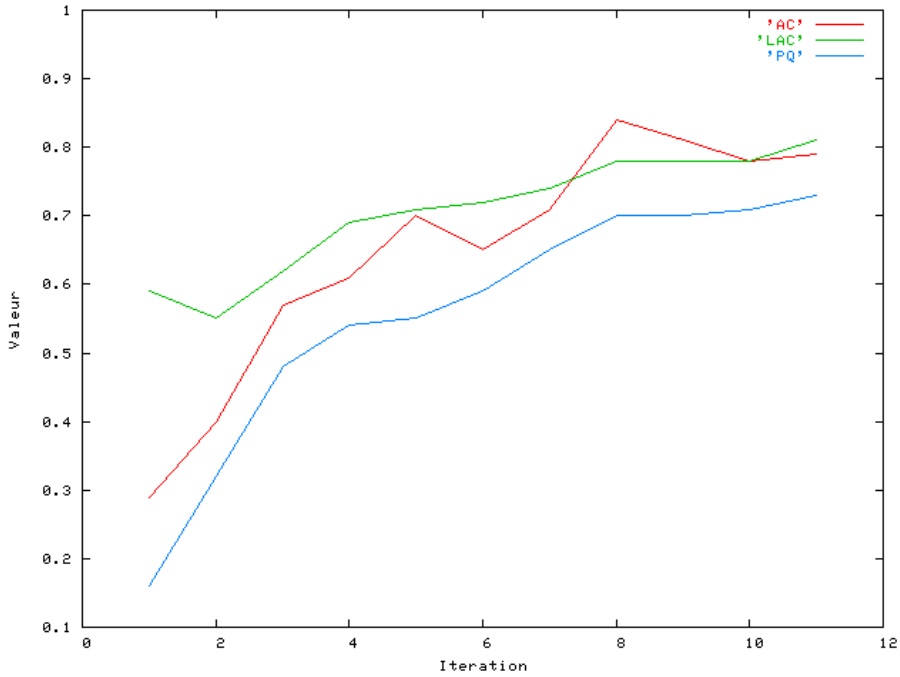

Figure 7.6 - Valeur de PQ, LAC et AC pour les différentes itérations (avec *K*58)

La Figure 7.7 montre que l'accroissement de PQ avec la valeur théorique de K est plus rapide. Cependant, on constate que globalement la qualité des résultats en terme de codes retrouvés est quasiment identique au final entre les deux méthodes.

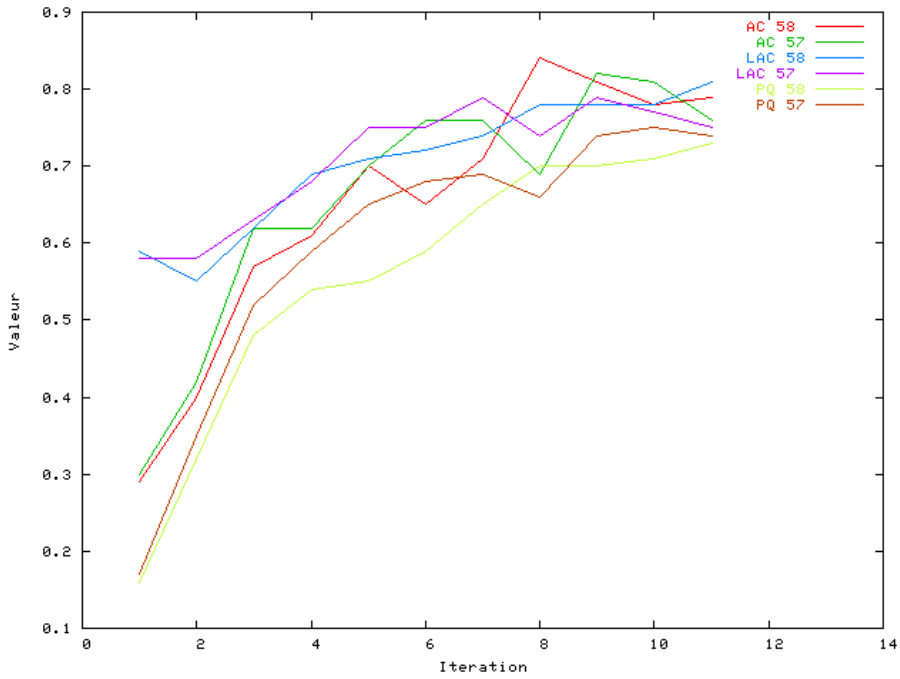

Figure 7.7 – Valeurs de PQ, LAC et AC pour les différentes itérations

Nous venons de décrire la répartition des codes à travers les classes trouvées. Nous nous intéressons à présent aux thématiques des classes, c'est-à-dire aux termes représentatifs de celles-ci.

Etiquettes incorrectes	
*K*57	*K*58
communauté européen	infraction disposition
code déontologie	établissement public
premier parti	iv journal officiel
code rural ancien	**code déontologie**
journal officiel 1er	**code rural ancien**
liquidation impôt	ministre chargé
recouvrement impôt	liquidation impôt
présent article	iii article
marin marchand	conseil supérieur
autorité administratif	règle général
liquidation judiciaire entreprise	
avis réception	
règle général	
date entrée	

Tableau 7.8 – Etiquettes non représentatives pour *K*57 et *K*58

Dans le Tableau 7.8, nous avons énuméré, pour les différentes valeurs de *K*, les thématiques non conformes des classes à l'aide d'un seul terme, le plus représentatif. Dans cette liste de termes non conformes, sont inclus les termes « code déontologie » et « code rural ancien » qui ne sont pas cohérents avec les regroupements des codes effectués. Nous constatons que la partition avec $K58$ donne de meilleurs résultats malgré une différence non significative de la qualité des partitions.

Dans le Tableau 7.9, nous mesurons la qualité des thématiques à l'aide des mesures définies dans la section 6.3.2 du Chapitre 6. La partition avec la valeur de *K* déterminée automatiquement donne de meilleurs résultats selon les critères AC et LAC ; le critère PQ est quasiment similaire pour $K57$ et $K58$.

Ces résultats permettent de conclure que notre méthode de classification avec ces paramètres par défaut donne des résultats satisfaisants et que la méthode de détection automatique de *K* permet d'améliorer globalement les performances de l'algorithme.

	$K57$	$K58$
PQ	**0.74**	0.72
LAC	0.75	**0.80**
AC	0.76	**0.79**

Tableau 7.9 – Taux de PQ, LAC et AC pour différentes valeurs de *K*

Les expérimentations menées jusqu'à présent utilisent les paramètres définis par défaut. Dans les sections suivantes, nous faisons varier les différents paramètres pour en déterminer l'impact sur les résultats et, nous comparons également notre méthode avec d'autres méthodes de partionnement de référence telles que k-means.

7.6 Influence de l'initialisation sur la partition finale

Pour la plupart des méthodes de partitionnement, l'initialisation est une étape primordiale, car elle possède un impact sur la partition finale.

Dans ce paragraphe, nous utilisons différents critères pour initialiser les centres pour déterminer ce dit impact. Ces expérimentations s'effectueront pour deux valeurs de *K*, l'une prédéfinie et l'autre définie de façon automatique à la section 7.4.

Init.	AC	LAC	#codes_atteints	PQ	Rés.	#Itér.
Déf.	**0.76**	**0.75**	**47**	**0.74**	**1**	11
Init1	0.59	0.63	43	0.60	11	9
Init2	0.35	0.26	29	0.33	26	**8**
Rand	0.70	0.71	**47**	0.72	**1**	11

Tableau 7.10 – Evaluation des partitions finales pour différentes initialisations avec *K*57

Dans un premier temps, nous avons mené nos expérimentations avec 57 centres et nous avons utilisé 4 initialisations différentes ; les résultats qui en émanent sont présentés dans le Tableau 7.10. L'initialisation « part. init. » n'est pas utilisée avec $K57$, car elle est le fruit de la méthode de détection de *K* et son utilisation n'a donc aucun sens.

On constate que notre critère d'initialisation donne les meilleurs résultats, et ce pour tous les critères d'évaluation. Le choix aléatoire des centres donne également des résultats satisfaisants mais on ne peut se fier à une initialisation aléatoire. Les deux autres initialisations, Init1 et init2, donnent des résultats moyens même si la convergence est plus rapide dans les deux cas. Les résultats avec Init2 étaient attendus dans le sens où les centres, ayant un nombre de liens minimum, handicapent l'algorithme dans son efficacité à regrouper les codes malgré une fonction de fusion.

Init.	AC	LAC	#codes_atteints	PQ	Rés.	#Itér.
Déf.	**0.79**	**0.80**	**48**	**0.72**	**1**	11
Part. init.	0.77	0.77	**48**	**0.72**	**1**	**10**

Tableau 7.11 – Evaluation des partitions finales pour différentes initialisations avec *K*58

Dans un second temps, nous avons choisi 58 centres pour mener nos expérimentations - Tableau 7.11- ; cette valeur a été déterminée dans le § 7.4. Nous avons sélectionné uniquement deux méthodes d'initialisation, à savoir la méthode Déf. qui donne les meilleurs résultats (*cf.* Tableau 7.10) et la méthode utilisant la partition initiale trouvée : « Part. init. ».

Le premier constat est que le critère d'initialisation donne à nouveau les meilleurs résultats même si les deux partitions entraînent des résultats très analogues. Le second constat est que la partition initiale n'apporte aucune valeur ajoutée par rapport au critère. D'un autre point de vue, cette méthode permet d'obtenir à la fois un nombre de centres adéquat et un ensemble de centres donnant des résultats très satisfaisants en comparaison des meilleurs résultats.

Init.	AC	LAC	#codes_atteints	PQ	Rés.	#Itér.
Part. init.	0.65	0.64	46	0.71	12	3

Tableau 7.12 – Evaluation de la partition finale de l'algorithme naïf avec la partition initiale

Le Tableau 7.12 présente les résultats obtenus en utilisant l'algorithme naïf (*cf.* section 6.4 du Chapitre 6) et part. init., c'est-à-dire pour une classification sur 58 centres.

Les résultats sont très en deçà de ceux du Tableau 7.11 avec un nombre de documents non classés important. Toutefois, la convergence a été atteinte en trois itérations seulement.

On remarque que les résultats restent malgré tout intéressants et le fait que la convergence est rapide laisse penser que l'algorithme naïf pourrait être utilisé dans une méthode de classification dynamique.

En conclusion, notre critère d'initialisation par défaut donne dans tous les cas, c'est-à-dire avec *K*57 ou *K*58, les meilleurs résultats. Au regard des différents résultats obtenus, notre algorithme reste sensible à l'initialisation. Ainsi, le choix aléatoire des centres, donnant malgré tout pour la partition présentée des résultats satisfaisants, ne peut être une solution adéquate.

7.7 Classification suivant des mesures et des distances différentes

Dans ce paragraphe, nous faisons varier l'un des paramètres essentiels de l'algorithme : la mesure de ressemblance. Nous nous attachons principalement à des distances couramment utilisées et susceptibles de donner de meilleurs résultats que `cosine`. Ainsi, deux distances ont été choisies, à savoir la distance Euclidienne et la distance de Manhattan.

Distance	Euclidienne		Manhattan	
K	57	58	57	58
AC	**0.31**	0.22	0.5	**0.58**
LAC	0.6	**0.63**	0.66	**0.70**
#codes_atteints	45	45	46	**47**
PQ	**0.31**	0.28	0.45	**0.48**
Rés.	1	1	1	1
#Itér.	11	11	11	**7**

Tableau 7.13 – Evaluation des partitions obtenues avec différentes distances et différentes valeurs de *K*

Le Tableau 7.13 résume les résultats obtenus pour ces deux distances et pour deux initialisations différentes, l'une avec *K* prédéterminé et valant 57 et une autre avec *K* déterminé automatiquement et valant 58. Le choix de la distance ne semble pas interférer sur la méthode de détection automatique de *K*.

7.8 Coefficient d'homogénéité

Dans ce paragraphe, nous nous focalisons sur l'unique « seuil » de l'algorithme, à savoir le coefficient d'homogénéité intervenant dans la phase de détection des classes homogènes.

La Figure 7.8 montre l'évolution de l'indice PQ en fonction des différentes valeurs du coefficient comprises dans l'intervalle [0.5, 0.9]. Celle-ci indique que la valeur de l'indice est maximale pour un coefficient égal à 0.6.

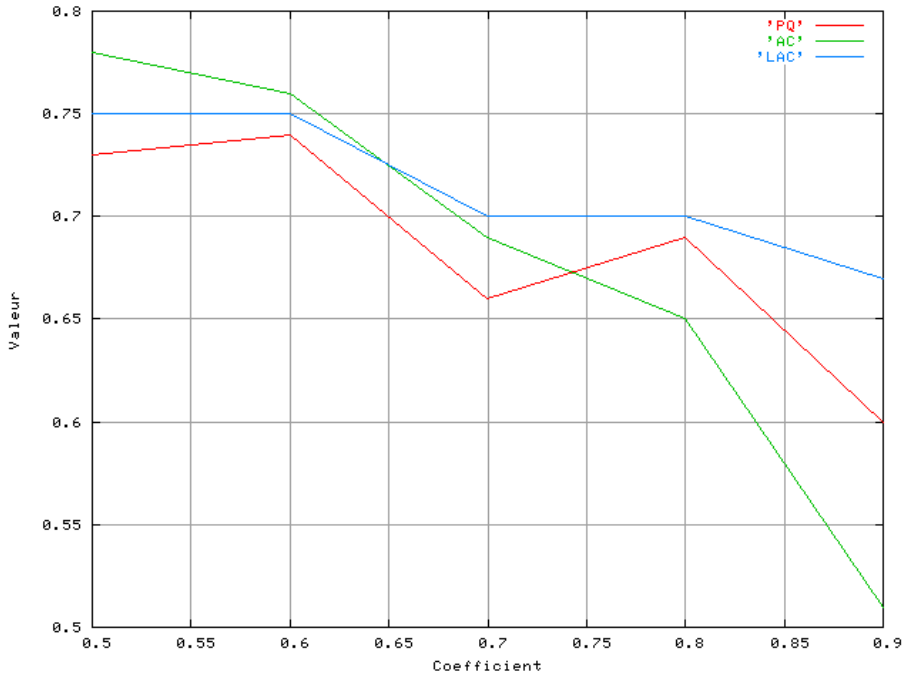

Figure 7.8 – Evolution des différents critères en fonction du coefficient d'homogénéité

Coefficient d'homogénéité ε	AC	LAC	#codes_atteints	PQ	Rés.
0.5	**0.78**	**0.75**	47	0.73	**1**
0.6	0.76	**0.75**	47	**0.74**	**1**
0.7	0.67	0.70	45	0.66	**1**
0.8	0.63	0.70	**48**	0.69	5
0.9	0.55	0.67	43	0.60	**1**

Tableau 7.14 – Evaluation de la partition finale pour différentes valeurs du coefficient d'homogénéité

Dans nos expériences, nous faisons varier notre coefficient d'homogénéité entre 0.5 et 0.9 avec une période de 0.1. Nous supposons qu'en deçà, ce coefficient n'est pas efficace, car non discriminant, pour la détection de classes homogènes. Dans le Tableau 7.14, nous évaluons les partitions finales obtenues avec différentes valeurs du coefficient d'homogénéité dans l'intervalle décrit précédemment et avec les autres paramètres par défaut. Nous pouvons constater que chaque métrique évolue de façon particulière et qu'il n'existe pas une valeur du coefficient d'homogénéité pour laquelle la partition finale correspondante rassemble la meilleure valeur pour chaque métrique. Au regard des résultats obtenus pour les métriques AC et LAC, nous pouvons déduire que l'impact du coefficient sur la détection des étiquettes est minime pour l'intervalle [0.5, 0.6]. Au delà de cet intervalle, plus la valeur du coefficient est grande et plus les valeurs des critères AC et LAC décroissent.

La détection de classes homogènes a un impact évident sur le choix des nouveaux centres et donc sur le résultat de la partition finale. Le coefficient ne peut être trop élevé, car il risque d'être pénalisant dès la première itération si l'on utilise le critère d'initialisation par défaut. En effet, les centres initiaux ont un nombre total de liens importants et peuvent engendrer du bruit. Un coefficient élevé n'est dans ce cas pas inadapté et peut favoriser la convergence vers des trous locaux, c'est-à-dire une convergence prématurée vers un centre fixe d'une itération à l'autre

Le Tableau 7.14 montre que les meilleurs résultats sont obtenus pour une valeur du coefficient d'homogénéité comprise dans l'intervalle [0.5, 0.6]. Dans les sections suivantes, la valeur par défaut sera de 0.6. Toutefois, des expérimentations faisant varier le coefficient dans l'intervalle défini seront effectuées.

7.9 Comparaison avec d'autres méthodes

Dans ce §, nous mettons en concurrence notre algorithme avec k-means, ND, c'est-à-dire avec des algorithmes de partitionnement de référence, ainsi que des algorithmes hiérarchiques.

Codes	#Eléments
CGLIVPF	474
CINDCIN	59
CJURFIN	966
CJUSADM	757
CJUSMIL	370
CLEGHON	176
CMARPUB	351

Tableau 7.15 – Codes constituant l'échantillon.

Pour ces expérimentations, nous utilisons un échantillon, car certains algorithmes ont une complexité telle en temps de calcul qu'il est préférable d'utiliser un nombre limité d'éléments. L'algorithme des nuées dynamiques (ND), dont une version est disponible à l'adresse suivante : http://www.pge.cnrs-gif.fr/bioinfo/nuees/, charge en mémoire l'intégralité de la matrice limitant ainsi la taille de la matrice.

Hormis le fait de comparer entre elles différentes méthodes, l'avantage d'utiliser cet échantillon est de découvrir l'évolution des résultats pour notre algorithme suivant la taille du corpus.

L'échantillon choisi est composé de 7 codes présentés dans le Tableau 7.15. L'élaboration de cet échantillon repose sur différents critères :

- le nombre maximum d'éléments d'une matrice pouvant être chargés en mémoire ;
- la taille des classes diversifiées.

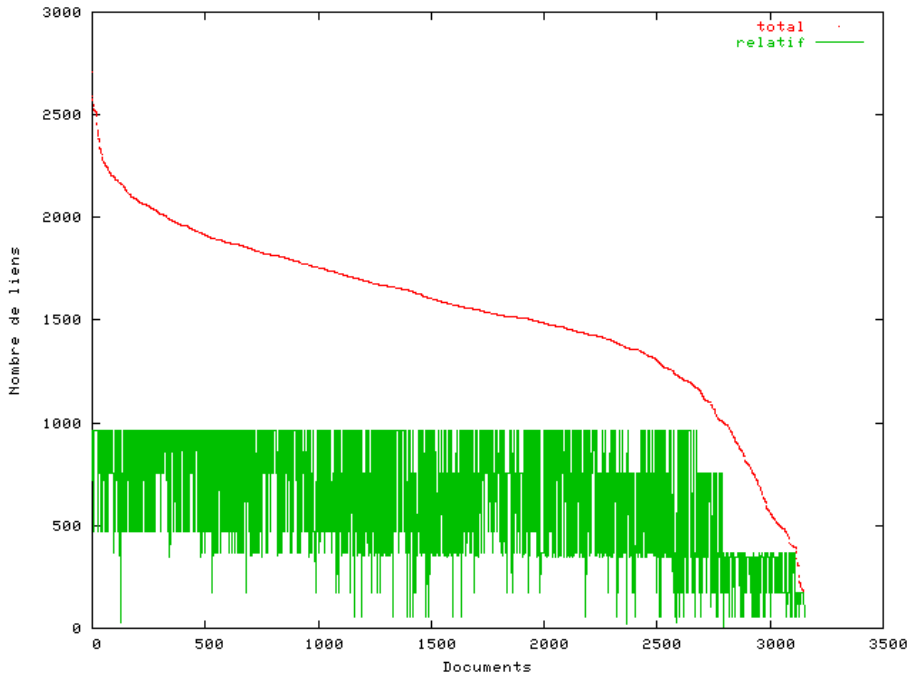

Figure 7.9 - Echantillon - Nombre de liens relatifs et nombre total de liens de chaque document : tri des documents par rapport au nombre total de liens

La Figure 7.9 montre que pour un code donné, c'est-à-dire pour un nombre de liens relatifs identique pour chaque élément le composant, le nombre total de liens varie dans un large intervalle ; ce phénomène est identique pour le corpus de référence.

Codes	Etiquettes	Précision	Rappel
CGLIVPF	Livre procédure	0.75	0.73
CINDCIN	Code industrie cinématographique	0.97	**1.00**
CJURFIN	Code juridiction financier	0.96	0.76
CJUSADM	Conseil état	0.85	0.99
CJUSMIL	Code justice militaire	**1.00**	0.99
CLEGHON	Légion honneur	0.99	**1.00**
CMARPUB	Etablissement public	0.78	**1.00**

Tableau 7.16 – Précision et rappel des classes de la partition finale

Le Tableau 7.16 présente l'évaluation des classes de la partition finale avec les paramètres et l'initialisation par défaut. On remarque que les résultats sont intéressants car plusieurs codes sont retrouvés en quasi-intégralité : taux de précision et de rappel proche ou égal à 1.

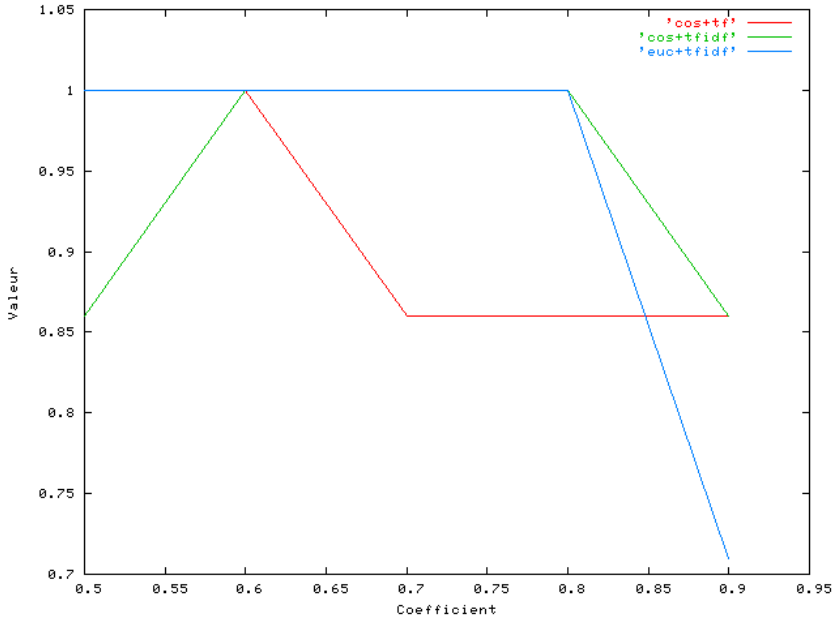

Figure 7.10 – Valeur de LAC pour différentes partitions en fonction du coefficient d'homogénéité

La Figure 7.10 montre l'évolution du critère LAC pour des distances et des pondérations différentes et ce, en faisant varier le coefficient d'homogénéité. L'influence de ce coefficient est différente suivant la distance et la pondération choisies. Pour la partition « cos+tf », la valeur maximale du critère est obtenue sur une seule valeur du coefficient ; ce qui n'est pas le cas des deux autres partitions puisque cette valeur maximale est obtenue sur un intervalle. La distance euclidienne est bien plus performante sur l'échantillon que sur le corpus.

Param.	Init.	AC	LAC	#codes_atteints	PQ	Rés.	#Itér.
Cos+TfIdf	Déf.	0.53	0.71	**7**	**0.92**	**0**	6
Variante[1]	Déf.	0.78	**1.00**	**7**	0.85	**0**	5
ND	×	**0.93**	0.85	**7**	0.85	**0**	10
K-Means	×	0.35	0.28	2	0.28	**0**	**4**
Lien simple	×	0.11	0.43	3	0.14	**0**	×
Lien complet	×	0.24	0.43	3	0.15	**0**	×

Tableau 7.17 – Evaluation de différentes méthodes de classification

Le Tableau 7.17 regroupe les évaluations des partitions obtenues avec des méthodes de partitionnement telles que k-means ou encore les nuées dynamiques, et des méthodes hiérarchiques. Nous avons également évalué notre méthode avec les critères par défaut –*cf.* Tableau 7.16- ainsi que notre variante des centres manquants.

[1] *Cf.* § 7.12.2.

Pour la méthode des nuées dynamiques, nous avons utilisé la distance euclidienne pour le calcul de distances entre documents. Pour toutes les autres méthodes, la mesure `cosine` a été utilisée.

Nous constatons, à travers le Tableau 7.17, que les meilleurs scores des différents critères sont partagés entre différentes méthodes de partitionnement. Du point du vue des thématiques et plus précisément du critère LAC, l'un des critères qui nous intéressent le plus, notre méthode avec la variante est la seule à avoir trouvé toutes les thématiques. Du point de vue du critère AC, la méthode des nuées dynamiques a été précise pour les classes de grandes tailles d'où un taux assez élevé. On constate que notre méthode par défaut bien que donnant des résultats intéressants est en deçà de la variante. La méthode k-means, quant à elle, a engendré une partition finale peu performante compte tenu que l'une des classes regroupe près de la moitié de l'échantillon. Toutefois, l'algorithme a convergé très rapidement même si notre méthode de la variante a nécessité juste une itération supplémentaire.

Les méthodes hiérarchiques n'ont pas donné de résultats satisfaisants. Le comportement de ces méthodes est identique à celui décrit dans le Chapitre 3. Nous constatons, par exemple, que le lien simple a généré une classe regroupant la majorité des éléments.

La méthode par défaut et la variante ont été utilisées avec $\varepsilon = 0.6$. Les résultats, avec $\varepsilon = 0.5$, sont identiques pour la méthode de la variante ; le taux de AC est supérieur pour la méthode par défaut : AC = 0.67, et ce pour une itération de moins.

En comparant les résultats obtenus sur le corpus *C* et sur l'échantillon, notre méthode est étonnamment moins performante sur un corpus de petite taille. Cependant, la méthode des variantes, qui, elle, est bien plus performante sur les deux corpus, démontre que ce désagrément provient essentiellement de la gestion des nouveaux centres. Ainsi, le gain de la variante n'étant pas si important sur *C*, elle prend, en revanche, une nette importance sur des petits corpus.

Cette comparaison de résultats montre qu'une dégradation se produit en fonction de la taille du corpus. Toutefois, nous ne pouvons pas définir si cette dégradation est de type linéaire ou bien exponentielle.

7.10 Classification aléatoire

La classification aléatoire est une méthode de comparaison simple même si celle-ci n'est plus très répandue.

	AC	LAC	PQ	Classe min.	Classe max.
Rand1	0.06	0.01	0.01	1043	1208
Rand2	0.09	0.14	0.04	282	2100

Tableau 7.18 – Evaluation de classifications aléatoires avec *K*57

Nous avons effectué deux séries de tests, nommées Rand1 et Rand2, et chacune d'entre elles est composée de cinq classifications. Les résultats présentés dans le Tableau 7.18 pour chaque série sont la moyenne arithmétique des résultats des cinq classifications correspondantes. La série Rand1 est une méthode aléatoire simple : pour chaque document, on détermine de façon aléatoire le numéro de sa classe, ce numéro étant compris entre 0 et *K*. La série Rand2, quant à elle, choisit aléatoirement, outre le document, le nombre de documents à classer. Les documents d'une même classe sont successifs dans la liste. Ceci permet d'obtenir de meilleurs résultats pour cette série. Toutefois, ces résultats, sans surprise, ne sont guère concluants.

D'autres méthodes aléatoires auraient pu être mises en œuvre telle que la méthode de permutation de [Farnstrom et al., 2000] qui, initialement, permet de mettre dans un ordre aléatoire une liste de documents, et qui peut être adaptée pour une méthode de classification.

7.11 Classification avec des SN et des unitermes

Pour la plupart des méthodes de classification de documents, ou des systèmes de recherche, l'une des premières étapes est l'abstraction des documents. Cette phase dont les principaux aspects sont décrits dans le Chapitre 2, et repris dans le Chapitre 4 pour la description de notre approche, utilise, dans le cadre de notre méthode, uniquement la notion de syntagmes nominaux. Ce choix a été justifié par le besoin d'unités sémantiques riches tant pour l'indexation que pour l'étiquetage des classes et donc des unités présentées à l'utilisateur, mais aussi par la nécessité d'obtenir une matrice de similarité creuse : caractéristique que l'on obtient plus facilement avec des SN que des mots, à filtrage équivalent.

Cette approche est discutable dans le sens où certains mots sont suffisamment porteurs de sens et n'ont nul besoin d'être affublés de noms, d'adjectifs, etc., pour être définis avec précision, comme par exemple le mot « surendettement ». Bien que les mots ne nous intéressent pas dans notre approche, une catégorie de termes peut s'avérer intéressante : les *unitermes*.

Les unitermes sont des mots simples qui ne sont pas compris dans un SN lors du découpage du texte. Ce sont donc principalement des noms.

Ce type de termes peut avoir comme avantage de faire ressortir les mots qui décrivent le mieux les thématiques des documents, sans l'inconvénient d'être submergé par une liste de mots. En effet, une thématique ou une sous-thématique peut être définie par un mot simple.

L'inconvénient de cette approche réside dans la construction de la matrice de similarité qui verra son nombre de mesures de ressemblance non nulles, entre documents, augmenter.

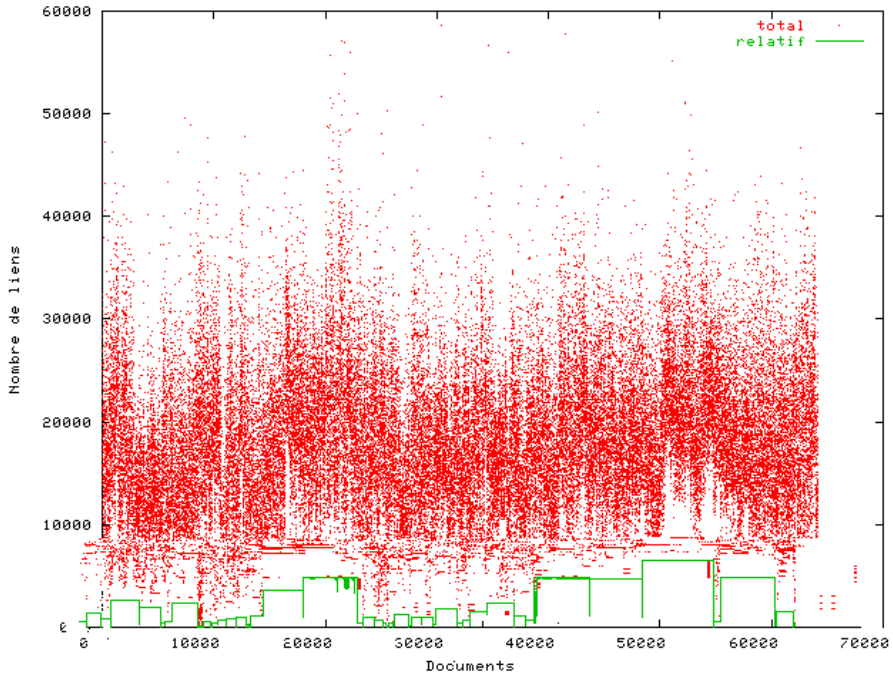

Figure 7.11 – Indexation des SN et des unitermes - nombre de liens relatifs et nombre total de liens pour chaque document : tri des documents par code

Dans un premier temps, nous avons indexé notre corpus de référence suivant la même approche que celle décrite dans le Chapitre 5 à la différence prêt que les unitermes sont à présent indexés. Le filtrage des unitermes est identique à celui des SN. Ainsi, un filtrage morpho-syntaxique est effectué, de même qu'un filtrage statistique.

Ainsi, un total d'environ 7,000 unitermes ont été indexés et pas moins de 4,500 unitermes éliminés sur le critère statistique (*cf.* §5.5) et après la réduction morpho-syntaxique. Cela représente une augmentation de quasiment 6 % du total des différents termes indexés. Sur les 4500 termes éliminés, seuls 25 ne sont pas des hapax.

Dans un second temps, nous avons construit notre matrice de liens entre documents. La conséquence immédiate est, de toute évidence, un nombre total de liens supplémentaires non négligeable.

L'expérience montre que la matrice de liens est creuse à 72 %, ce qui représente une moyenne de 17000 liens par documents. Cette matrice, avec une indexation des SN uniquement, est creuse à 83 %, ce qui représente un nombre de liens moyens par document de 10,000.

La Figure 7.11 montre que cette augmentation de liens touche tous les codes et qu'il existe très peu de documents dont le nombre total de liens se rapproche du nombre de liens relatif, ce que l'on remarque également sur la Figure 7.12.

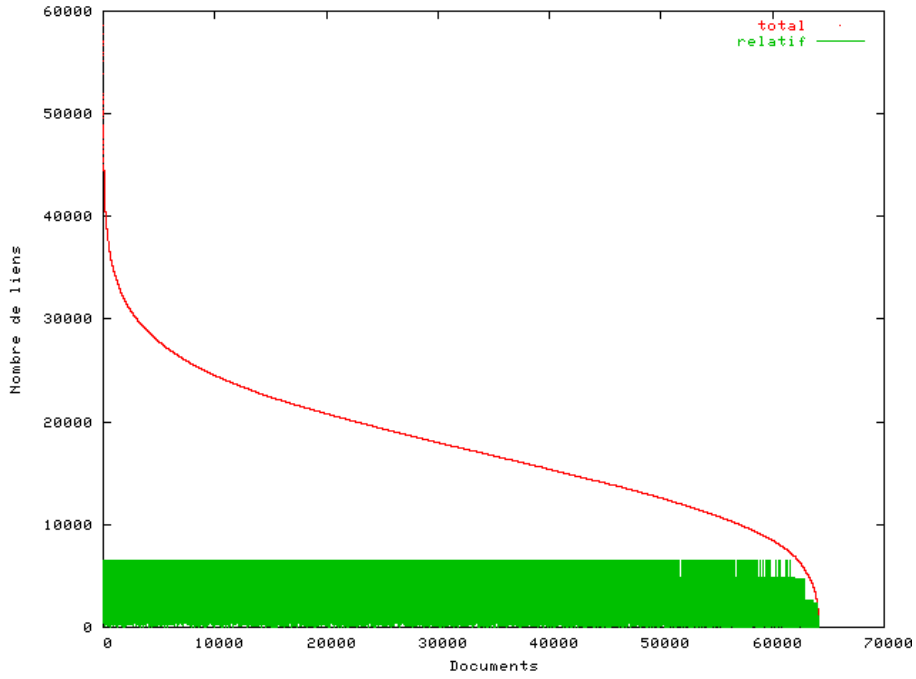

Figure 7.12 - Indexation des SN et des unitermes - Nombre de liens relatifs et nombre total de liens de chaque document : tri des documents par rapport au nombre total de liens

L'indexation des unitermes et l'augmentation du nombre total de liens va permettre de tester la robustesse de notre méthode de détection du nombre de classes, décrite dans le § 6.5.1 du Chapitre 6. Au regard de la Figure 7.12 et de l'algorithme de l'estimation de *K*, on prévoit facilement, sans pouvoir la déterminer, une valeur de *K* inférieure à celle trouvée auparavant ; ce que prouve l'expérimentation en détectant 52 classes. Cette valeur est vraisemblable si l'on regroupe les 6 codes de déontologie en un seul.

Unitermes			
Mots indexés	#Eléments	Mots rejetés	#Eléments
Travail	6168	Etat	31010
droit	5649	livre	30920
organisation	5404	décret	30277
collectivité	5245	titre	28819
délai	5182	chapitre	27770
santé	5004	conseil	24685
établissement	4803	disposition	17083
régime	4614	loi	14263
assiette	4518	condition	11031
procédure	4392	vigueur	9188
impôt	4377	cas	8040

Tableau 7.19 - Echantillon de la liste des unitermes indexés et rejetés

Le Tableau 7.19 montre un échantillon des unitermes, indexés dans le corpus de référence et, présents dans le plus grand nombre de documents. Cet échantillon est, toutefois, peu convaincant quant à la description des thématiques des classes : le filtrage des unitermes est identique à celui des SN. Parmi cette liste d'unitermes, nous trouvons des mots juridiques tels que abandon (29 documents), aliéné (29), successeur (27) ou encore abroger (43).

Pour déterminer l'impact de l'indexation des unitermes, nous avons mené des expérimentations uniquement avec les paramètres par défaut et pour les initialisations par défaut *K*57 et *K*58.

Init.	AC	LAC	#codes_atteints	PQ	Rés.	#Itér.
*K*57	0.10	0.23	29	0.30	0.1	11
*K*58	0.12	0.23	26	0.26	0.1	11

Tableau 7.20 – SN + unitermes : évaluations des partitions finales

Dans le Tableau 7.20, nous remarquons que les résultats sont très en deçà de ceux trouvés avec l'indexation des SN uniquement. Ces résultats s'expliquent par une forte présence de mots dans les étiquettes, entraînant un taux faible pour AC et LAC ainsi que PQ. En effet, le mot « famille », par exemple, est l'une des étiquettes trouvées pour les deux partitions. Bien que celle-ci regroupe le code de la famille ainsi que de nombreux éléments, elle ne peut être valide car elle est différente de l'étiquette « code de la famille ». L'impact de ces mots est que de nombreux éléments de codes différents sont rassemblés au sein d'une même classe. Ils ont beaucoup plus d'impact que les SN en l'absence d'indexation de mots.

Etiquettes			
*K*57		*K*58	
Intitulé	#Eléments	Intitulé	#Eléments
Impôt	2216	Invalidité	1893
Invalidité	1906	Impôt	1844
Aide	1474	Habitation	1412
Mayotte/Bas Rhin	1195	Etablissement	1160
Taxe	1087	Taxe	959

Tableau 7.21 – Echantillon des étiquettes trouvées pour *K*57 et *K*58

Au regard du Tableau 7.21, nous constatons que certains mots sont trop génériques tels que « taxe » ou « impôt » et regroupent donc différents aspects les concernant. Bien qu'ils soient trop génériques, la plupart des termes de ce tableau sont des termes juridiques. Cela veut dire que, si le filtrage statistique des unitermes nécessite une borne maximale moins élevée, alors une partie des unitermes éliminés seront des mots juridiques.

D'un autre point de vue, cette classification, même différente de la classification de référence, peut être intéressante expérimentalement. En effet, à partir des classes représentées

par un mot générique tel que « taxe », on peut se demander comment vont s'articuler les différents pans du droit avec les sous-thématiques du mot en question.

Nous avons appliqué notre algorithme naïf avec les paramètres et l'initialisation par défaut. L'effet des mots génériques énoncés ci-dessus est ici amplifié puisque plus de 65 % des étiquettes sont des unitermes. Les résultats sont évidemment en deçà de notre algorithme. Toutefois, l'algorithme naïf a convergé au bout de cinq itérations.

En conclusion, l'indexation des unitermes nécessite un filtrage statistique plus draconien de ces derniers pour éviter la création de liens « perturbateurs » : le taux empirique de filtrage de 5 % serait ici suffisant.

7.12 Classe résidu

La classe résidu regroupe temporairement, à chaque itération, un ensemble de documents classés de façon incorrecte pour diverses raisons, et qui représente une source pour la recherche de nouveaux centres éventuels. En théorie, cette classe peut contenir des documents à la fin de l'algorithme de classification. En pratique sur notre corpus de référence, cette classe ne doit contenir aucun ou très peu de documents.

L'analyse de cette classe en fin de processus pour différentes configurations montre qu'un petit ensemble de documents n'est pas classé après la phase d'affectation.

7.12.1 Etude

Dans ce paragraphe, nous analysons la classe résidu pour différentes configurations : celles donnant *a posteriori* les meilleurs résultats. Cette analyse a pour but de déterminer d'éventuelles caractéristiques des sous-ensembles de codes présents dans cette classe, ceci afin de dégager une fonction de gestion de la classe résidu pouvant se substituer et améliorer la fonction définie dans le § 6.5.4.3 du Chapitre 6.

Code	#Eléments		|Code|	Code retrouvé
CMINIER	14	(7%)	187	Non
CARTISA	18	(40%)	45	Non
CGLIVPF	99	(21%)	474	Non
CDVETER	37	(68.5%)	54	Non
CVOIRIE	4	(2%)	244	Oui
CDMEDIC	66	(58%)	114	Non

Tableau 7.22 – Classe résidu de la partition finale Cos+TfIdf+*K*57

Nous analysons dans un premier temps la classe résidu générée avec les paramètres suivants : Cos+TfIdf+*K*57 et dont le contenu est présenté dans le Tableau 7.22. Le premier

constat est que la plupart des codes s'y trouvant n'ont pas été trouvés dans la partition finale. Le second est que ces codes sont de taille relativement petite.

A partir de cette classe, notre approche a été jusqu'à présent de déterminer des éléments susceptibles de regrouper, lors de la phase d'affectation, la plupart des éléments de la classe résidu. Le choix de ces éléments était en phase avec notre critère d'initialisation. Toutefois, comme ce dernier, le risque est de choisir des éléments susceptibles de regrouper le même ensemble de documents, c'est-à-dire en choisissant des éléments d'un même code.

Pour pallier ce problème, nous adoptons une toute autre approche qui est de détecter, à partir de la classe résidu, le nombre de sous-ensembles présents dans cette classe. Nous utilisons donc notre méthode d'estimation de *K*, décrite dans le § 6.5.1 du Chapitre 6. Cette approche permet d'obtenir une partition et de définir pour chaque classe le noyau correspondant, les nouveaux centres étant dans ce cas, les noyaux précédemment déterminés.

Classe	Etiquette
CVOIRIE : 2	ouvrage art
CDMEDIC : 1,CVOIRIE : 1,CGLIVPF : 5	alinéa article
CDMEDIC : 1,CVOIRIE : 1,CGLIVPF : 5,CARTISA : 5	ministre chargé
CMINIER : 14	régime particulier
CARTISA : 13	code artisanat
CDVETER : 37	code déontologie vétérinaire
CDMEDIC : 64	code déontologie médical
CGLIVPF : 89	livre procédure

Tableau 7.23 – Partition trouvée pour la classe résidu

Le Tableau 7.23 montre les classes obtenues d'après la nouvelle approche décrite ci-dessus. Nous constatons que les classes ont été en très grande partie trouvées même si deux classes supplémentaires, regroupant des éléments de divers codes, sont détectées. L'approche s'avère donc intéressante tout en choisissant les noyaux suivant un ordre croissant du cardinal des classes. En effet, le nombre de nouveaux centres nécessaires sera peut-être différent du nombre de classes détectées.

Code	#Eléments		|Code|	Code retrouvé
CDABBOI	31	(56 %)	55	Non
CMINIER	98	(52 %)	187	Non
CVOIRIE	5	(2 %)	244	Oui
CEXPROP	181	(76 %)	237	Non

Tableau 7.24 – Classe résidu pour la partition finale Cos+TfIdf+K58

Nous avons effectué le même traitement sur la classe résidu obtenue avec les paramètres suivants : Cos+TfIdf+*K*58. La composition de cette classe est présentée dans le Tableau 7.24.

On constate que les codes présents dans cette classe sont également de faible taille et qu'ils y possèdent plus de la moitié de leurs éléments, hormis le code de la voirie avec seulement 2 éléments.

Classes	Etiquette
CVOIRIE : 3	Ouvrage art
CEXPROP : 4,CVOIRIE : 1,CDABBOI : 2,CMINIER : 1	alinéa article
CEXPROP : 2,CVOIRIE : 1,CDABBOI : 6,CMINIER : 12	ministre chargé
CDABBOI : 23	code débit boisson
CMINIER : 85	livre ier
CEXPROP : 175	code expropriation

Tableau 7.25 - Partition trouvée pour la classe résidu

Le Tableau 7.25 présente la partition trouvée sur cette classe résidu dont le résultat est proche de la partition trouvée précédemment : la plupart des codes sont détectés et deux classes supplémentaires ont été détectées, regroupant des éléments de divers codes.

Au regard des résultats obtenus sur les classes résidus des deux partitions, nous supposons que notre approche sur la détection des nouveaux centres peut améliorer les performances de notre méthode, ce que nous vérifions dans le paragraphe ci-après.

7.12.2 Variante des centres manquants

Suite à l'étude de la classe résidu, nous supposons que la détection des centres manquants (*cf*. le § 6.5.4.3 du Chapitre 6) peut être améliorée. Ainsi, nous proposons une variante –Algorithme 7.1– à l'approche de la sélection des centres manquants.

Soit k le nombre de centres manquants et R la classe résidu.
1 - Appliquer la méthode de détection de centres sur R.
Soit k' le nombre de centres trouvés.
2 - Trier les centres par ordre croissant de la taille des classes trouvées.
3 - Si $k<k'$, prendre les k premiers centres parmi les k'.
Sinon prendre les k' centres et choisir les (k-k') centres en appliquant la méthode précédente.

Algorithme 7.1 – Variation du choix des centres manquants

Nous avons appliqué notre algorithme ainsi modifié avec différentes combinaisons au niveau de l'initialisation en faisant varier le coefficient d'homogénéité suivant un intervalle réduit. Cet intervalle contient les valeurs du coefficient qui ont abouti aux meilleurs résultats

dans les expérimentations décrites précédemment. Ainsi, seules les valeurs de 0.5 et 0.6 ont été retenues dans les expérimentations qui suivent.

	Init.	AC	LAC	#codes_atteints	PQ	Rés.	#Itér.
*K*57	Déf.	0.80	0.77	49	0.77	2	9
*K*58	Part. init.	0.77	**0.78**	50	**0.78**	**1**	11
58	Déf.	**0.83**	**0.78**	**51**	**0.78**	**1**	**5**

Tableau 7.26 – Partitions trouvées avec des initialisations différentes et un coefficient d'homogénéité valant 0.6

Le Tableau 7.26 résume les partitions trouvées avec différentes initialisations et une valeur du coefficient d'homogénéité de 0.6. En comparant les résultats présentés ci-dessus et ceux du § 7.6, nous obtenons globalement, pour les différentes initialisations, de meilleurs résultats avec la variante des centres manquants.
D'après les critères, l'initialisation composée de *K*58+Déf. donne les meilleurs résultats avec un taux LAC de 0.78, un taux AC de 0.83 et une convergence obtenue à la 5ème itération. Notons toutefois que la différence des résultats obtenus est relative car ces trois partitions restent proches au regard des valeurs des différents critères. Seul le critère AC est déterminant : une différence d'environ 3800 documents est enregistrée entre « *K*58+Part. init. » et « *K*58+Déf. ».

	Init.	AC	LAC	#codes_atteints	PQ	Rés.	#Itér.
*K*57	Déf.	0.79	0.77	48	0.77	**1**	**6**
*K*58	Part. init.	**0.80**	**0.76**	**50**	**0.79**	**1**	11
58	Déf.	0.76	0.74	47	0.72	**1**	8

Tableau 7.27 – Partitions trouvées avec des initialisations différentes et un coefficient d'homogénéité valant 0.5

Nous avons effectué les mêmes expérimentations décrites ci-dessus mais cette fois-ci avec une valeur du coefficient d'homogénéité de 0.5 ; les résultats sont présentés dans le Tableau 7.27. Cette valeur du coefficient dégrade les taux AC et LAC des trois partitions, en comparaison des valeurs correspondantes du Tableau 7.26. Une valeur de 0.5 n'est donc pas suffisamment élevée pour éliminer des classes homogènes les éléments dont le nombre de liens avec les autres éléments de la classe est trop faible pour pouvoir récupérer la classe (i.e. le code) à l'itération suivante. En ce qui concerne la convergence, les résultats sont disparates, mais on constate que le critère d'initialisation par défaut permet de converger plus rapidement, quelle que soit la valeur du coefficient d'homogénéité et *K*.

Nous constatons que cette approche de la détection des nouveaux centres, décrite dans le § 7.12.1, donne globalement de meilleurs résultats que ceux obtenus avec l'approche par

défaut. Cette approche n'est pas pénalisante en temps de calcul si la classe résidu est de taille « raisonnable ».

7.13 Exemple de relations

Notre objectif est de retrouver les codes dans leur quasi-intégralité dans les différentes classes. Toutefois le nombre de classes dont le taux de précision et de rappel vaut 1 étant faible, la plupart des codes ont des sous-ensembles dispersés dans différentes classes. Dans ce paragraphe, nous nous intéressons à la répartition d'un code, pris comme exemple, dans les différentes classes. A travers cette répartition, nous essayons de détecter les liens qui existent entre le code en question et les autres codes, et ce en déterminant les termes qui les ont liés. Nous avons choisi le code du commerce -Figure 7.13- avec la partition suivante : *K*58+Déf.+0.6.

Les termes qui lient ces classes n'ont rien à voir avec les étiquettes trouvées pour chaque classe, même si la plupart des termes, dans l'exemple proposé, correspondent effectivement aux étiquettes des classes. Ces termes sont déterminés à partir du centre et de l'ensemble des documents du code en question.

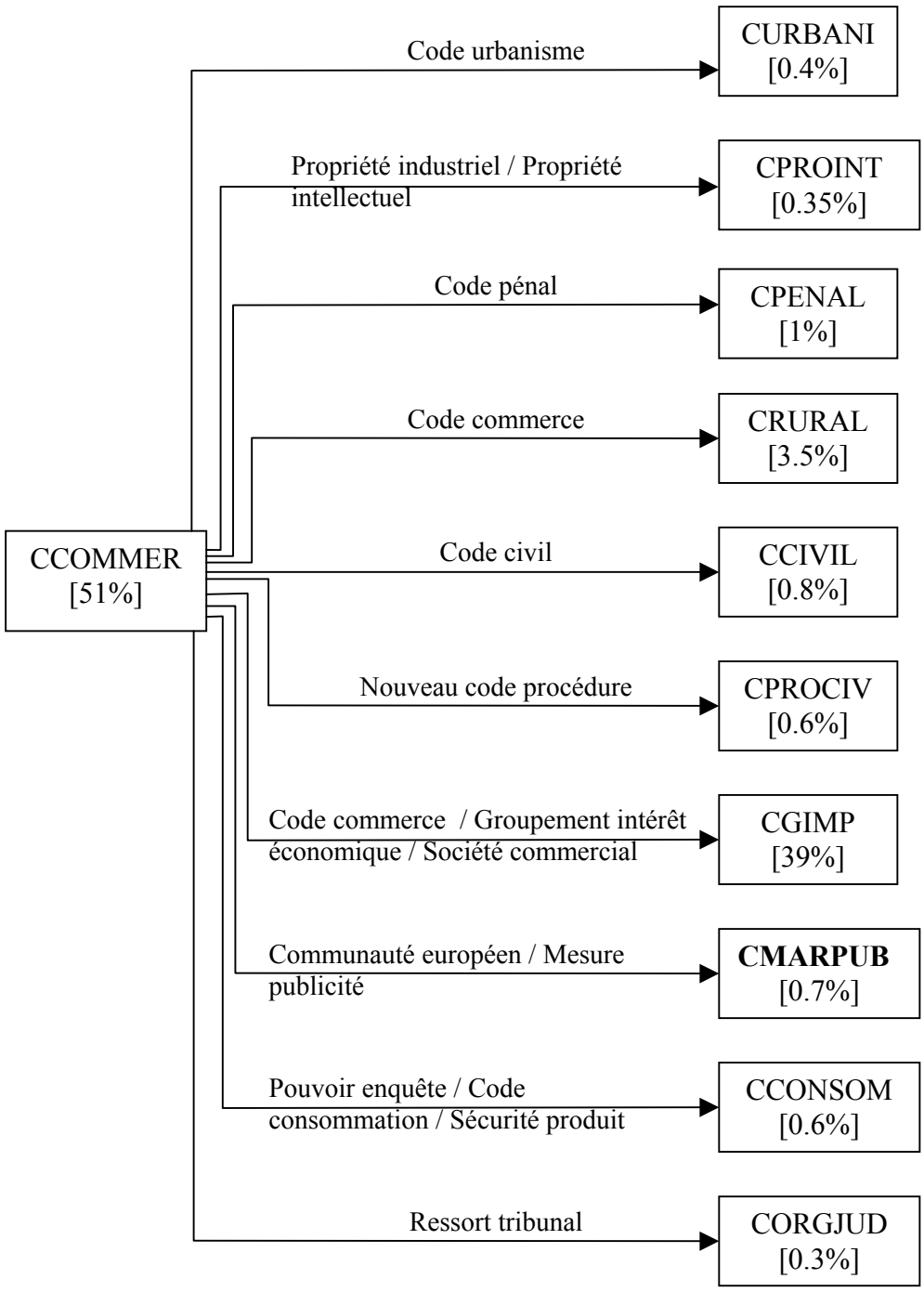

Figure 7.13 – Répartition du code de commerce dans les différentes classes

La Figure 7.13 montre qu'une classe regroupe donc 51 % du code de commerce ; une autre représentant le code de la consommation, par exemple, regroupe 0.6 % du code du commerce et ces deux classes sont liées par les termes suivant : pouvoir enquête / code consommation / sécurité produit.

7.14 Noyau

Dans nos expériences, le noyau de chaque classe, et dans notre cas le médoïde, est composé d'un seul élément. Bien que ce choix comporte quelques avantages, il peut entraîner certains inconvénients décrits dans le Chapitre 2.

Nous avons vu à travers nos expériences que ce médoïde jouait un rôle important dans la qualité de la partition finale, de par ses liens avec les éléments de sa classe (liens internes) ainsi que par ses liens avec les éléments des autres classes (liens externes). Suivant la quantité de liens externes du médoïde, la classe converge plus ou moins rapidement. La quantité des liens internes influence, quant à elle, le nombre d'éléments de la classe que l'on va trouver dans la partition finale. De plus, l'initialisation de l'algorithme, comme la plupart des algorithmes de partitionnement, influence fortement le résultat de la partition finale. Bien que le critère défini au § 6.4.3.1 du Chapitre 6 semble donner les meilleurs résultats, il n'en est pas moins possible d'essayer d'atténuer l'influence de l'initialisation tout en essayant de converger plus rapidement.

Un mode de représentation courant des classes est d'utiliser les *k* documents centraux de la classe (*cf*. Chapitre 2), où *k* est choisi empiriquement avec une valeur de 3. Cependant, cette valeur constante pour toutes les classes ne peut être plus satisfaisante pour une classe de grande taille qu'une représentation avec un seul élément. Ainsi, l'idée n'est pas d'utiliser un nombre *k* d'éléments prédéfinis et identiques pour toutes les classes, mais de définir une valeur de *k* de façon dynamique pour chaque classe.

L'objectif est donc de représenter chaque classe C_i par un ensemble de $k(C_i)$ éléments, et ce pour chaque itération. Suivant notre modèle, cette représentation est sensible au nombre de liens internes et externes. Ainsi, l'approche « naturelle » est d'identifier cet ensemble $k(C_i)$ à la classe homogène correspondante.

Cette approche entraîne des réflexions à différents niveaux de l'algorithme :

- la phase d'affectation : l'approche classique pour un médoïde composé de plusieurs éléments est d'affecter un élément au noyau qui lui est le plus proche suivant un calcul de moyenne arithmétique des distances entre l'élément en question et les éléments du noyau. Or dans notre cas, pour un élément donné, il n'existera pas forcément un lien avec tous les éléments composant les différents noyaux. De ce fait, il faut déterminer si l'on veut affecter un élément sur une seule composante : la distance (i.e. moyenne arithmétique sur les liens existants), ou bien sur deux composantes : la distance et le nombre de liens (i.e. moyenne arithmétique sur le cardinal de la classe homogène).
- La phase de fusion : elle regroupe plusieurs classes homogènes. La classe résultante peut être elle-même considérée comme homogène d'après le critère de fusion ; cette classe sera ainsi un noyau.
- La classe résidu : la variante des centres manquants partitionne cette classe en différentes classes. L'approche naturelle est alors de déterminer, pour chaque classe, la classe homogène correspondante qui deviendra ainsi le noyau.

7.15 Conclusion

Les expérimentations pour notre algorithme Ω-means, menées à la fois sur notre corpus de référence et sur un échantillon de ce dernier, ont permis d'apporter les conclusions suivantes :

- La mesure de ressemblance `cosine` couplée avec la fonction de pondération `tf.idf` donne de bien meilleurs résultats que la distance Euclidienne ou encore que la distance de Manhattan. Des expérimentations avec d'autres distances auraient pu être réalisées. Cette conclusion est globalement en adéquation avec les résultats trouvés dans la littérature. Toutefois, cette mesure est supposée moins bien fonctionner sur des documents de petite taille.
- La méthode de détection automatique de *K* donne de très bons résultats. Pour notre corpus de référence, nous avons détecté 58 classes pour 57 théoriques et, pour l'échantillon, nous avons détecté le nombre exact de classes. Cette méthode, fondée sur le nombre de liens existant entre les différents éléments du corpus, a une dépendance limitée par rapport à ces liens. En effet, lors de l'indexation des SN et des unitermes générant une augmentation de 70 % du nombre de liens moyen, 52 classes ont été détectées.
- L'initialisation est une phase importante dans la plupart des algorithmes de partitionnement. Le critère d'initialisation défini par défaut supplante expérimentalement les autres que nous avons testés ; la partition initiale obtenue par la méthode détection de *K* ne permet pas d'obtenir de meilleurs résultats. Ce critère est propre à notre algorithme et ne peut être un moyen d'initialisation efficace pour la plupart des autres algorithmes. A noter, que nous nous sommes refusé à utiliser une autre méthode de partitionnement pour initialiser la nôtre pour des raisons de temps de calcul.
- La fonction de détection de classes homogènes est importante dans notre méthode puisqu'elle permet de définir un sous-ensemble fortement connecté permettant de choisir dans de bonnes conditions les nouveaux centres.
- La classe résidu qui regroupe tous les éléments inclassables, à chaque itération, demande une gestion toute particulière, et notamment sur des corpus de petite taille. L'approche qui consiste à partitionner cette classe pour en déterminer les éventuelles « sous-classes » (*cf.* la variante des centres manquants) s'est montrée plus performante que l'approche initiale. De plus, elle permet de converger plus rapidement si elle est couplée avec l'initialisation par défaut.
- Le coefficient d'homogénéité, qui est l'un des rares seuils de notre méthode, possède une valeur permettant d'obtenir les meilleurs résultats. Toutefois, l'incrément utilisé est de 0.1 et les résultats laissent penser qu'il existe un intervalle, et non une valeur unique, pour lequel les résultats obtenus sont très proches en termes de performance.
- En comparant les résultats obtenus sur l'échantillon et sur le corpus de référence, nous remarquons une dégradation des résultats sur le corpus, confirmant ainsi l'hypothèse qu'on ne peut étendre les performances d'une méthode obtenue sur un faible échantillon à un corpus de grande taille.

- La comparaison avec les méthodes usuelles sur l'échantillon montre que nous obtenons globalement de meilleurs résultats.

L'initialisation joue un rôle important dans la plupart des méthodes de partitionnement. Nous supposons que cette influence peut être atténuée par l'élargissement du noyau ; c'est-à-dire en considérant le noyau comme un ensemble d'éléments dont le cardinal est déterminé automatiquement. Cette conjecture énoncée dans le § 7.14, n'a pu être démontrée expérimentalement par manque de temps.

Les prétraitements effectués sur les documents, décrits dans le Chapitre 5, sont nécessaires et suffisants pour notre corpus de référence. Toutefois, d'autres traitements sont nécessaires suivant le corpus d'expérimentation. Pour un corpus regroupant des documents anglo-saxons, la permutation des mots dans les SN augmente la qualité des résultats. L'utilisation de dictionnaires de synonymes ou bien de thésaurus est un traitement qui peut s'avérer avantageux pour certains corpus. Enfin, la segmentation des documents est un élément à prendre en compte dans certains cas. Ce trait n'est pas abordé dans nos expériences car notre corpus regroupe principalement de « petits » documents équivalents à un paragraphe. Ce traitement n'aurait eu donc aucun effet sur celui-ci.

Chapitre 8

SearchXQ : un algorithme de navigation et de recherche par expansion de requête

Résumé

Certains systèmes de recherche d'information intègrent une méthode de classification pour présenter les éléments par ordre de pertinence de classes ou bien pour permettre à l'utilisateur de choisir parmi une liste de classes, sous forme d'ensembles de mots, pour une expansion de la requête.

Dans ce chapitre, nous utilisons l'algorithme de classification défini au chapitre précédent afin de l'intégrer dans des modèles de navigation classiques tels que le plan de classement (approche statique) ou bien l'expansion de requêtes. Dans ce dernier modèle, nous proposons deux approches différentes : une approche dynamique et une nouvelle approche « semi-dynamique » dénommée SearchXQ. Cette dernière approche est une combinaison de l'approche statique et dynamique et tend à pallier les inconvénients de celles-ci.

8.1 Introduction

La navigation sur le Web s'effectue généralement à travers le mode de requête itérative, mode le plus utilisé mais qui nécessite de l'utilisateur un effort dans sa quête d'informations. Toutefois, d'autres modes existent (*cf.* Chapitre 3) tels que le plan de classement ou bien l'expansion de requête automatique ou semi-automatique. Ces modes sont différents par les techniques mises en œuvre pour accéder à l'information, mais également par le type d'information donné, par l'utilisateur ou par le système, ou encore par le temps de réponse.

Notre objectif est d'aider l'utilisateur dans sa recherche d'information à travers le mode qui nous semble le meilleur compromis entre une réponse adaptée à la requête et le temps d'intervention de l'utilisateur. En effet, le mode requête nécessite un effort de l'utilisateur si ce dernier ne trouve pas les documents voulus avec sa requête d'origine. Pour combler ce type de lacune, engendrée par un taux de pertinence statique de chaque document, certains systèmes utilisent la classification pour améliorer l'ordonnancement des documents en réponse à une requête. Toutefois, ces systèmes, utilisés sur les retours de moteurs de recherche, ne sont guère exploitables car le temps de réponse est trop élevé : l'analyse et la classification de quelques centaines de documents sont coûteux en temps.

A contrario, les plans de classement ou répertoire ne nécessitent aucun effort de l'utilisateur hormis celui, mais non des moindres, de trouver la ou les catégories qui correspondent a son besoin. L'information fournie par ce mode n'est pas satisfaisante pour une requête précise : elle fournit les sites relatant les thématiques, une recherche sur ledit site était ensuite nécessaire. Toutefois, ce mode est rapide.
Les systèmes d'expansion de requêtes automatique ne font pas intervenir l'utilisateur après sa requête ; tout est « transparent » ; ce système repose sur la même hypothèse que celle des moteurs de recherche.

Ainsi, l'intervention de l'utilisateur est nécessaire pour une aide à la navigation « individualisée », mais celle-ci doit se faire dans la lisibilité et la facilité et dans un temps raisonnable. C'est ainsi que nous optons pour l'expansion de requête par thèmes (et par termes uniquement) : nous supposons de ce fait que l'ensemble de listes de mots est contraignant pour l'utilisateur.

Ce chapitre est composée de trois sections dont chacune est relative à une approche de l'aide à la navigation par présentation de thèmes : l'approche statique, l'approche dynamique et enfin l'approche semi-dynamique. Pour chacune de ces approches, les intérêts, les avantages et les inconvénients sont présentés.

8.2 Approche statique

8.2.1 Principe

La navigation par une approche statique se résume à choisir une liste de termes (i.e. d'étiquettes ou de catégories) dans une hiérarchie pré-calculée : il existe un certain « degré de déconnexion » de la réponse par rapport à la requête. En effet, les termes présentés à l'utilisateur, pour une requête donnée, sont pré-calculés en fonction du corpus global et non du sous-ensemble que représentent les documents en réponse.

Ainsi, une telle navigation peut être conçue de deux façons différentes :

- Une hiérarchie de termes (ou répertoire), c'est-à-dire que le corpus est présenté par thèmes (ou catégories). A chaque niveau de la hiérarchie, diverses informations sont fournies à l'utilisateur : une liste de sous-thèmes pour affiner le thème courant, et une liste de documents en réponse à la requête représentée par la combinaison de thèmes.
- Recherche par requête, c'est-à-dire qu'on présente les sous-catégories apparentées à une requête donnée ainsi qu'une liste de documents en réponse.

8.2.2 Internet

L'approche statique sur internet fait référence à des annuaires ou des répertoires. On pouvait penser que celle-ci allait être supplantée par les moteurs de recherche et était vouée à disparaître. Toutefois, les annuaires sont toujours présents dans le paysage de la recherche d'information et la plupart des moteurs de recherche intègrent aujourd'hui un répertoire. On peut citer entre autres Yahoo (www.yahoo.com), le précurseur de l'annuaire sur Internet, ou encore Google (www.google.com), qui a publié son répertoire bien après le lancement de son moteur de recherche. De plus, cette approche est d'autant plus d'actualité qu'elle fait l'objet d'un projet sous le nom de *Open Directory Project* (http://dmoz.org). A noter que le contenu du répertoire de Google est fondé sur *Open Directory*.

La plupart des répertoires arborent le même plan de classement fondé sur la classification de Dewey même si la classification d'Open Directory diffère.

Par exemple, pour la catégorie « droit français » (*cf.* Figure 8.1), Google présente comme dernières sous-catégories : « droit civil », « droit du travail », « avocats », etc. La classification d'Open Directory (*cf.* Figure 8.2) propose pour « legal information », des sous-catégories telles que « tax », « health », « aviation law », etc.

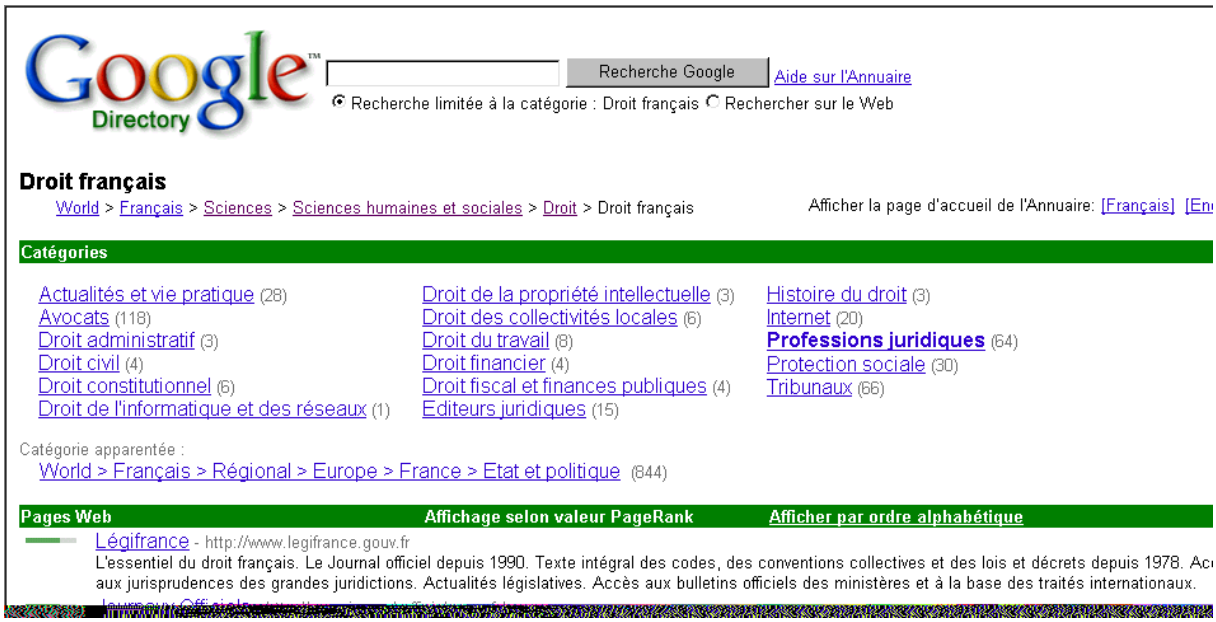

Figure 8.1 – Les sous-catégories de « Droit français » sur Google

Le principe des annuaires et des répertoires est donc de classer des sites Web dans une hiérarchie de catégories prédéfinies afin d'aider l'utilisateur dans sa démarche de recherche d'informations. Ainsi, la navigation se fait de « proche en proche », c'est-à-dire que pour chaque niveau de la hiérarchie, il faut identifier la catégorie qui se rapporte au besoin. Pour chaque catégorie, une liste de sites Web est proposée, et pour chacun d'entre eux une description est présentée. L'utilisateur n'a d'autre choix que de parcourir les sites.

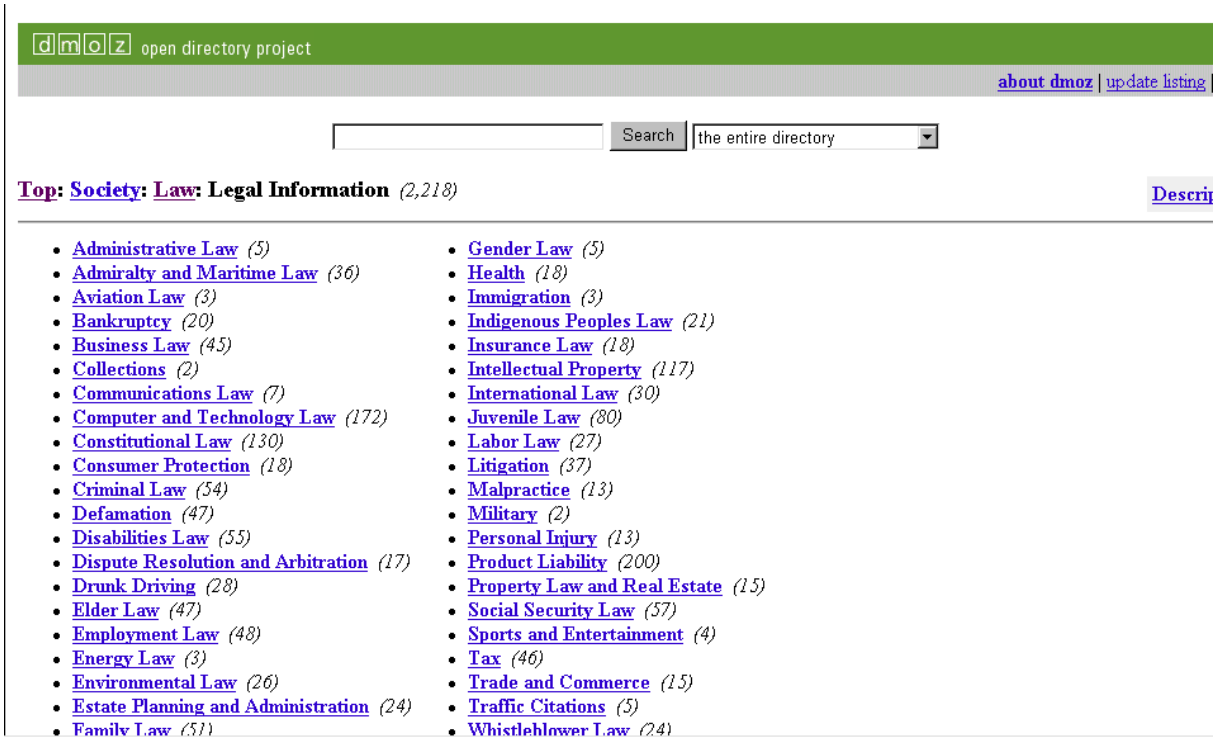

Figure 8.2 – Les sous-catégories de « *Legal information* » sur Open directory

L'inconvénient majeur des annuaires est le manque de recherche en profondeur du fait même de leur principe de classer des sites Web. En effet, la profondeur est limitée par le thème générique du site. Il ne peut y avoir plus de profondeur que si les pages d'un site Web sont classées indépendamment les unes des autres, ce qui représenterait un travail colossal à l'échelle du Web.

L'utilisation des répertoires a évolué ces dernières années. Dans certains moteurs, elle est intégrée dans la recherche par requête, c'est-à-dire que pour une requête donnée, des liens vers des catégories peuvent être proposés si celles-ci sont appropriées à la requête. Ceci peut être utile pour faire prendre conscience à l'utilisateur de la portée de sa requête.

L'utilisation d'un moteur de recherche dans un répertoire est également fréquente ; l'intérêt pour l'utilisateur est de limiter sa recherche dans une catégorie de la hiérarchie si ce dernier ne sait comment le faire avec un moteur. Cette approche répond partiellement à un problème sous-jacent : quels sont les termes à ajouter et/ou à retirer dans une requête pour étendre ou limiter la recherche ?

Ainsi, notre approche statique et celle des moteurs de recherche et des annuaires sur Internet sont différentes dans le sens où les annuaires classent des sites alors que nous classons les pages d'un site ; ce qui correspond en quelque sorte à une extension d'un annuaire. L'approche est d'autant plus différente que nos catégories ne sont pas prédéfinies mais déterminées automatiquement.

8.2.3 Construction d'un niveau de la hiérarchie

Dans cette section, nous présentons la construction d'un niveau i de la hiérarchie avec $i > 1$ (*cf.* Figure 8.3), le niveau 1 étant le partitionnement du corpus étudié au chapitre précédent. La construction d'un niveau de la hiérarchie nécessite d'en définir les principes, les hypothèses et les objectifs. Des exemples d'expérimentation, menés sur les meilleures partitions obtenues au chapitre précédent, seront présentés dans la section suivante.

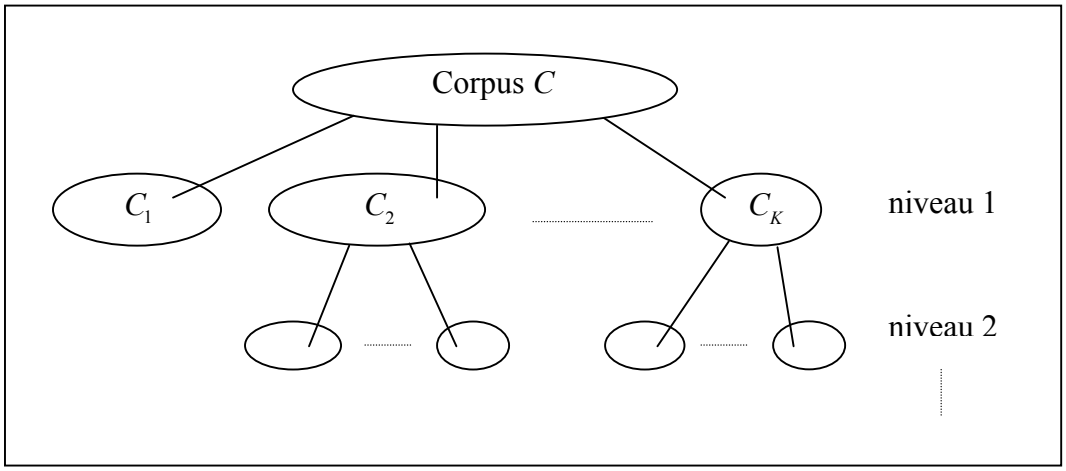

Figure 8.3 – Exemple de hiérarchie

Le principe de la construction d'un niveau *i* de la hiérarchie est simple puisqu'il est fondé sur notre méthode de classification qui est appliquée de façon récursive sur toutes les classes de chaque partition du niveau $i-1$; pour le niveau 2, il n'existe qu'une seule partition. Ainsi, nous pouvons reprendre les hypothèses définies au chapitre précédent.

Si le principe est simple, la mise en œuvre nécessite, quant à elle, des étapes supplémentaires autres à la simple application de l'algorithme. En effet, comme toute méthode hiérarchique, il faut mettre à jour la matrice à chaque étape.

Cette mise à jour nécessite une analyse partielle de chaque élément du corpus afin d'éliminer un certain nombre de liens entre les éléments de chaque classe. En effet, tous les éléments de chaque classe d'une partition sont connectés à plus de $100 \cdot \varepsilon$ % éléments de ladite classe, où ε est le coefficient d'homogénéité et vaut 0.6 par défaut, le centre de chaque classe étant connecté à tous les éléments. L'application de l'algorithme de classification aurait donc peu d'intérêt dans ces conditions. La mise à jour nécessite donc pour chaque classe de :

- détecter un ensemble de termes trop fréquents ;
- créer ou mettre à jour la matrice de liens et de distances.

La détection des termes trop fréquents d'une classe peut se faire de la même façon que celle décrite dans le Chapitre 5, c'est-à-dire qu'on élimine de la classe C_i tous les termes *t* pour lesquels $\text{Idf}(t) \geq \alpha |C_i|$. Les pré-traitements linguistiques sur les éléments de chaque classe ont déjà été effectués sur le corpus. Les hapax ne sont plus à gérer puisqu'ils ont été éliminés lors des pré-traitements statistiques du corpus.

L'objectif principal de la construction d'un niveau de la hiérarchie reste le même que dans le cas précédent : nous voulons retrouver le plus grand nombre d'étiquettes du corpus. Toutefois, cette tâche est d'autant plus difficile qu'à chaque niveau les partitions trouvées ont un différentiel plus ou moins important avec les partitions du corpus.

8.2.4 Construction de la hiérarchie

Nous avons présenté, dans la section précédente, la façon de construire un niveau de la hiérarchie. Dans ce paragraphe, nous nous attachons au dernier niveau de la hiérarchie et plus précisément à l'heuristique qui permet de définir la valeur du dernier niveau.

Dans les systèmes de classification hiérarchique descendante, le dernier niveau est atteint lorsque toutes les classes sont singletons, c'est-à-dire dans notre cas lorsque nous obtenons |*C*| classes. Une autre façon classique est de déterminer le nombre de classes *K* que l'on veut obtenir pour déterminer le dernier niveau équivalent alors à $C - K$.
Pour les deux cas énoncés ci-dessus, il ne peut y avoir de solution car, pour l'un, il n'est pas intéressant au niveau de la navigation d'aboutir à un seul document et, pour l'autre, il nous est difficile de déterminer la valeur de *K*.

En faisant référence à notre corpus de référence, la hiérarchie s'arrête au cinquième niveau, le niveau des codes y compris. De plus, un de nos centres d'intérêt est d'aboutir à une solution en « quelques clics », c'est-à-dire en combinant « quelques syntagmes nominaux ». Dans ces conditions, notre dernier niveau de la hiérarchie n'excédera pas le cinquième niveau.

8.2.5 Expérimentations sur le niveau 2

Dans ce paragraphe, nous nous attachons au comportement de l'algorithme du chapitre précédent sur les meilleures partitions obtenues expérimentalement. Nous utilisons les notations établies au § 7.2 du Chapitre 7.

Pour nos expérimentations, nous utiliserons l'algorithme avec les paramètres suivants :

- Cos+Tf.Idf ;
- l'initialisation par défaut ;
- la détection automatique de K ;
- $\varepsilon = 0.6$;
- la variante des centres manquants.

Ces choix reposent sur les performances obtenues sur le corpus de référence.

En utilisant la partition obtenue à partir du corpus avec les paramètres décrits ci-dessus, nous ne pouvons pas évaluer en totalité toutes les classes suivant la hiérarchie du corpus car certaines classes sont « atypiques ». A noter que ces classes ne sont pas spécifiques aux paramètres énoncés mais elles se retrouvent dans toutes les partitions trouvées au chapitre précédent.

Ces classes atypiques ont les caractéristiques suivantes :

1. la classe regroupe plusieurs codes en quasi-totalité ou en totalité ainsi qu'un petit nombre d'éléments d'autres codes : taux de précision faible et taux de rappel élevé.
2. La classe regroupe très majoritairement une partie d'un code : taux de précision élevé et taux de rappel faibles.
3. La classe regroupe plusieurs parties de codes différents : taux de précision et de rappel faibles.
4. La classe regroupe un code en quasi-totalité ou en totalité mais il est « noyé » dans un ensemble d'éléments d'autres codes : taux de précision faible et taux de rappel élevé.

Dans les paragraphes suivants, nous allons nous intéresser principalement à ces classes atypiques. Notons toutefois que ces classes ne constituent pas la majorité des classes.

Cas n°1

Pour ce cas, nous avons détecté une classe qui regroupe plusieurs codes en totalité : ces codes sont relatifs à la déontologie de différentes professions telle que la profession paramédicale des sages-femmes (les acronymes des codes sont définis en Annexe A).

Code		#Eléments
Intitulé	#	
CTRAVAI	4845	3
CSANPU	4720	14
CRURAL	4911	2
CPROCPE	2404	1
CDARCHI	**49**	**49**
CDCHIRD	**87**	**87**
CDSAGES	**69**	**69**
CDPOLIC	**20**	**20**

Tableau 8.1 – Partition 'Cos+*K*58+déf.' : composition de la classe n°47 ayant pour étiquette 'code déontologie'

Dans le Tableau 8.1, nous présentons la composition de cette classe qui regroupe quatre codes en totalité ainsi que des éléments de différents codes constituant uniquement 8% de la classe. L'objectif recherché est de retrouver les étiquettes adéquates pour les codes présents en totalité et, dans une moindre importance, les étiquettes des autres codes. On suppose qu'il est plus facile de retrouver la structure des codes entiers que celle des petits sous-ensembles des autres codes.

Dans le Tableau 8.2, nous présentons la partition trouvée suivant les étiquettes trouvées pour chaque classe, le nombre d'éléments par étiquettes, les codes et le nombre d'éléments se rattachant à chaque classe. Pour cette classe n°47, onze « sous-classes » ont été détectées pour huit codes représentés. Parmi ces sous-classes, nous retrouvons trois des quatre étiquettes des codes entiers ; le quatrième code est quant à lui divisé en sous-classes au nombre de cinq. La structure de ce code est simple puisque le niveau deux est composé de 7 titres et qu'il n'existe pas de niveaux inférieurs. Les étiquettes suivantes trouvées pour ce code sont correctes : « devoir confraternité », « devoir général chirurgien-dentiste », « exercice profession », « devoir chirurgien-dentiste ». Seule l'étiquette « cabinet secondaire » n'est pas correcte. La différence entre les sept sous-classes théoriques et les cinq trouvées s'explique par le regroupement dans une même sous-classe des différents « devoirs chirurgiens dentistes ». En effet, parmi les 7 titres, trois concernent les devoirs des chirurgiens-dentistes :

- envers les patients (13 éléments) ;
- en matière de médecine sociale (13 éléments) ;
- envers les membres des professions de santé (2 éléments).

Etiquette			Code	
Numéro	Intitulé	#	Intitulé	#
1	Code déontologie architecte	47	CDARCHI	47
2	Devoir chirurgien-dentiste	29	CDCHIRD	29
3	Code déontologie police national	20	CDPOLIC	20
4	Continuité soin Code santé	3 3	CSANPU CDSAGES CPROCPE	3 2 1
5	Exercice profession	17	CDCHIRD CDSAGES CSANPU CDARCHI CRURAL	12 3 2 2 1
6	cabinet secondaire	5	CDCHIRD CDSAGES	4 1
7	devoir général chirurgien-dentiste	29	CDCHIRD	29
8	Devoir confraternité	15	CDCHIRD CDSAGES	10 5
9	-	-	CACTSOC	1
10	code déontologie sage-femme	56	CDSAGES	56
11	décret conseil conseil état	13 13	CSANPU CTRAVAI CRURAL	9 3 1

Tableau 8.2 – Partition trouvée avec la classe n°47.

Tous ces éléments étant regroupés dans une même classe, nous supposons que le discernement entre les différents « devoirs » se fera hypothétiquement au niveau supérieur. A noter que pour la ligne n°5, le regroupement s'est fait sur deux étiquettes différentes : la somme des # par code est supérieure à la # de la classe.

Cas n°2

Les classes qui répondent à ce cas sont difficiles à définir car la notion de « partie de classe » n'est pas assez précise : une approche satisfaisante serait de définir des intervalles pour le taux de rappel et de précision. Par conséquent, le nombre de classes qui représente ce cas est difficile à établir.

L'objectif recherché pour ces classes est de retrouver une partie de la hiérarchie théorique. La hiérarchie trouvée est partiellement liée à l'ensemble des éléments manquants.

Pour illustrer ce cas, nous avons choisis la classe n°28 qui regroupe environ 72 % du code forestier avec un taux de précision de 0.91.

Code		#Eléments
Intitulé	#	
CFOREST	1149	857
CRURAL	4911	36
CSANPU	4720	21
CGIMP	4901	7

Tableau 8.3 - Partition 'Cos+K58+déf.' : extrait de la composition de la classe n°28

Dans le Tableau 8.3, nous présentons un extrait de la classe n°28 qui regroupe principalement le code forestier et treize autres codes dont six sont représentés par un hapax.

Etiquette			Code	
Numéro	Intitulé	#	Intitulé	#
1	Code rural Propriété forestier	34 33	CRURAL CFOREST CURBANI	30 23 2
2	Bois particulier Police bois Forêt général	26 16 16	CFOREST	38
3	Forêt protection	71	CFOREST	71
4	Mis valeur ressource ligneux	92	CFOREST	94
5	Office national forêt	96	CFOREST	96
6	Disposition particulier département outre-mer	101	CFOREST	101

Tableau 8.4 – Extrait de la partition trouvée avec la classe n°28.

Le Tableau 8.4 présente un extrait des 14 sous-classes détectées. Bien que le nombre de sous-classes détectées corresponde au nombre de différents codes présents dans la classe, la partition trouvée n'est en rien une répartition de ces codes dans les différentes classes. Toutefois, les codes CRURAL et CSANPU sont regroupés en quasi-totalité dans des classes différentes en présence d'un certain nombre d'éléments de CFOREST (*cf.* étiquette 1).
Le code forestier possède cinq livres dont seuls trois sont retrouvés sur la base des étiquettes (*cf.* étiquette 2,3 et 4). Pour les deux autres livres, une partie des éléments étant manquante, des parties du niveau supérieur sont retrouvées (*cf.* étiquette 5 et 6). A noter que dans cette partition, l'étiquette 5 sur « l'office national des forêts » regroupe environ 100 éléments, bien plus que dans la hiérarchie théorique. Parmi les 14 étiquettes, deux ne font pas référence à des étiquettes du niveau 2 ou du niveau supérieur.

Des expérimentations menées sur la classe n°23, regroupant majoritairement le code de l'aviation civile, montrent des résultats similaires, c'est-à-dire que l'on retrouve une partie des étiquettes du niveau 2 et des étiquettes des niveaux supérieurs.

Cas n°3

Les classes du cas n°3 regroupent différentes parties de codes différents sans qu'aucun d'eux ne soit majoritaire au niveau de la classe. Ce cas est assez rare dans notre partition et correspond généralement pour la classe à une étiquette incorrecte. L'objectif recherché pour ce cas est de retrouver des étiquettes de la hiérarchie à des niveaux différents ou bien de regrouper les codes dans des classes différentes.

Code		#Eléments
Intitulé	#	
CURBANI	1493	680
CTRAVAI	4845	390
CGIMP	4901	269
CCONSTR	2295	94
CSECSOC	6500	45

Tableau 8.5 - Partition 'Cos+K58+déf.' : extrait de la composition de la classe n°53.

Le Tableau 8.5 présente un extrait de la classe n°53 composée de 28 codes différents dont huit représentés par un hapax. Dans cette classe, aucun code n'a plus de la moitié de ses éléments présents et aucun n'est majoritaire.

Etiquette			Code	
Numéro	Intitulé	#	Intitulé	#
1	Code électoral Election député	30 30	CELECTO	30
2	Plan occupation	80	CURBANI	117
3	Code général impôt	165	CGIMP	166
4	Opération aménagement	157	CURBANI CRURAL	143 17
5	Département outre-mer Réglementation travail	74 57	CTRAVAI CURBANI CGIMP …	58 54 7
6	Mode utilisation sol Régime général	192 66	CURBANI	230
7	Code sécurité	33	CSECSOC CTRAVAI	31 2

Tableau 8.6 - Extrait de la partition trouvée avec la classe n°53.

Pour la classe n°53, 19 sous-classes ont été détectées ; le Tableau 8.6 présente les caractéristiques de quelques sous-classes dont l'étiquette est valable. Dans ce tableau, certaines étiquettes font référence à celles du niveau 2 de la hiérarchie théorique des codes (*cf.*

étiquette 6) mais aussi à celles de niveaux supérieurs telle que l'étiquette 4 (niveau 3) ou encore l'étiquette 2 (niveau 4). Les éléments de certains codes sont regroupés en totalité ou en quasi-totalité au sein d'une classe comme par exemple les 30 éléments de CELECTO (non présents dans le Tableau 8.5).

Cas n°4

Il existe plusieurs classes concernant ce cas dans notre partition. L'objectif recherché est de détecter le code noyé dans la classe, soit en réunissant tous les éléments de ce code dans une sous-classe et en déterminant la structure du niveau 2 à l'étape suivante, soit en déterminant la structure à l'étape courante.

Code		#Eléments
Intitulé	#	
CDVIMAR	**84**	**84**
CTRAVAI	4845	26
CPORMAR	421	12
CSECSOC	6500	9
CDYANES	477	3
CTRAVMA	145	3
CORGJUD	951	3
CSERVNA	555	3
CENVIRO	971	2

Tableau 8.7 - Partition 'Cos+K58+déf.' : extrait de la composition de la classe n°27

Pour illustrer ce cas, nous avons pris le code CDVIMAR –*cf.* Tableau 8.7–, composé de 84 éléments, intégrés dans une classe comprenant 16 codes différents dont sept représentés par un seul élément. Ce code possède un taux de précision de 0.55 pour un taux de rappel de 1.00.

Dans le Tableau 8.8 est présenté un extrait de la partition trouvée ; le nombre de sous-classes détectées est de 15. La structure du code CDVIMAR est simple puisque le niveau 2 est composé de 4 titres et l'un d'entre eux est divisé en 5 chapitres (niveau 3). La structure trouvée pour ce code met au même niveau les divisions et les subdivisions. Nous trouvons ainsi des étiquettes de niveau 2 et de niveau 3. Une subdivision du code est en présence d'autres éléments qui ne permettent pas de retrouver l'étiquette à cette étape. Une autre est quant à elle divisée en plusieurs sous-classes et l'étiquette ne pourra pas être retrouvée. L'influence des autres codes est mesurée dans cet exemple au regard de la répartition de ces derniers dans les différentes classes.

Etiquette				Codes	
Numéro	Intitulé	validité	#	Intitulé	#
1	Perte navire Accident navigation	√	9 9	CDVIMAR	9
2	Enquête préliminaire	×	3	CDVIMAR	3
3	Police intérieur navire	√	21	CDVIMAR	21
4	Décret conseil état	×	27	CTRAVAI CPORMAR CSECSOC …	15 6 5
5	Police navigation	√	12	CDVIMAR	12
6	Tribunal maritime commercial	√	13	CDVIMAR CORGJUD CTRAVAI	11 3 1
7	Présent loi Disposition générale	× √	10 7	CDVIMAR CTRAVAI	9 1
8	Cas force majeur	×	4	CDVIMAR CSECSOC	3 1
9	Administrateur affaire	×	11	CDVIMAR CTRAVAI …	8 2

Tableau 8.8 - Extrait de la partition trouvée avec la classe n°27.

Code		#Eléments
Intitulé	#	
CTRAVAI	4845	89
CSECSOC	6500	68
CRURAL	4911	61
CSANPU	4720	49
CARTISA	**45**	**45**
CGCTERR	3586	35
CCOMMER	1915	23
CGIMP	4901	21
CPROCPE	2404	19
CORGJUD	951	10

Tableau 8.9 - Partition 'Cos+K58+déf.' : composition de la classe n°56

Dans l'exemple précédent, le code à retrouver constituait le plus grand ensemble de la classe malgré un taux de précision faible. Dans l'exemple que nous prenons à présent, le code à retrouver (CARTISA) est composé de 45 éléments et il n'est pas le code le plus représentatif de la classe : son taux de précision est de 0.09 et son taux de rappel de 1.00. La classe est composée de 34 codes différents dont 10 représentés par un seul élément pour un total de 495

éléments. Les codes les plus représentés dans cette classe, hormis CARTISA, sont de grande taille mais l'ensemble d'éléments présents dans celle-ci est de petite taille.

Le nombre de classes détectées est de 9 ; le code CARTISA est présent dans cinq d'entre elles. Une classe regroupe 31 éléments ($P = 0.94$ et $Q = 0.69$), une autre regroupe 5 éléments ($P = 0.45$ et $Q = 0.11$). Les autres éléments sont noyés dans différentes classes. L'interaction des codes est plus importante dans cet exemple même si 69 % de CARTISA est regroupé dans une classe. Les codes les plus représentatifs ont le même type de répartition en ce sens qu'une classe regroupe une partie des éléments et le reste est distribué dans des classes composées d'éléments de différents codes.

Nous avons vu jusqu'à présent des cas de classes atypiques uniquement ; ces classes ont toutefois donné des résultats satisfaisants par rapport aux objectifs définis. Nous allons à présent expérimenter le cas idéal c'est-à-dire le cas d'une classe composée d'un code unique et dans sa totalité ($P = 1$ et $Q = 1$).

Cas n°5

Ce cas est idéal car les conditions d'expérimentation sont identiques à celles de la classe théorique, autrement dit à celles du code. L'objectif est donc, pour ce cas, de retrouver la hiérarchie théorique.

Etiquette				#Eléments
Numéro	Intitulé	Validité	#	
1	exercice médecine contrôle	√	4	4
2	devoir général médecin	√	26	26
3	Rapport médecin Profession santé	√	10 10	10
4	Mode exercice Règle commun	√	13 13	13
5	Exercice clientèle	√	9	9
6	Exercice salarié médecine	√	5	5
7	Conseil ordre	×	2	2
8	Conseil national Condition exercice	×	3 2	4
9	Médecin traitant	×	3	3

Tableau 8.10 - Extrait de la partition trouvée avec la classe n°5

Nous avons pris comme exemple, le code CDMEDIC (classe n°5) pour nos expérimentations ; ce code contient 114 éléments. La structure de ce code est composée de 5 divisions de niveaux 2 et la quatrième est elle-même scindée en 5 subdivisions (niveau 3).

Pour cette classe, la valeur du nombre de sous-classes a été déterminée automatiquement à 19, ce qui est supérieur à la valeur théorique. Dans le Tableau 8.10, nous présentons un extrait de la partition résultante. Nous remarquons que deux des étiquettes de niveau 2 sont trouvées ainsi que quatre étiquettes de niveau 3. Il n'y a donc pas de distinction entre le niveau 2 et 3 ; tout est regroupé au même niveau. Le nombre élevé de la valeur *K* entraîne la constitution de sous-classes de petite taille : 1 classe contient un hapax et 8 classes ne contiennent que 2 éléments. Toutefois, les résultats restent intéressants puisqu'un nombre non négligeable d'étiquettes est trouvé. Une expérimentation sur cette classe pour une valeur de *K* fixée à 5 ne permet pas de récupérer les autres étiquettes ; les résultats sont même en deçà en termes d'étiquettes trouvées.

En conclusion, nous avons déterminé le processus de création d'une hiérarchie sur *n* niveaux ; cette hiérarchie est fondée sur notre méthode de classification définie au chapitre précédent.

Les expérimentations menées sur le niveau 2 montrent des comportements différents en ce qui concerne les subdivisions trouvées. En comparant la structure théorique à celle obtenue pour certaines classes, nous constatons que dans certains cas :

- un niveau supplémentaire sera nécessaire pour retrouver la structure du niveau 2 ;
- un niveau sera omis et nous retrouverons ainsi la structure de niveau 3 au niveau inférieur.

8.3 Approche dynamique

8.3.1 Principe

La navigation par une approche dynamique consiste à choisir un terme parmi une liste calculée dynamiquement, c'est-à-dire que la liste de termes est fonction de la requête. Le corpus de référence n'est plus le corpus dans sa totalité mais un sous-ensemble des éléments retournés en réponse à une requête, dont le cardinal est déterminé empiriquement. Cette liste dynamique est calculée à chaque étape du processus itératif.

8.3.2 Méthodologie

Dans une approche dynamique, les problématiques sont identiques à une approche statique mais les contraintes diffèrent. Dans les deux cas, la contrainte principale est celle du temps : il faut fournir une réponse le plus rapidement possible. Or, les étapes de pré-traitements que l'on effectue dans une approche statique sont maintenant effectuées en temps réel : par exemple, la construction de la matrice de similarité.

Une autre contrainte est la mise en mémoire des données. En effet, pour que le système ne soit pas pénalisé, les données doivent bien évidemment être intégralement mises en mémoire.

Sous ces deux contraintes, la taille du corpus de référence doit être limitée, c'est-à-dire qu'il faut prendre les *n* premiers éléments retournés par une requête donnée pour constituer ce corpus, avec *n* fixé de façon empirique.

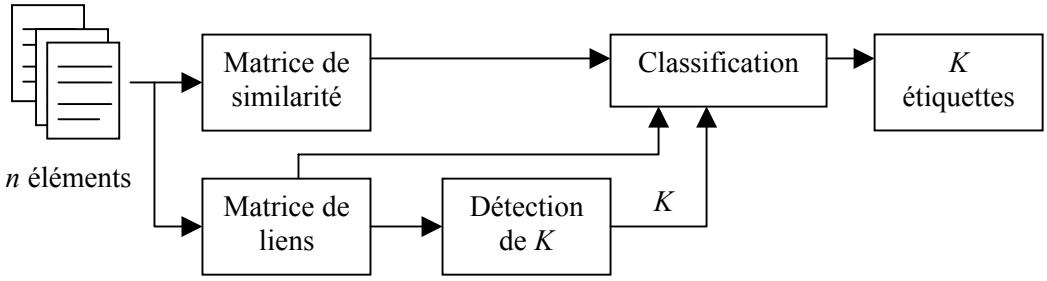

Figure 8.4 – Méthodologie de l'approche dynamique

Le choix des *n* premiers éléments peut se faire à partir du retour d'un moteur de recherche qui trie les réponses suivant un critère de pertinence donné.

Le processus de création de la liste de termes à partir de l'ensemble d'éléments – *cf*. Figure 8.4– est une suite d'étapes décrites ci-après :

- indexation des éléments pour en extraire l'ensemble des SN. Cette étape, décrite dans le Chapitre 5, est composée de filtrages statistiques et morpho-syntaxiques. Dans notre cas, les éléments sont connus et peuvent donc être indexés en pré-traitement. De ce fait, cette étape se résume alors à charger les SN de chaque élément tout en éliminant ceux trop fréquents.
- Création de la matrice de similarité et de la matrice de liens simultanément à partir de l'étape précédente. Ces deux matrices sont stockées en mémoire sous forme d'automates. Cette étape est la plus critique en temps de calcul.
- Détection automatique de *K* à partir de la matrice de liens et choix des *K* centres.
- Classification des *n* éléments en *K* classes en utilisant les paramètres par défaut. Cette étape est également critique en temps de calcul. En effet, la classification doit être très rapide, c'est-à-dire qu'une itération de l'algorithme doit s'exécuter rapidement et que la convergence doit être atteinte en quelques itérations seulement. Dans le chapitre précédent, nous avons remarqué que l'algorithme naïf permettait généralement une convergence rapide en moins de cinq itérations. Toutefois, notre algorithme converge rapidement sur des corpus de petites tailles.
- Détermination des étiquettes à partir des *K* classes trouvées.

8.3.3 Expérimentations

L'objectif de ces expérimentations est de déterminer dans quelle mesure cette méthode est applicable, et de mesurer la qualité des résultats.

Ainsi, nous avons pris comme exemple de requête le mot « déontologie ». Ce mot a été choisi car il regroupe plusieurs codes en totalité. A noter que ce mot n'est pas aussi

fréquemment posé que le mot « article » par exemple. L'objectif en terme de résultats est de distinguer les différents codes relatifs à la déontologie. Pour cette requête, nous retrouvons 460 éléments en réponse dont une partie est présentée dans le Tableau 8.11. Nous allons traiter la totalité des éléments.

Code		#Eléments
Intitulé	#	
CDARCHI	49	49
CDCHIRD	87	87
CDMEDIC	114	114
CDPOLIC	20	20
CDSAGES	69	69
CDVETER	54	54
CMONFIN	1299	3
CPOSTES	708	3
CTRAVAI	4845	3
CSANPU	4720	36
CSECSOC	6500	15

Tableau 8.11 – Extrait des éléments retrouvés pour la requête « déontologie ».

En appliquant notre processus, nous constatons que le temps de création des deux automates est de l'ordre de deux secondes. Le temps de détection des centres et de classification est très inférieur à la seconde. Le processus complet prend moins de cinq secondes, ce qui est « acceptable ». Pour cet ensemble d'éléments, nous avons trouvé une valeur de *K* égale à 10 pour 6 codes relatifs à la déontologie et plusieurs autres codes en faible quantité. Les résultats de ce processus sont présentés dans le Tableau 8.12. Nous constatons que les différents codes relatifs à la déontologie sont retrouvés, généralement en totalité. Le symbole « √/× » de ce tableau est synonyme de notre incapacité, en tant que « non-spécialiste juridique », à juger de la pertinence de l'étiquette pour cette requête. Les autres codes sont réunis entre eux : les codes en faible quantité gravitent autour de ceux qui sont les plus importants en terme de nombre d'éléments présents. Ces résultats, bien que difficilement comparables à ceux du chapitre précédent –du moins en ce qui concerne la classe avec l'étiquette « code déontologie »–, permettent de retrouver les codes avec un taux de précision et de rappel proche ou égal à 1.

Etiquette				Codes	
Numéro	Intitulé	Validité	#	Intitulé	#
1	code déontologie architecte	√	49	CDARCHI	49
2	code déontologie chirurgien-dentiste	√	87	CDCHIRD	87
3	code santé	√/×	26	CSANPU CDMEDIC	23 2
4	code déontologie médical	√	112	CDMEDIC	112
5	décret conseil état	×	24	CSECSOC CSANPU …	14 12 …
6	médecin travail	√/×	4	CTRAVAI CSECSOC	4 3
7	code déontologie police national	√	20	CDPOLIC	20
8	code déontologie sage-femme	√	68	CDSAGES	68
9	code déontologie vétérinaire	√	54	CDVETER	54
10	conseil supérieur	√/×	4	CPOSTES …	3

Tableau 8.12 – Partition trouvée pour la requête « déontologie ».

D'autres expérimentations ont été menées avec différentes requêtes. Parmi les requêtes présentes dans les fichiers de `log` et posées sur les codes, nous avons choisi la requête « contrat travail » qui génère environ 1700 éléments en réponse. Lors du processus de création de la liste de termes, nous constatons que le temps pour créer les deux automates est en dehors de la « limite acceptable » d'attente d'une réponse. En effet, la création de ces deux automates est de l'ordre de la minute. Du point de vue du temps, des requêtes générant quelques centaines d'éléments en réponse permettent à notre processus de créer la liste de termes dans un temps « acceptable » de quelques secondes. Du point de vue de la qualité des résultats, une partie des termes de référence sont retrouvés quand la possibilité se présente, c'est-à-dire qu'un certain nombre d'éléments d'une sous-catégorie sont présents dans la liste des éléments retrouvés. Pour la requête « aérodrome », livre 2 du code de l'aviation, nous retrouvons, entre autres, les cinq sous-thématiques.

En conclusion, cette approche peut donner des résultats satisfaisants dans un temps acceptable de quelques secondes. Toutefois, l'objectif serait de répondre en moins d'une seconde. De plus, certaines conditions doivent être prises en compte pour appliquer cette méthode :

- Le nombre d'éléments pour la création de la liste doit être limité dans un souci de temps de réponse : la limite peut être fixée à 500.
- Un nombre minimum d'éléments est nécessaire pour que cette méthode soit utile : Il faut au minimum plus de 10 réponses (valeur correspondant généralement à une page de réponses).

8.4 SearchXQ

8.4.1 Principe

Nous venons de voir, dans les sections précédentes, deux approches antinomiques pour la navigation. Dans cette section, nous proposons une nouvelle approche statistique qui combine à la fois une approche statique et une approche que nous pouvons qualifier de semi-dynamique.

La section 8.2 présente une approche classique de la navigation statique, c'est-à-dire d'une navigation à travers un plan de classement. La rapidité de la réponse à une requête est au détriment de la personnalisation de cette réponse, d'où une certaine déconnexion de la réponse par rapport à la requête. Cet inconvénient est totalement compensé par l'approche dynamique, qui consiste à appliquer une méthode de classification sur l'ensemble ou un sous-ensemble pertinent des éléments retournés pour une requête donnée. L'inconvénient est le temps de réponse qui nécessite généralement de prendre un sous-ensemble des éléments retournés. Ainsi, l'objectif est de répondre rapidement à une requête tout en donnant une liste de termes en adéquation avec la requête.

Notre approche est d'utiliser le plan de classement pré-calculé ou plus précisément la partie du plan contenant les éléments en réponse à une requête, afin de déterminer les termes en adéquation avec cette requête. L'intérêt de l'utilisation d'un tel plan de classement réside dans l'organisation de son corpus : les thèmes généraux sont extraits au premier niveau de la classification puis des sous-thèmes pour les niveaux supérieurs. Ainsi, pour une requête donnée, nous pouvons déterminer un ensemble de termes dont chacun est relatif à un thème ou à un sous-thème qui, de plus, est le plus représentatif du sous-ensemble d'éléments correspondant au thème ou au sous-thème.
L'avantage de cette approche est bien évidemment d'utiliser un plan de classement pré-calculé, réduisant ainsi le temps de réponse à la détermination des classes et des sous-classes contenant les éléments en réponse, et à partir de ces dernières à la détermination des termes qui seront proposés.
Bien qu'utilisant un plan de classement pré-calculé, les termes proposés ne sont pas pour autant déconnectés de la requête et ce, par un choix des termes les plus représentatifs dans chaque sous-ensemble des classes.

Dans le paragraphe suivant, nous développons la notion du choix des classes pour une requête donnée ainsi que celle des termes les plus représentatifs des sous-ensembles.

8.4.2 Algorithme

Dans le § précédent, nous avons développé notre approche de la navigation « semi-dynamique » en utilisant un plan de classement. Cette approche est dans un premier temps fondée sur un choix de classes et de sous-classes pour une requête donnée.

Pour illustrer le choix des classes et des sous-classes, nous avons présenté dans le Tableau 8.13 différentes combinaisons que nous sommes susceptibles de trouver pour une requête donnée : un sous-ensemble de niveau 1 et 2 (classe C_i), une classe de niveau 2 (classe C_j), une classe de niveau 1 (classe C_k), une classe en quasi totalité de niveau 1 et/ou des classes en quasi-totalité/totalité de niveau 2 (classe C_l).

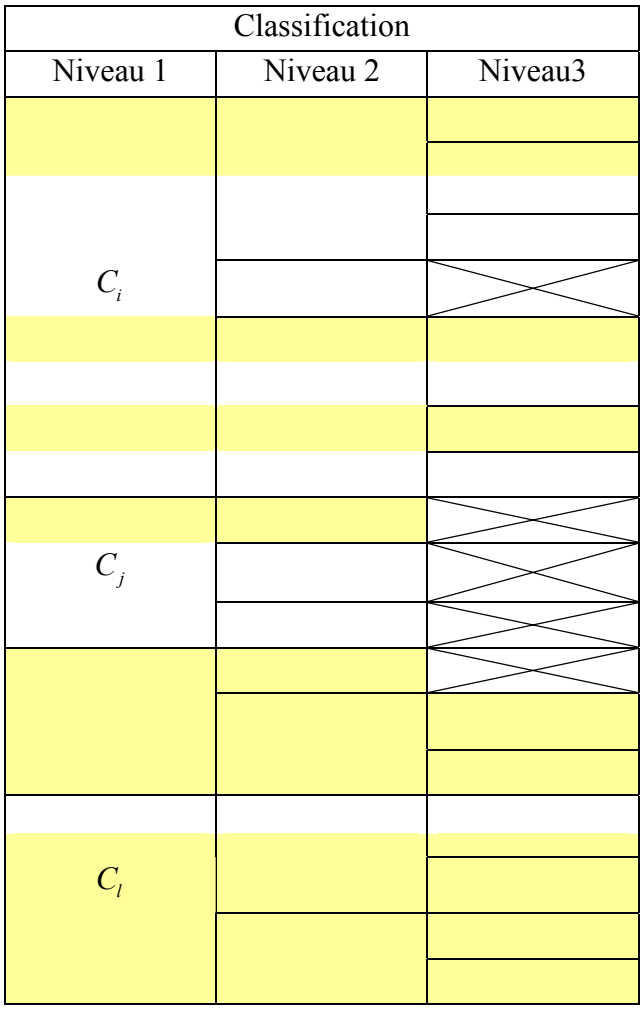

Tableau 8.13 – Exemple de sous-ensembles d'éléments trouvés (en jaune), dans la classification pré-calculée, pour une requête donnée

Au regard du Tableau 8.13, certains termes seront choisis au niveau 1 car la totalité ou la quasi-totalité des éléments d'une classe sont trouvés : par exemple, la classe C_k ou la classe

C_l. D'autres termes seront choisis au niveau 2 telle que la classe C_j ou encore la classe C_i. Pour cette dernière, deux termes sont alors proposés et correspondent aux deux classes de niveaux 2 atteintes.

L'intérêt d'utiliser plusieurs niveaux est de ne pas tomber dans des trous locaux en ce qui concerne le choix des thèmes. En effet, l'utilisation d'un même niveau à chaque itération ne permet pas d'affiner le thème courant.

Le choix du niveau étant effectué, il faut ensuite choisir, pour chaque sous-ensemble de classe, l'étiquette qui le représentera le plus. Pour cela, il faut choisir le(s) terme(s) fortement présent(s) en termes d'occurrences mais aussi faiblement présent(s) dans les autres éléments de la classe.

8.4.3 Exemple

Dans cette section, nous présentons un extrait des résultats d'une requête (*cf.* Tableau 8.14) : cette requête a déjà été utilisée dans le Chapitre 2 pour l'interrogation de différents moteurs de recherche.

Etiquettes		
Numéro	Intitulé	Validité
1	Accident du travail	√
2	Contrat de travail	√
3	Contrat d'apprentissage	√
4	Convention relative au travail	×
5	Réglementation du travail	√
6	Centre de formation	√

Tableau 8.14 - Extrait des étiquettes trouvées pour la requête : contrat travail

Les résultats trouvés dans le Tableau 8.14 ne nous permettent pas de conclure sur la pertinence des termes pour cette requête. Toutefois, nous constatons qu'un nombre non négligeable des termes proposés sont des termes juridiques. Nous ne pouvons pas comparer les résultats avec d'autres moteurs sachant que le corpus de base est différent.

8.5 Conclusion

Dans ce chapitre, nous avons adapté notre algorithme de classification Ω-means pour différentes approches de navigation : statique, dynamique et semi-dynamique. Nous constatons que ces trois modes sont tous exploitables avec des contraintes particulières pour certaines et notamment l'approche dynamique, coûteuse en temps de calcul. L'approche statique, bien que son utilisation diffère de celle des moteurs de recherche, se rapproche

surtout de l'utilisation que l'on peut en faire dans des domaines particuliers tels que la médecine ou le droit. Enfin, la dernière approche permet de retrouver une liste de thèmes proches de la requête dans un temps raisonnable. Bien que les résultats trouvés pour chaque approche soient globalement intéressants (résultats proches de la classification théorique ou termes trouvés juridiques), seuls les utilisateurs pourront juger de l'efficacité de ces approches.

Chapitre 9

Conclusions et perspectives

9.1 Contexte et objet de la thèse

De nos jours, il est fréquent de manquer d'inspiration, face au bruit généré par les moteurs de recherche, pour formuler ou reformuler une requête afin aboutir à une liste de documents attendus. Dans ce cas, l'approche de la plupart des utilisateurs est de manipuler les documents ou les résumés de ces documents dans l'espoir de modifier la requête de façon déterminante, soit en ajoutant des termes, soit en éliminant des termes suivant une combinaison booléenne.

Cette thèse s'inscrit dans la perspective d'améliorer les conditions de l'utilisateur face à l'absence de requête pertinente. Ainsi, nous avons présenté une méthode d'aide à la navigation qui, au travers d'une nouvelle méthode de classification non-supervisée et de type partitionnement, permet à l'utilisateur d'orienter facilement et rapidement sa recherche moyennant un effort modéré.

La classification est un domaine qui a été largement exploré et dont le regain d'intérêt est manifeste notamment depuis l'émergence de l'Internet grand public. Il existe plusieurs grandes familles d'algorithmes de classification dont la taxonomie est parfois discutée. Au regard des méthodes existantes, avec les avantages et les inconvénients que chacune d'elles présente, la catégorie d'algorithmes de partitionnement nous est apparue adéquate par rapport à l'objectif recherché dans cette thèse. L'objectif de cette dernière a été d'élaborer une nouvelle variante de k-means dont le but est de classer un grand nombre de documents avant une étape d'étiquetage de classes permettant d'alimenter notre moteur de recherche. L'élaboration d'un corpus de référence nous a permis grâce à l'aide de considérations d'experts juridiques d'être en adéquation avec le domaine d'application, et de nous contenter de ce corpus sans avoir recours à des expérimentations faisant intervenir des utilisateurs.

9.2 Buts atteints

9.2.1 Une variante de k-means

L'étude des méthodes de classification nous a permis de dégager le type de méthode qui correspond le mieux à nos attentes. Le type k-means présente plusieurs avantages dont une convergence rapide avec une complexité en temps de calcul généralement linéaire avec le nombre de documents. Toutefois, la grande dépendance des performances en fonction du choix des centres initiaux et le manque de méthode pour déterminer la valeur de *K* constituent les inconvénients majeurs de ce type de méthode. Un autre inconvénient est également la limitation des données à cause du calcul des distances entre documents pris deux à deux. Ainsi, notre variante a résolu en partie ces derniers inconvénients :

- En éliminant la notion de distance dans la phase de recentrage. Seule la notion de liens entre documents existe. Ces liens ne sont pas des liens hypertextes.
- En utilisant uniquement les SN des documents. Ceci permet de créer une matrice de similarité suffisamment creuse pour ne pas avoir de problèmes de stockage.

9.2.2 Une méthode de détection de *K*

Les méthodes de partitionnement souffrent de l'absence d'une méthode satisfaisante de détection de *K*. Ainsi, avons-nous élaboré une méthode en adéquation avec notre algorithme avec toutefois les contraintes suivantes :

- Pas d'utilisation de distances, ce qui permet d'économiser leur temps de calcul. Nous utilisons ainsi uniquement la notion de lien citée au paragraphe ci-dessus.
- La notion de seuil, qui généralement est difficile à déterminer, et qui peut varier suivant le corpus de référence, n'est pas utilisée.

Les expérimentations menées avec cette méthode ont donné des résultats très satisfaisants. Toutefois, cette méthode s'applique de préférence sur une matrice de liens creuses.

9.2.3 Mise en œuvre

Notre méthode d'aide à la navigation, SearchXQ, a été mise en œuvre sur notre portail juridique Adminet. Voir l'annexe C pour des vues d'écran de l'outil.

9.3 Perspectives

9.3.1 Vers une meilleure intégration

La première amélioration envisageable consisterait à achever la mise en œuvre de SearchXQ sur l'ensemble des données disponibles sur notre portail comme par exemple les textes européens, le Journal Officiel (JO) ou encore la jurisprudence. Notre méthode n'a été appliquée, pour le moment, qu'aux seuls codes.

Une seconde amélioration possible concerne l'interface : Nous avons vu que le domaine juridique regroupe des termes dont le sens propre et commun diffèrent mais aussi des termes ayant un sens uniquement juridique. L'amélioration serait donc de proposer pour les termes juridiques, la possibilité d'avoir accès à leur définition.

9.3.2 Evaluation de la méthode par des utilisateurs

Un des objectifs a été d'utiliser un corpus nous permettant d'éviter des expérimentations avec des utilisateurs. Toutefois, l'outil final pourrait faire l'objet d'une étude de performances qui puisse faire intervenir à la fois des experts du domaine mais aussi des novices. Cette étude n'a pas été envisagée car il est difficile d'évaluer objectivement ce genre d'outils sans un échantillon important d'utilisateurs (c'est-à-dire quelques centaines d'utilisateurs).

Les fichiers de logs pourraient faire également l'objet d'une étude, permettant de voir si l'outil est utilisé. Si tel est le cas, il serait intéressant de voir dans quelles proportions et avec quelle croissance.

9.3.3 Utilisation d'autres corpus

Nous avons expérimenté notre méthode de classification sur un corpus spécifique au droit. Il serait intéressant d'expérimenter sur d'autres corpus, ou bien dans des corpus juridiques de langues étrangères.

Bibliographie

[Bellot, 1999] P.Bellot. Méthodes de classification et de segmentation pour la recherche documentaire. Workshop "Fouille de textes", GDR I3, Ecole Polytechnique, Paris, Juin 1999.

[Bellot, 2000] P. Bellot. Méthodes de classification et de segmentation locales non supervisées pour la recherche documentaire. Thèse de Doctorat de l'Université d'Avignon, janvier 2000

[Berkin, 2002] P. Berkhin. *Survey Of Clustering Data Mining Techniques*. Technical Report, Accrue Software, 2002.

[Bezdek, 1981] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.

[Botafago, 1993] A.R. Botafogo. *Cluster Analysis for Hypertext Systems*. In Proceedings. of the 16-th Annual ACM SIGIR Conference of Res. and Dev. in Info. Retrieval, pages 116-125, 1993.

[Botafogo et Schneiderman, 1991] A.R. Botafogo, B. Shneiderman. *Identifying Aggregates in Hypertext Structure*. In Proceedings of the Third ACM Conference on Hypertext, pages 63-74, 1991.

[Bourdoncle, 1997] F. Bourdoncle. LiveTopics: Recherche Visuelle d'Information sur l'Internet, Dossiers de l'Audiovisuel, La Documentation Française, numéro 74 (juillet-aout) 36-38, 1997.

[Bourdoncle, 1999] F. Bourdoncle. Panorama et perspectives des outils de recherche d'information textuelle sur internet. In actes du colloque IDT'99, 1999.

[Bourigault, 1993] D. Bourigault. *An Endogenous Corpus-based Method for Structural Nous Phrase Disambiguation*. In Proceedings of the 6th Conference of the European Chapter of the Association of Computational Linguistics (EACL '93), Utrecht, pp. 81-86, 1993.

[Bradley et al., 1998] P. S. Bradley, U. Fayyad, and C. Reina. *Scaling Clustering Algorithms to Large Databases*. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pages 9--15, New York, August 1998.

[Celeux, 1992] G. Celeux and G. Govaert. *A Classification EM Algorithm for Clustering and Two Stochastic Versions*. Computational Statistics and Data Analysis, 14, 315-332. (1992).

[Constant, 1991] P. Constant. Analyse syntaxique par couche. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Paris, 1991.

[Constant, 1995] P. Constant. L'analyseur linguistique Sylex. Ecole d'été du CENT, 1995.

[Cornu, 2000] G. Cornu. Linguistique juridique. Dunod, Paris, 2000.

[Croft, 1977] W. B. Croft. *Clustering Large Files of Documents using the Single-link Method*. Journal of the American Society for Information Science, 28(6):341-- 344, November 1977.

[Cutting et al., 1992] D.R. Cutting, J.O. Pedersen, D. Karger, and J.W. Tukey. Scatter/Gather: A cluster-based Approach to Browsing Large Document Collections. In *Proc. of the 15th Annual International ACM/SIGIR Conference*, pages 318-329, Copenhagen, Denmark, 1992.

[Cutting et al., 1993] D.R. Cutting, D. Karger, and J. Pedersen. *Constant Interaction-time Scatter/Gather Browsing of Very Large Document Collections*. In *Proc. of the 16th Annual International ACM/SIGIR Conference*, pages 126-135, Pittsburgh, PA, 1993.

[Daille, 1994] B. Daille. Approche mixte pour l'extraction de terminologie : statistique lexicale et filtres linguistiques. PhD thesis, Université Paris 7, Paris, France, 1994.

[Daille, 1995] B. Daille. Repérage et extraction de terminologique par une approche mixte statistique et linguistique. Traitement automatique des langues (TAL) 36(1-2), pp. 101-118, 1995.

[Dempster et al., 1977] A. P. Dempster, N. M. Laird, and D. B. Rubin. *Maximum-likelihood from Incomplete Data via the EM Algorithm*. Journal of Royal Statistical Society B, 39:1–38, 1977.

[Diday et al., 1982] E. Diday, J. Lemaire, J. Pouget et F. Testu. Eléments d'analyse des données, Dunod Informatique, 1982.

[Duda et Hart, 1973] R. Duda and P. Hart, "*Pattern Classification and Scene Analysis*", Wiley, New York, 1973.

[El-Hamdouchi et Willet, 1986] A. El-Hamdouchi A. and P. Willet. *Hierarchic Document Clustering using Ward's Method*. In proceedings of the Ninth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1986.

[Estivill-Castro, 2002] V. Estivill-Castro. *Why so many Clustering Algorithms – A Position Paper*. In ACM SIGKDD Explorations, 4 (1):65-75, 2002.

[Evans et al., 1999] D.A. Evans, A. Huettner, X. Tong, P. Jansen and J. Bennett. *Effectiveness of Clustering in Adhoc Retrieval*. In E.M. Voorhees and D.K. Harman (Eds.), Information Technology: The Seventh Text REtrieval Conference (TREC-7), NIST Special Publication 500-242, pp. 143-148, 1999.

[Farnstrom et al., 2000] F. Farnstrom, J. Lewis and C. Elkan. *Scalability for Clustering Algorithms Revisited*. SIGKDD Explorations, 2(1):51-57, 2000.

[Fisher, 1996] D. Fisher. *Iterative Optimization and Simplification of Hierarchical Clusterings*. Journal of Artificial Intelligence Research, 4:147-180, 1996.

[Forgy, 1965] E.W. Forgy. *Cluster Analysis of Multivariate Data:Efficiency Versus Interpretability of Classifications*. Biometric Soc. Meetings, Riverside, California (Abstract in Biometrics 21, No. 3, 768), 1965.

[Frakes and Baeza-Yates, 1992] W.B. Frakes and R. Baeza-Yates eds. *Information Retrieval: Data Structures & Algorithms*. Englewood Cliffs, NJ: Prentice Hall, 1992.

[Gibson et al., 1998] D. Gibson, J. Kleinberg, and P. Raghavan. *Inferring Web Communities from Link Topology*. In Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia, Pittsburg, PA, pp. 225-234, June 20-24, 1998.

[Govaert, 2003] G. Govaert (éditeur). Analyse des données. Hermès, 2003.

[Grabar, 2001] N. Grabar and S. Berland. Construire un corpus Web pour l'acquisition terminologique. Unité de recherche et innovation INIST-CNRS, editor, Actes de conférence TIA'2001, Terminologie et intelligence artificielle, pages 44-54, Nancy, France, mai 2001.

[Griffiths et al., 1984] A. Griffiths, L. A. Robinson, and P. Willett. *Hierarchic Agglomerative Clustering Methods for Automatic Document Classification*. Journal of Documentation, 40(3): 175- -205, 1984.

[Habert et al. 1997] B. Habert, S. Bertrand-Gastaldy, A. Nazarenko, F. Dupuis, E. Naulleau, M. Lemieux & C. Delisle. Recyclage d'analyses syntaxiques automatiques pour le repérage de variantes de termes. In Atelier des projets franco-canadiens, RIAO'97, volume 2, pp. 751-760, Montréal, 1997.

[Hamon et Nazarenko, 2001] T. Hamon et A. Nazarenko. *Detection of Synonymy Links between Terms: Experiment and Results*. Recent Advances in Computational Terminology. Pages 185-208. John Benjamins, 2001.

[Harman, 1992] D. Harman. *Ranking Algorithms*. In [Frakes and Baeza-Yates, 1992], chapter 14, 1992.

[Harman et Smeaton, 1997] D. Harman and A. Smeaton. *The TREC Experiments and their Impact on Europe*. Journal of Information Science. 1997.

[Hearst, 1995] M. Hearst. *TileBars: Visualization of Term Distribution Information in Full Text Information Access*. In Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'95), 59-66, 1995.

[Hearst, 1997] M.A. Hearst. *TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages*. Computational Linguistics, 23 (1), pp. 33-64, March 1997.

[Hearst, 1999] M.A. Hearst. *User Interfaces and Visualization*. In Ricardo Baeza-Yates and Berthier Ribeiro-Neto, editors, Modern Information Retrieval, chapter 10, pages 257-323. ACM Press, 1999.

[Hearst & Pederson, 1996] M.A. Hearst and J.O. Pedersen. *Reexamining the Cluster Hypothesis: Scatter/gather on Retrieval Results*. In Proceedings of the 19th Annual International SIGIR Conference, Zurich, Switzerland, 1996.

[Hindle, 1990]D. Hindle. *Noun Classification from Predicate-argument Structures*. Meeting of the Association for Computational Linguistics, pages 268-275, 1990.

[Jacquemin, 2001] C. Jacquemin. *Spotting and Discovering Terms through Natural Language Processing*. Cambridge, MIT Press, 2001.

[Jain et al., 1999] A. K. Jain, M. N. Murty, and P. J. Flynn. *Data clustering: A review*. In ACM Computing Surveys, vol. 31, no. 3, pages 264-323, 1999.

[Karypis et Kumar, 1998] G. Karypis and V. Kumar. hMETIS*: A Hypergraph Partitioning Package*. Available at http://www.cs.umn.edu/ karypis, 1998.

[Karypis et al, 1999a] Karypis, G.; Han, E.-H.; and Kumar, V. *Chameleon: A Hierarchical Clustering Algorithm using Dynamic Modeling*. IEEE Computer: Special Issue on Data Analysis and Mining 32(8), pages 68-75, 1999.

[Kleinberg, 1999] J.M. Kleinberg. *Authoritative Sources in a Hyperlinked Environment*. Journal of the ACM, 46(5) : 604-532, 1999.

[Kruskal, 1956] J. Kruskal. *On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem*. Proceedings of the American Mathematical Society, 7:48-50, 1956.

[Kumar et al. 1999] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tompkins, "*Trawling the web for emerging cyber-communities*", Proc. 8th International World Wide Web Conference, 1999.

[Lame, 2000] G. Lame. Acquisition de connaissances à partir de textes, vers l'élaboration d'une ontologie du droit. In M. Ayel and J-M Fouet, editors, Actes des RJCIA'2000. cinquièmes rencontres nationales des jeunes chercheurs en intelligence artificielle, pages 211-221, Lyon, 2000.

[Lame, 2001a] G. Lame. *A Categorization Method for French Legal Documents on the Web*. In H. Prakken, editor, Proceedings of the 8th International conference on artificial intelligence and law, pages 219-220, Saint-Louis MO USA, 2001.

[Lame, 2001b] G. Lame. Classement automatique de documents et analyse terminologique de corpus. In unité de recherche et innovation INIST CNRS, editor, Actes de la conférence TIA'2001, quatrièmes rencontres Terminologie et Intelligence Artificielle, pages 149-158, Nancy, France, 2001.

[Lame, 2002] G. Lame. Construction d'ontologies à partir de textes : une ontologie du droit dédiée à la recherche d'informations sur le Web. PhD thesis, Ecole des Mines de Paris, 2002.

[Lance et Williams, 1967] G. N. Lance and W. T. Williams. *A General Theory of Classificatory Sorting Strategies. HierarchicaSystems*. Computer Journal 9, pages 373-380, 1967.

[Larson 1996] R. R. Larson. *Bibliometrics of the World Wide Web: an Exploratory Analysis of the Intellectual Structures of Cyberspace*. Proc. SIGIR'96, 1996.

[Lefèvre, 2000] P. Lefèvre. La recherche d'information, du texte intégral au thésaurus. Hermes Science, Paris, 2000.

[Lelu et Hallab, 2000] A. Lelu et M. Hallab. Consultation floue de grandes listes de formes lexicales simples et composes : un outil préparatoire pour l'analyse de grands corpus textuels. Actes des JADT'2000, Lausanne, Mars 2000.

[L'Homme, 2001] M.C. L'Homme. Nouvelles technologies et recherche terminologique. Techniques d'extraction des données terminologiques et leur impact sur le travail du terminographe. *L'impact des nouvelles technologies sur la gestion terminologique*, Université York, Toronto, août 2001.

[L'Homme, 2002] M.C. L'Homme. Fonctions lexicales pour représenter les relations sémantiques entre termes. *Traitement automatique de la langue (TAL)* 43(1), pp. 19-41, 2002.

[Luhn, 1957] H.P Luhn. *A statistical approach to mechanized encoding and searching of literary information*. IBM Journal of Research and Development, 4(4), 600-605, 1957.

[MacQueen, 1967] J. B . MacQueen. *Some Methods for Classication Analysis of Multivariate Observations*. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability I, pages 281-297, CA: University of California Press, 1967.

[Mercier, 1997] Mercier. Analyse, Indexation documentaire dans un centre de documentation.1997.

[Michelet, 1988] B. Michelet. L'Analyse des Associations. Unpublished Ph.D. Thesis, Université Paris VII, Paris, 1988.

[Modha et Sangler, 2000] D. Modha and W. S. Spangler. *Clustering Hypertext with Applications to Web Searching*. In Proceedings of ACM Conference on Hypertext and Hypermedia, 2000.

[Moens, 2000] M.-F. Moens. *Automatic Indexing and Abstracting of Document Texts* (The Kluwer International Series on Information Retrieval 6). Kluwer Academic Publishers: Boston, 2000.

[Moreno, 2001] A. Moreno et C. Perez. *From Text to Ontology : Extraction and Representation of Conceptual Information*. In CNRS Unité de Recherche et Innovation, INIST, editor, Actes de la conférence TIA'2001, Terminologie et Intelligence Artificielle, pages 233-242, Nancy, France, mai 2001.

[Morin, 1999] E. Morin. Des patrons lexico-syntaxiques pour aider au dépouillement terminologique. Traitement Automatique des Langues, 40(1): 143-166, 1999.

[Nédellec et al., 2001] C. Nédellec, M. Ould Abdel Vetah et P. Bessières. *Sentence Filtering for Information Extraction in Genomics, a Classification Problem*. In Proceedings of the Conference on Practical Knowledge Discovery in Databases, PKDD'2001, p. 326-338, Freiburg, septembre 2001.

[Oueslati, 1999] R. Oueslati. Aide à l'acquisition de connaissances à partir de corpus. Thèse de doctorat, Université Louis Pasteur Strasbourg, 1999.

[Pearson, 1998] J. Pearson. *Terms in Context*. Amsterdam – Philadelphia, John Benjamins, 1998.

[Pereira et al., 1993] F. Pereira, N. Z. Tishby, and L. Lee. *Distributional Clustering of English Words*. In 30th Annual Meeting of the Association for Computational Linguistics, pages 183-190, Columbus, Ohio, 1993.

[Pelleg et Moore, 2000] D. Pelleg and A. Moore. *X-Means: Extending k-means with Efficient Estimation of the Number of Clusters*. In Proceedings of ICML-2000, pages 727-734, 2000.

[Pirolli et al., 1996] P. Pirolli, J.E. Pitkow, and R. Rao. *Silk from a Sow's ear: Extracting Usable Structures from the Web*. In Proceedings of CHI '96: Human factors in computing systems, Vancouver, B.C., Canada, April 1996.

[Pirolli et al., 1996a] P. Pirolli, P. Schank, M.A. Hearst, and C. Diehl. *Scatter/gather Browsing Communicates the Topic Structure of a Very Large Text Collection*. In Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages* 213-220, Zurich, Switzerland, May 1996.

[Prim, 1957] R. C. Prim, *Shortest Connection Networks and some Generalizations*, Bell Systems Technology Journal 36, pages 1389-1401, 1957.

[Rousselot et al., 1996] F. Rousselot, P. Frath P and R. Oueslati. *Extracting Concepts and Relations from Corpora*. In Proceedings of the Corpus-Oriented Semantic Analysis Workshop of ECAI'96 Budapest p.74-78, 1996.

[Saint-Jean, 2001] Ch. Saint-Jean. Classification paramétrique robuste partiellement supervisée en reconnaissance des formes. PhD thesis, Université de La Rochelle, L3I, Décembre 17, 2001.

[Salton, 1983] G. Salton and M. J. McGill, *Introduction to modern information retrieval*, McGraw-Hill, New York, 1983.

[Salton, 1989] G. Salton. *Automatic Text Processing: the Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, MA, 1989.

[Salton, 1994] G. Salton and J. Allan. *Text Retrieval Using the Vector Processing Model*. In Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval (SDAIR), 1994.

[Salton et Buckley, 1988] G. Salton and C. Buckley. *Term weighting approaches in automatic text retrieval*. Information Processing and Management, vol. 24, no. 5, pages 513-523, 1988.

[Saporta, 1990] G. Saporta. Probabilités, analyse des données et statistique. Editions Technip, Paris, 1990.

[Schmidt, 1997] C. Schmidt. La langue juridique: maux et remèdes. http://juripole.u-nancy.fr/tradjur.html.

[Schneiderman, 1997] B. Shneiderman. *Designing the User Interface: Strategies for Effective Human-computer Interaction*. Addison-Wesley, Reading, MA, 1997.

[Schütze et Siverstein, 1997] H. Schütze and C. Silverstein. *Projections for Efficient Document Clustering*. In Proceedings of the 20th International ACM SIGIR Conference, 1997.

[Silverstein & Pedersen, 1997] C. Silverstein and J.O. Pedersen. *Almost-constant Time Clustering of Arbitrary Corpus Subsets*. In Proceedings of the 20th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97), 60-66, 1997.

[Silverstein et al., 1998] C. Silverstein, M. Henzinger, and H. Marais. *Analysis of a Very Large Altavista Query Log*. Technical note #1998-014, Digital SRC, Oct. 1998.

[Sinclair, 1996] J. Sinclair. *Preliminary Recommendations on Corpus Typology*. Technical report, EAGLES (Expert Advisory Group on Language Engineering Standards), 1996.

[Smadja et McKeown, 1990] F.A. Smadja, K. McKeown. *Automatically Extracting and Representing Collocations for Language Generation*. In Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics, pages 252-259, 1990.

[Tesniere, 1959] L. Tesniere. Elément de syntaxe structurale, Klincksieck, Paris, 1959.

[Van Rijsbergen, 1979] C. J. van Rijsbergen. *Information Retrieval*. Butter- worths, London, 2nd edition, 1979.

[Voorhees, 1986a] E. Voorhees, *The Effectiveness and Efficiency of Agglomerative Hierarchic Clustering in Document Retrieval*, Ph.D. Thesis, 1986.

[Voorhees & Harman, 1999] E.M. Voorhees and D. Harman. *Overview of the Seventh Texte Retrieval Conference (TREC 7)*. Actes de Text REtrieval Conference TREC-7, Gaithersburg, Maryland, NIST special publication 500-242, pages 1-24, 1999.

[Ward, 1963] J. H. Ward. *Hierarchical Grouping to Optimize an Objective Function*. Journal of the American Statistical Association, 58(301), pages 236-244, 1963.

[Weiss et al., 1996] R. Weiss, B. Velez, M. Sheldon, C. Namprempre, P. Szilagyi, A. Duda, and D. Gifford. *Hypursuit: A Hierarchical Network Search Engine that exploits Content-link Hypertext Clustering*. In Proceedings of the Seventh ACM Conference on Hypertext, pages 180-193, 1996.

[White et McCain, 1998] H. D. White and K. W. McCain. *Visualizing a Discipline: An Author Co-citation Analysis of Information Science*, 1972---1995. Journal of the American Society for Information Science, 49, 4, pages 327-356, 1998.

[Xu et al., 2002] X. Liu, Y. Gong, W. Xu and S. Zhu. *Document Clustering with Cluster Refinement and Model Selection Capabilities*. In Proceedings of SIGIR'02: 191-198, 2002.

[Zhang, 2000] B. Zhang. *Generalized k-harmonic Means – Boosting in Unsupervised Learning.* Technical Report HPL-2000-137, Hewlett-Packard Labs, 2000.

Table des figures

TABLE DES FIGURES

Liste des tableaux

Liste des algorithmes

Annexe A

Corpus de référence : acronymes et intitulés des codes

Les acronymes sont empruntés du site http://legifrance.gouv.fr

Acronyme	**Intitulé**
CACTSOC	code de l'action sociale et des familles
CARTISA	code de l'artisanat
CASSURA	code des assurances
CAVIACI	code de l'aviation civile
CCIVILL	code civil
CCOMMER	code de commerce
CCOMMUN	code des communes
CCONSOM	code de la consommation
CCONSTR	code de la construction et de l'habitation
CDABBOI	code des débits de boissons et des mesures contre l'alcoolisme
CDARCHI	code de déontologie des architectes
CDCHIRD	code de déontologie des chirurgiens-dentistes
CDMEDIC	code de déontologie médicale
CDPOLIC	code de déontologie de la police nationale
CDSAGES	code de déontologie des sages-femmes
CDVETER	code de déontologie vétérinaire
CDVIMAR	code disciplinaire et pénal de la marine marchande
CDWOMET	code du domaine de l'état
CDXFLUV	code du domaine public fluvial et de la navigation intérieure
CDYANES	code des douanes
CEDUCAT	code de l'éducation
CELECTO	code électoral
CENVIRO	code de l'environnement
CEXPROP	code de l'expropriation pour cause d'utilité publique
CFAMILL	code de la famille et de l'aide sociale

CFOREST	code forestier
CGCTERR	code général des collectivités territoriales
CGIMP	code général des impôts, cgi
CGLIVPF	livre des procédures fiscales
CINDCIN	code de l'industrie cinématographique
CJURFIN	code des juridictions financières
CJUSADM	code de justice administrative
CJUSMIL	code de justice militaire
CLEGHON	code de la légion d'honneur et de la médaille militaire
CMARPUB	code des marches publics
CMINIER	code minier
CMONFIN	code monétaire et financier
CMUTUAL	code de la mutualité
CORGJUD	code de l'organisation judiciaire
CPENALL	code pénal
CPENSIC	code des pensions civiles et militaires de retraite
CPENSIM	code des pensions militaires d'invalidité et des victimes de la guerre
CPENSIR	code des pensions de retraite des marins français du commerce, de pêche ou de plaisance
CPORMAR	code des ports maritimes
CPOSTES	code des postes et télécommunications
CPROCIV	nouveau code de procédure civile
CPROCPE	code de procédure pénale
CPROINT	code de la propriété intellectuelle
CROUTE	code de la route
CRURAL	code rural
CSANPU	code de la santé publique
CSECSOC	code de la sécurité sociale
CSERVNA	code du service national
CTRAVAI	code du travail
CTRAVMA	code du travail maritime
CURBANI	code de l'urbanisme
CVOIRIE	code de la voirie routière

Annexe B

Liste de termes juridiques

Abandon, abandon de famille, abattement supplémentaire, abrogation, abroger, absence, abstention, abstention constructive, abstention positive, abus, acceptation, accession, accessoire, accident du travail, accord amiable, accord de Schengen, accord européen, accord international bilatéral, accord international multilatéral, accord social, accroissement, accusatoire, accusé, acompte, acquiescement, acquis communautaire, acquit, acquittement, acquitter, acquéreur, acquêt, acte, acte administratif, acte authentique, acte conservatoire, acte d'administration, acte d'état civil, acte de commerce, acte de disposition, acte de notoriété, acte de procédure, acte juridictionnel, acte notarié, acte sous seing privé, actif, action, action civile, action directe, action en justice, action estimatoire, action oblique, action paulienne, action personnelle, action publique, action réelle, actionnariat des salariés, activité agricole, ad hoc, ad litem, additionnelle, adhésion, adjudicataire, adjudication, administrateur, administrateur ad hoc, administrateur de biens, administrateur judiciaire, administration, administration légale, administration pénitentiaire, admission des créances, admonestation, adoptabilité de l'enfant étranger, adoption, adoption internationale, adoption plénière, adoption simple, affacturage, affaire pendante, affiliation, affrètement, agence immobilière, agents de justice, agriculture raisonnée, agrégation, agrément, ags, aide de préadhésion, aide juridictionnelle, aide juridique, aide sociale à l'enfance, aide à l'accès au droit, ajournement, alignement, aliments, aliénation, allocataire, allocation, allégation, alternative aux poursuites pénales, aléatoire, amende, amende civile, amiable, amiable compositeur, amicus curiae, amnistie, aménagement, anatocisme, annulation, antenne de justice, antichrèse, antériorité, apatride, apostille, apparence, apparentement, appel, appel en garantie, appel nullité, appel-nullité, appellation d'origine, appellation d'origine contrôlée, appellation d'origine protégée, apports en société, approfondissement, arbitrabilité, arbitrage, arbitrage international, arbitrage multipartite, arbitre, architecture européenne, argument, arrhes, arrérage, arrêt, arrêt de mise en accusation, arrêté, arrêté de cessibilité, ascendant, asile, assemblée, assemblée générale de copropriété, assesseur, assiette, assignation, assises, assistance éducative, assistant de justice, assistante maternelle, association, association intermédiaire, associations de parents adoptifs, associé, associés d'exploitation, assujettissement, assurance, assurance construction, assurance de protection juridique, assurance personnelle, assurance professionnelle, assurance rcp, assurance volontaire, assurance trc - tous risques chantiers -, assuré social, astreinte, attendu, attestation de spécificité, audience, audience de départage, audience foraine, audience solennelle, audiovisuel, auditeur de justice, audition, auteur, authentique, autorisation, autorité centrale, autorité de la chose jugée, autorité parentale, auxiliaire de justice, auxiliaires de

justice, aval, avance, avancement d'hoirie, avant-dire droit, avantage acquis, avantage en nature, avantages contributifs, avantages non contributifs, avenant, aveu, avocat, avocat au conseil d'état et à la cour de cassation, avocat commis d'office, avocat général, avoué, ayant cause, ayant droit, bail, bail commercial, bail d'habitation, bail emphytéotique, bail professionnel, bail rural, bail à cheptel, bail à colonat partiaire, bail à complant, bail à ferme, bailleur, banque centrale européenne, banqueroute, bans, barreau, baux, bce, bien, bien commun réservé, bien immobilier, bien propre, biens communs, biens corporels, biens immobiliers, biens incorporels, biens indivis, biens propres, bigamie, bilatéral, billet à ordre, bonne foi, bornage, brevet, budget, bâtonnier, bénéfice agricole, bénéfices industriels et commerciaux, caducité, caisse de congés payés, calamités agricoles, canon emphytéotique, cantonnement, capable, capacité, capacité adoptive, capacité juridique, capital social, carence, carte professionnelle, cas de force majeure, casier judiciaire, cassation, cause, caution, cautionnement, cecos, cedh, centre d'arbitrage, centre d'étude et de conservation des oeufs et du sperme humains, centre de détention, centre de placement immédiat, centre de semi-liberté, centre pénitentiaire, centre éducatif renforcé, certificat de coutume, certificat de nationalité française, certification, cessation des paiements, cession, cession de créance, cessionnaire, chambre, chambre d'accusation, chambre de l'instruction, chambre du conseil, chancelier, chancellerie, charge, charge de la preuve, charges récupérables, charte des droits fondamentaux, charte des services publics, charte sociale, chemin d'exploitation, chemin rural, chirographaire, chose, circulation des mineurs étrangers, circulation des personnes, citation, citation directe, citation en justice, citoyen, citoyenne, citoyenneté, citoyenneté de l'union, civil, classement sans suite, classement sous condition, classification des dépenses, clause, clause compromissoire, clause d'exemption, clause de suspension, clause pénale, clause résolutoire, clerc, cmu, code, code civil, code de commerce, code de procédure civile, code de procédure pénale, code du travail, code pénal, codicille, codification des textes législatifs, codébiteur, coefficient d'occupation des sols, cohérie, cohéritier, cohésion économique et sociale, coindivisaire, collatéral, collectivité, collectivité territoriale, collectivités territoriales, collocation, collégialité, colégataire, comitologie, comité de conciliation, comité de probation, comité des régions, comité et groupes de travail, comité politique, comité économique et social, commandement, commanditaire, commandite, commandité, commerce d'enfants, commerçant, comminatoire, commis, commis d'office, commissaire, commissaire-priseur, commission, commission d'indemnisation des victimes d'infraction pénale, commission de recours amiable, commission de surendettement des particuliers, commission européenne, commission rogatoire, commission rogatoire internationale, commissions parlementaires, commissoire, commodat, communal, communale, communautarisation, communauté conjugale, commune, communicable, communication des causes, comourants, comparution, comparution immédiate, comparution personnelle, compensation, complainte, complice, complicité, composition de la commission, composition du parlement européen, composition pénale, compromis, compromis de Luxembourg, compromis de vente, compte-courant, compulsoire, compétence, compétence communautaire, compétence des arbitres, compétence externe de la communauté européenne, compétence subsidiaire, compétitivité, concentration, conception-réalisation, conciliateur, conciliateur de justice, conciliation, conclusions, concordat, concubinage, concurrence, condamnation, condamnation avec sursis, condamnation définitive, condamnation par défaut, condamné, conditions suspensives, confirmation, conflit d'intérêts, confrontation, confusion, confusion des peines, conférence, conférence européenne, conférence intergouvernementale, conférence intergouvernementale bilatérale, congrès, congé, conjoint, connaissement, connexité, conseil constitutionnel, conseil d'état, conseil de famille, conseil de famille des pupilles de l'état, conseil de l'union, conseil de prud'hommes, conseil des prud'hommes, conseil départemental de l'accès au droit, conseil européen, conseil général, conseil supérieur de l'adoption, conseil supérieur de la magistrature,

conseil syndical, conseiller, consentement à l'adoption, conservatoire, consignation, consolidation, consolidation des textes législatifs, consort, consorts, constat, constitution, constitutionnel, constitutionnelle, constructeur, constructibilité limitée, consultation, consultation juridique, contentieux, contestation de paternité légitime, contractant, contractante, contradiction, contradictoire, contrainte, contrainte par corps, contrat, contrat d'arbitrage, contrat d'architecte, contrat d'entreprise, contrat d'intégration, contrat de location, contrat de mariage, contrat de maîtrise d'œuvre, contrat territorial d'exploitation, contravention, contre-lettre, contredit, contribution aux charges du mariage, contrôle de l'application du droit communautaire, contrôle judiciaire, contrôle sanitaire des enfants étrangers, contrôle technique, convention, convention de conversion, convention de croupier, convention de mise à disposition, convention de Schengen, convention européenne des droits de l'homme, coobligé, coopération policière et judiciaire en matière pénale, coopération politique européenne, coopération renforcée, coordination, copie exécutoire, copropriétaire, copropriété, coreper, coreu, cos, cotisations salariales, cotraitant, coupable, cour, cour d'appel, cour d'assises, cour de cassation, cour de justice, cour des comptes, courtier, couverture maladie universelle, covendeur, crime, critère d'adhésion, critère de convergence, critère de Copenhague, croupier, créance, créancier, créancier chirographaire, créancier hypothécaire, créancier privilégié, création d'un lien de filiation entre l'enfant et sa famille adoptive, crédit bail, crédit documentaire, crédit-bail, crédit-bailleur, culpabilité, culture, curatelle, curateur, dation en paiement, degré de juridiction, demande additionnelle, demande en intervention, demande reconventionnelle, demandeur, denier, descendant, descente sur les lieux, destination, destination du père de famille, dette, devis, dialogue social, dilatoire, diligence, diligenter, discrétionnaire, discuter, disjonction, dispense de peine, dispositif, dispositif du jugement, distances à respecter pour les plantations, divorce, doctrine, document d'urbanisme, dol, domicile, domicile élu, domiciliation, dommage, dommage-intérêt, dommage-ouvrage, dommages aux existants, dommages de travaux publics, dommages et intérêts, dommages-intérêts, don, donataire, donateur, donation, donner acte, dotal, dotaux, droit, droit commun, droit communautaire, droit d'alerte, droit d'initiative, droit d'usage et d'habitation, droit de délaissement, droit de plaidoirie, droit de propriété, droit de préemption urbain, droit de préférence, droit de pétition, droit de repentir, droit de reprise, droit de rétention, droit de rétrocession, droit de se clore, droit de suite, droit international privé, droit privé, droit public, droit rural, droits civils, droits de l'homme, droits dérivés, ducroire, dup, débats, débauchage, débiteur, débitrice, débours, débouter, débouté, débrayage, déchéance, décision administrative, décision de justice, décision-cadre, décisoire, déclaration, déclaration au greffe, déclaration d'utilité publique, déclaration de créances, déclaration de petersberg, déclaration préalable, déclinatoire, décret, décès, défaut, défendeur, défense, défense collective, défenseur, déficit démocratique, défrichement, déférer, dégâts de gibier, délai congé, délai de carence, délai de grâce, délai de procédure, délai de viduité, délai franc, délai préfix, délai-congé, délais de procédure, délibéré, délinquant, délit, délégation, délégation d'autorité parentale, délégué du procureur, démembrement du droit de propriété, déni de justice, dénomination sociale, dénonciation de nouvel oeuvre, départage, département, départemental, départementale, dépens, dépositaire, déposition, dépôt de garantie, désaveu de paternité, déshérence, désintéressement, désistement, dessaisissement, détachement, détention, détention provisoire, détenu, développement durable, développement rural, dévolution, effet de commerce, emphytéose, empiétement, emplacements réservés, employeur, empreintes génétiques, empêchement, enchère, endossement, enlèvements d'enfants, enquête, enquête judiciaire, enquête parcellaire, enquête préalable sur l'utilité publique, enquête préliminaire, enquête publique, enquête sociale, enregistrement, enrichissement sans cause, enrôler, ententes, entraide agricole, entrepreneur, entreprise, environnement, envoi en possession, espace de liberté de sécurité et de justice, espaces boisés, ester en justice, estimatoire, eurocorps,

eurofor, euromarfor, europe, europol, exception, exception d'incompétence, exception de connexité, exciper, exclusion, exécution provisoire, exequatur, exigibilité, exonération, exorbitant, expert immobilier, expert judiciaire, expertise, expertise judiciaire, expertise sanguine, exploit, exploit d'huissier de justice, exposé des motifs, expropriation, expulsion, expédient, expédition, extra petita, extradition, extrajudiciaire, exécuteur, exécution, exécution provisoire, faculté, faillite personnelle, faire droit, fausses déclarations de naissance, faute, faute inexcusable, faux, fermage, feuille d'audience, filiale, filiation, fin de non recevoir, fisc, fiscal, fiscalité, fiscalité et adoption, flagrant délit, foncier, foncière, fond, fonds de commerce, fonds de garantie, fonds européen d'orientation et de garantie agricole, fonds structurels et fonds de cohésion, fongible, force de chose jugée, force exécutoire, force majeure, force publique, forclusion, forfait, formation professionnelle, former un pourvoi, formule exécutoire, formule ou force exécutoire, fortuit, fortune de mer, foyer d'action éducative, frais d'acquisition, frais de justice, frais de notaire, frais et dépens, frais irrépétibles, frais professionnels, franchise, fraude, fruits, frustratoires, fusion, gage, garantie, garantie de parfait achèvement, garantie financière, garanties, garanties biennale et décennale, garanties dûes par les constructeurs, garde, garde des sceaux, garde à vue, gestion d'affaires, gie, globalisation de l'économie, gouvernement, gracieuse, gratification, greffe, greffier, greffier en chef, grosse, groupement agricole d'exploitation en commun, groupement d'employeurs, groupement d'entreprises, groupement d'intérêt économique, grâce, guichet unique de greffe, gérance, gérant de société, habilitation, haut-commissaire, hiérarchie des actes communautaires, hiérarchie des normes, hoir, hoirie, homicide, homicide involontaire, homicide volontaire, homologuer, honoraires, hors de cause, huis clos, huis-clos, huissier, huissier de justice, hypothèque, hypothécaire, héritage, héritier, identité européenne de sécurité et de défense, illicite, illégal, immeuble, immigration, immobilier, immunité, impartir, impenses, implication, imputation, in solidum, inaliénabilité, inaliénable, incapable, incapacité, incarcération, incident, incompétence, incorporels, indemnité, indemnité d'immobilisation, indexation, indication géographique protégée, indignité successorale, indivis, indivisaire, indivisibilité, indivisible, indivision, indu, indû, inexcusable, inexécution, infirmation, infirmer, information judiciaire, infra petita, infraction, injonction, injonction de faire, injonction de payer, injonction de soins, innocence, insaisissabilité, inscription de faux, insolvabilité, insolvable, inspection, installation, instance, instance arbitrale, institution, instruction, instrumentaire, instruments juridiques communautaires, intention, interdiction, interjeter, interlocutoire, intermédiaire, interprétation, intestat, intimé, intuitu personae, intégration de l'accord social, intéressement, inventaire, investiture de la commission, irrecevabilité, irréfragable, irrépétible, irrévocabilité, irrévocabilité de l'adoption plénière, itératif, jeX, jetons de présence, jeunesse, jonction, jouissance légale, jours et vues sur les fonds voisins, juge, juge aux affaires familiales, juge commissaire, juge d'appui, juge d'instruction, juge de l'application des peines, juge de l'expropriation, juge de l'exécution, juge de la mise en état, juge des enfant, juge des enfants, juge des référés, juge des tutelles, juge du siège, juge départiteur, jugement, jugement avant dire droit, jugement contradictoire, jugement d'adoption, jugement d'expédient, jugement par défaut, jugement sur le fond, juges non professionnels, juridiction, juridiction administrative, juridiction civile, juridiction de droit commun, juridiction pénale, juridiction spécialisée, juridictionnel, juridictionnelle, juridictions, jurisprudence, jury, juré, justice, justice et affaires intérieures, label agricole, leasing, leg, lettre, lettre de change, lettre de mission, liberté surveillée, libre circulation des personnes, libéralité, libération conditionnelle, libératoire, licenciement, licitation, lien de filiation, lien de subordination, ligne, ligne de succession, liquidation, liquidation judiciaire, lisibilité des traités, litige, litiges d'ordre civil, litisconsorts, litispendance, livre blanc, livre vert, livres verts, locataire, location, location meublée, location-accession, location-gérance, locations saisonnières, lock-out, loi, lot, lotissements, louage, lutte contre la criminalité internationale organisée et le blanchiment de l'argent, lutte contre la

drogue, lutte contre la fraude, lutte contre le racisme et la xénophobie, lutte contre le terrorisme, légal, légale, légalisation, légalisation des dossiers d'adoption, légataire, légataire universel, légitimation, légitime défense, léonin, lésion, magasins généraux, magistrat, magistrat du siège, magistrats du ministère public, magistrats du parquet, magistrats du siège, mainlevée, maison centrale, maison d'arrêt, maison de justice, maison de justice et du droit, majeur, majeur protégé, majorité, majorité qualifiée, majorité qualifiée renforcée, malfaçon, mandant, mandat, mandat d'amener, mandat d'arrêt, mandat de comparution, mandat de dépôt, mandataire, mandataire liquidateur, manquement, marc le franc, marchand de biens, marchand de listes, marché a forfait, marchés de travaux, mariage, marque de fabrique, maternité de substitution, matrimonial, maîtrise d'ouvrage, membres associés de l'ueo, mercuriale, mesure conservatoire, mesure d'administration, mesure de réparation, mesures conservatoires, mesures provisoires, mettre à néant, meuble, meubles corporels, meubles incorporels, milieu ouvert, mineur, mineurs demandeurs d'asile, ministre, ministère public, minorité, minute, minute de jugement, mise au rôle, mise en accusation, mise en cause, mise en demeure, mise en examen, mise en état, mise à pied, mission de base bâtiment, missions de petersberg, mitoyenneté, mobilier, modulation des aides, mondialisation, monopole, montage de promotion immobilière, moratoire, motif, motif surabondant, moyen, moyens et motifs, multilatéral, multipartite, mutation, mutuelle, médiateur, médiateur de la république, médiateur européen, médiateur judiciaire, médiation, médiation judiciaire, médiation pénale, mémoire, métayage, métayage ou bail à colonat partiaire, méthodes communautaire et intergouvernementale, nantissement, national, nationale, nationalité, naturalisation, naturalisé, naturalisée, nolisement, nom commercial, nomenclature, non avenu, non-lieu, norme, notaire, notification, notification d'une décision, notoire, notoriété, novation, noyau dur, nu-propriétaire, nue-propriété, nullité, nus-propriétaires, négociations d'adhésion, obligation, obligation alimentaire, obligation d'information, obligation de divisibilité, obligation de réserve, obligation de réserve, observateur auprès de l'ueo, oeuvre d'adoption, office des migrations internationales, office européen de lutte antifraude, office européen de police, office public d'aménagement et de construction, officier de l'état civil, officier de police judiciaire, officier ministériel, officier ministériel ou officier public, officier public, officier public ou ministériel, officiers ministériels, offres réelles, olaf, olographe, omission de statuer, onéreux, opac, opc - ordonnancement, opposabilité, opposition, opposition à tiers détenteur, opting out, ordonnance, ordonnance d'expropriation, ordonnance de taxe, ordonnance pénale, ordonnance sur requête, ordre, ordre administratif, ordre judiciaire, ordre public, organisation du traité de l'atlantique nord, organisation judiciaire, organisme, organisme agréé, organisme autorisé, original, otan, outrage, pac, pacs, pacte, pacte civil de solidarité, pacte commissoire, pacte de préadhésion sur la criminalité organisée, pacte de stabilité et de croissance, pae, paiement, paraphe, parlement, parlement européen, parlements nationaux, parquet, parquet général, parrainage, partage, partage d'ascendants, partenaires associés de l'ueo, partenaires sociaux, partenariat pour l'adhésion, partie civile, parties, passerelle communautaire, passif, paternité, patrimoine, patrimonial, paulienne, payeur, paysage, pendante, pension, pension alimentaire, permis de construire, perquisition, personnalité juridique, personnalité juridique de l'union, personne morale, perte d'une chance, pesc, pig, pignoratif, piliers de l'union européenne, pilotage, pièce du dossier, pièce à conviction, placement, placement en vue de l'adoption plénière, placement familial, plafond légal de densité, plainte, plan d'occupation des sols, plan de redressement, pld, plumitif, police de l'audience, police judiciaire, police unique de chantier, politique agricole commune, politique commerciale commune, politique commune des transports, politique de défense commune, politique monétaire, politique sociale, politique économique, politique étrangère et de sécurité commune, pollicitation, pondération des voix au conseil, portable, pos, position commune, position dominante, possession d'état, possessoire, postulation, potestatif, potestative, pourvoi, pourvoi

en cassation, pouvoir discrétionnaire, pouvoir souverain, pouvoirs publics, preneur, prescription, prescription civile, prestation, prestation compensatoire, prestations familiales et adoption, pretium doloris, preuve, preuve par écrit, principe d'immunité, principe de la non-discrimination, principe de précaution, principe de subsidiarité, prise à partie, prison, privilège, privilège de juridiction, prix forfaitaire, prix unitaire, probation, procréation médicalement assistée avec tiers donneur, procuration, procureur, procureur de la république, procureur général, procès, procès verbal, procès-verbal, procès-verbaux, procédure, procédure abusive, procédure civile, procédure d'ordre, procédure de codécision, procédure de coopération, procédure de divorce, procédure de l'avis conforme, procédure de l'avis simple, procédure pénale, procédure électorale uniforme, programme d'aménagement d'ensemble, programme de l'union européenne, projet d'intérêt général, promesse de vente, promoteur, propriété, protection civile, protection des consommateurs, protection judiciaire de la jeunesse, protocole social, protêt, provision, prud'hommes, préavis, précarité, préciput, préciputaire, précompte, préemption, préfix, préférence, préjudice, préjudice corporel, préjudice d'agrément, préjudice matériel, préjudice moral, préjudiciel, présidence de l'union, président, président de la commission européenne, présomption, présomption d'innocence, présomption de paternité, présomptions, prétention, prétentions, prévenu, prêt, publicité, publicité des jugements, publique, puc, pupille, pupille de l'état, purger, putatif, péremption, péremption de l'instance, période suspecte, pétitoire, qualification, qualité pour agir, quasi-contrats, quasi-délit, question préjudicielle, quittance, quote-part, quotes-parts, quotes-parts, quotité, quérable, radiation, rappel à la loi, rapport, rapport de suivi, ratification, rcd, recel, recevabilité, recherche de paternité naturelle, recherche des origines, recherche et développement, reconduction, reconnaissance, reconnaissance d'enfant naturel, reconnaissance de plein droit des décisions étrangères, recours, recours amiable, recours en cassation, recours en révision, recouvrement, rectification, rectification judiciaire d'un acte d'état civil, redressement, redressement judiciaire, refonte des textes législatifs, registre du commerce, registre spécial des agents commerciaux, regroupement familial, relaxe, relevé de forclusion, reliquat, reliquataire, relèvement, renonciation, rente, renvoi, repentir, reprise de l'acquis communautaire, représentant des créanciers, représentation, requérant, requérante, requête, rescision, responsabilité civile, responsabilité civile du fait des travaux, responsabilité civile décennale, responsabilité civile professionnelle, responsabilité contractuelle, responsive, ressort, ressources propres, retenue de garantie, retrait, retranchement, revendication, revenu minimum d'insertion, risque, ristourne, rnu, rogatoire, rotation de la présidence, rupture du lien de filiation préexistant, règlement de copropriété, règlement national d'urbanisme, récidive, récognitif, récolement, récompense, récursoire, récusation, rédhibitoire, référendaire, référé, régimes matrimoniaux, région, régional, régionale, régions ultrapériphériques, réglement, réhabilitation, réintégrande, réintégration, réméré, répertoire des métiers, réputé contradictoire, répétition de l'indu, réquisition d'emprise totale, réquisitions, réquisitoire, réserve héréditaire, résiliation, résolution, résolutoire, rétention, rétracter, rétroactif, rétrocession, réversion, révision, révision des traités, révocation, révocation de l'adoption simple, rôle, saisie, saisie conservatoire, saisie des rémunérations, saisie-attribution, saisie-vente, saisine, salaire, salaire différé, salarié, santé publique, satisfactoire, sauvegarde de justice, scec, scellés, Schengen, schémas directeurs, sci, scpi, screening, seing privé, sem, semi-liberté, sentence, serment, serment décisoire, serment supplétoire, service central d'état civil, service public, service pénitentiaire d'insertion et de probation, service universel, services d'intérêt général, services d'intérêt économique général, servitude, servitude de marchepied, servitude de passage en cas d'enclave, servitude de passage le long du littoral, servitude de visibilité, servitude de vue, servitudes, servitudes d'utilité publique, servitudes d'écoulement des eaux, shob, shon, sieur, signature, signification, simplification des traités, simplification législative, simulation, siège, siège de l'arbitrage, siège social, smic, société, société civile d'attribution, société

civile de construction-vente, société civile de location ou de gestion, société civile immobilière, société d'économie mixte, solidarité, solvabilité, solvable, sommation, sommation de payer, soulte, sous-acquéreur, sous-traitance, sous-traitants, soutenir, souverain, staries, statuer, statut, statuts, stellionat, stipuler, stratégie commune, stratégie coordonnée pour l'emploi, stratégie de préadhésion, subrogation, subrogé-tuteur, subsides, subsidiaire, subsidiarité, substitut, substitut du procureur, substitut général, substitution, succession, succession et adoption, suivi socio-judiciaire, superprivilège, superstaries, supplétif, supplétoire, suppression du protocole social, surabondant, surenchère, surendettement, surseoir, sursis, sursis avec mise à l'épreuve, sursis simple, suspensif, suspicion légitime, synallagmatique, syndic, syndic de copropriété, syndicat, syndicat des copropriétaires, sécurité sociale, sénat, séparation de biens, séparation de corps, séquestre, sûreté, tacite, tarif de responsabilité, tarifs de chancellerie, taux du ressort, taxe, taxe départementale des espaces naturels sensibles, tentative, territorialité, testament, testament authentique, testament olographe, testament-partage, testamentaire, testateur, testimonial, testimoniale, ticket modérateur, tierce opposition, tierce-opposition, tiers, tiers détenteur, titre exécutoire, tontine, torts, traduction des documents constitutifs du dossier d'adoption, trafic d'enfants et adoption internationale, traite, traité d'Amsterdam, transaction, transcription du jugement d'adoption, travail d'intérêt général, travaux dispensés, travaux soumis, tribunal, tribunal administratif, tribunal correctionnel, tribunal d'instance, tribunal de commerce, tribunal de grande instance, tribunal de police, tribunal des affaires de sécurité sociale, tribunal des affaires de sécurité sociale, tribunal des conflits, tribunal maritime commercial, tribunal paritaire des baux ruraux, tribunal pour enfants, trouble, trouble de voisinage, troubles de voisinage, troïka, tréfonds, trésor, tutelle, tutelle aux prestations sociales, tutelle de l'enfant mineur, tuteur, télécommunications, témoignages, témoignages ou présomptions, témoin, témoin assisté, uem, ueo, ultra petita, unanimité, unilatéral, union de l'Europe occidentale, union libre, union libre ou concubinage, union économique et monétaire, unité de planification et d'alerte rapide, unité de référence, usucapion, usufruit, usufruitier, usufruitière, usure, vacant, vaine pâture, valeur vénale, valeurs mobilières, vente, vente d'herbe, vente immobilière, verdict, viager, viagère, vice du consentement, vice-président, vices cachés, vices du consentement, victime, vider un délibéré, vie privée, violence, visa, visa d'entrée, voie d'exécution, voie de fait, voie de recours, voies d'exécution, voies de recours, voies de recours dites extraordinaires, voies de recours ordinaires, vrp, vues, vérification, vérité, warrant, warrants agricoles, zac, zad, zone agricole protégée, zone d'aménagement concerté, zone d'aménagement différé, zone de conflit, ès-qualités, échange, échec de l'adoption, échéance, écriture, éducateur, éducateur de la protection judiciaire de la jeunesse, éducation, efficacité internationale des décisions étrangères, égalité de traitement entre les hommes et les femmes, égalité des chances, élargissement, élire, élément matériel d'une infraction, élément moral d'une infraction, émancipation, émender, émolument, équilibre institutionnel et légitimité démocratique, établissement public d'aménagement urbain, établissement pénitentiaire, état, état civil, état civil de l'enfant adopté, état d'accueil, état d'origine, études d'esquisse, évaluation des biens, éviction, évocation

Annexe C

Exemple de navigation avec SEARCHXQ

Dans cette annexe, nous présentons un exemple de navigation avec l'outil SearchXQ.

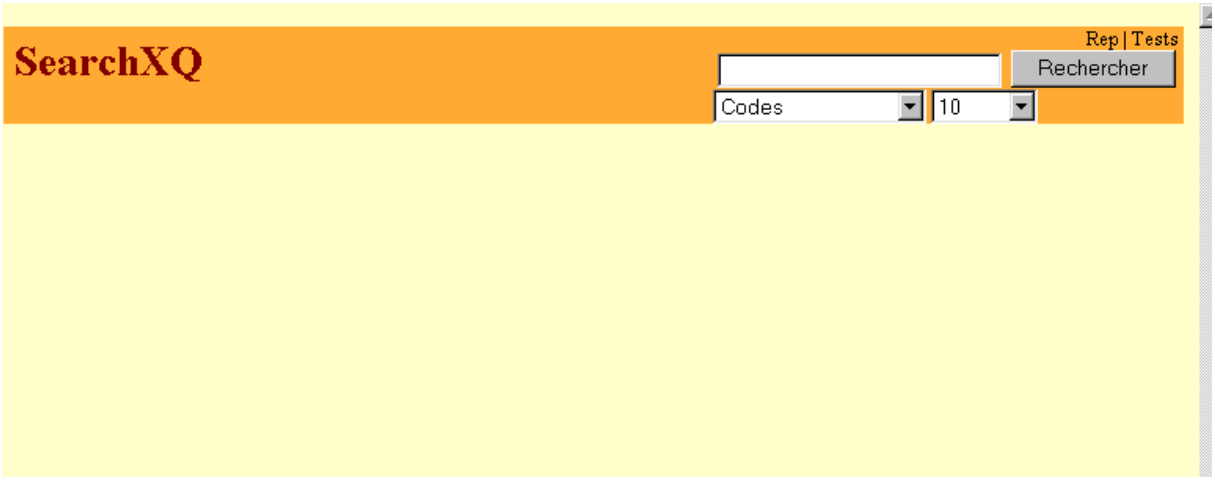

Figure C.1 – Page de garde de l'outil

La Figure C.1 présente la page de garde de l'outil. Seules deux options sont actuellement disponibles : la source du corpus et le nombre de documents en réponse par page. A noter que la mise en œuvre ne concerne, pour le moment, que les codes.

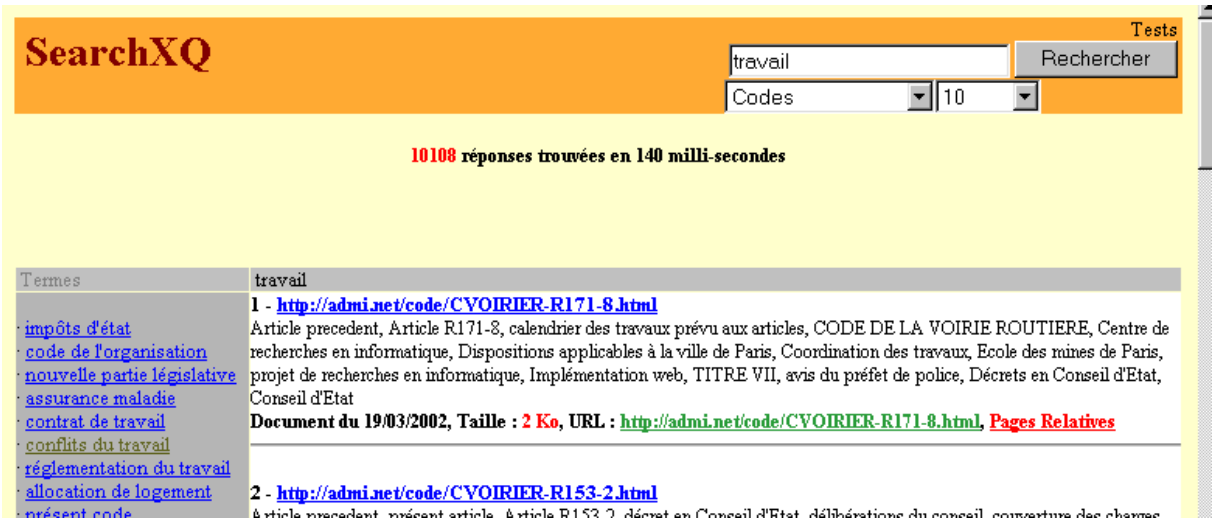

Figure C.2 – Réponse de la requête « travail »

La Figure C.2 présente la page en réponse à la requête « travail ». Pour cette requête, dix termes sont proposés (contrat de travail, réglementation du travail, etc.) et 10108 documents sont retournés en réponse.

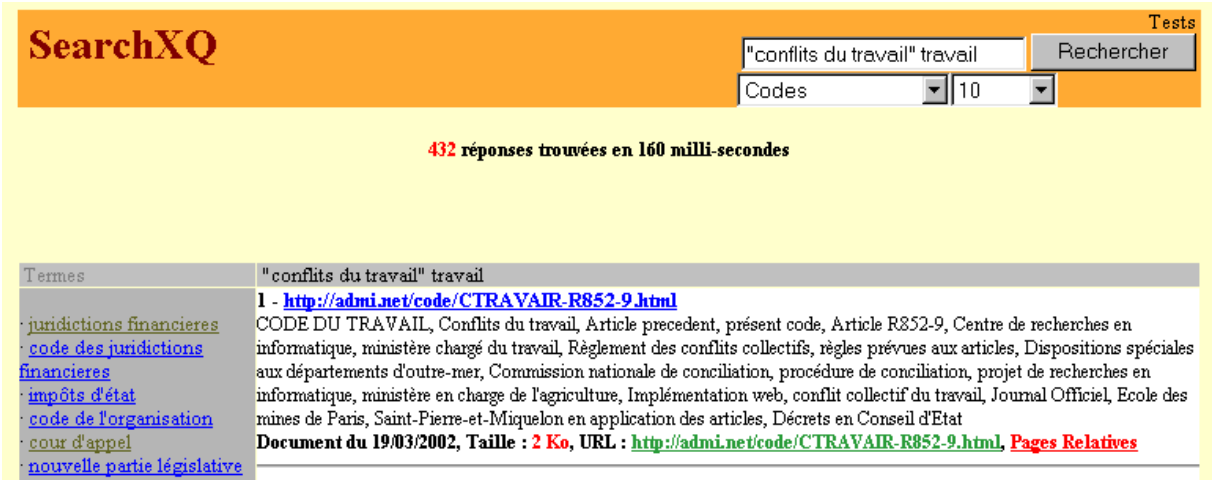

Figure C.3 – Réponse de l'expansion de la requête « travail » avec le terme « conflits de travail »

La Figure C.3 présente la page en réponse à la requête « travail » en choisissant le terme « conflits de travail ». Le nombre de documents en réponse a chuté fortement pour atteindre quelques centaines de documents : Ainsi, le ratio est d'environ 20.