UNIVERSITE IBN ZOHR

CENTRE DES ETUDES DOCTORALES IBN ZOHR

Formation doctorale Mathématiques Informatique et Applications

ETABLISSEMENT Faculté des Sciences

THESE

Présentée par

ALI IDARROU

Pour l'obtention du grade de

DOCTORAT NATIONAL

En cotutelle avec L'Université de Toulouse I Capitole, France

Spécialité : Informatique

Entreposage de documents multimédias : comparaison de structures

SOUTENUE LE 30/03/2013

Devant la commission d'examen composée de :

MOHAMED WAKRIM	Professeur à l'Université Ibn Zohr Agadir, Maroc	Président
JEAN-MARIE PINON	Professeur à l'INSA Lyon, France	Rapporteur
ABDERRAHIM SEKKAKI	Professeur à l'Université Hassan II, Casablanca Maroc	Rapporteur
CHANTAL SOULE-DUPUY	Professeur à l'Université de Toulouse I, France	Directrice de Thèse
DRISS MAMMASS	Professeur à l'Université Ibn Zohr Agadir, Maroc	Directeur de Thèse
NATHALIE VALLES PARLANGEAU	Maître de Conférences Université de Toulouse I, France	Co-encadrante

Résumé

Le volume de documents multimédias disponible aujourd'hui, et qui ne cesse d'augmenter, constitue une source d'information importante. Cependant, toute cette information serait sans intérêt si elle n'est pas exploitée efficacement. Gérer et exploiter de telles sources nécessite d'avoir à disposition des outils automatiques permettant de faciliter l'accès à des granules (l'information fine) documentaires, indépendamment de l'hétérogénéité sous-jacentes de ces documents en termes de type, taille, format, contenu, structure, etc. La classification automatique est une solution qui permet d'organiser et de structurer une large collection de documents afin de réduire l'espace de recherche et par conséquent d'améliorer les performances du processus d'accès à l'information. Les approches qui ont abordé la classification documentaire se distinguent par le modèle utilisé pour représenter les documents et par la démarche utilisée pour classer ces documents.

S'agissant des documents multimédias, la problématique de classification découle de la complexité de leur représentation. En effet, un *document multimédia* est composé de plusieurs objets de différentes natures : image, texte, son, etc. Il est multi-structuré par essence ; issu de la composition de plusieurs sous-documents et chaque sous-document a une ou plusieurs structures. Ces structures peuvent être de même nature ou de natures différentes (structure physique, logique, temporelle, etc). La multi-structuralité induit des relations complexes et multiples entre deux mêmes composants d'un document. Il est donc nécessaire d'utiliser un modèle de représentation riche afin de pouvoir classer les documents à structures multiples.

Les travaux de recherche que nous avons menés au cours de cette thèse visent ainsi à étudier les modèles de représentation des documents multimédias à structures multiples et à développer des outils capables de traiter de grandes masses de données en prenant en compte les contraintes liées au *partage* de sous-structures (*sous-graphes*) par des structures hétérogènes. Une des problématiques principales est de savoir comparer deux documents multi-structurés, et en conséquence de pouvoir *comparer des structures* de documents afin d'évaluer leur similarité. Nous nous sommes intéressés à la représentation des structures documentaires à l'aide des graphes. Comparer structurellement deux documents revient donc à comparer les graphes qui les représentent.

Les méthodes classiques de comparaison de documents sont basées sur les similarités dites de « *surface* » : un modèle de similarité basé sur les propriétés descriptives des objets sans tenir compte des relations entre ces propriétés. Ces méthodes ne tiennent pas compte de l'information implicite qui est véhiculée par la structure documentaire. Cependant, les mêmes composants structurels peuvent ne pas avoir le même rôle, ni la même importance dans deux documents différents. Nous pensons que les informations apportées par les *relations structurelles* présentent un intérêt incontournable dans un processus de comparaison. Nous montrons, par conséquent, que les mesures standards existantes ne peuvent pas répondre efficacement à notre problématique.

Pour évaluer la similarité entre deux graphes, nous proposons une *nouvelle mesure de similarité structurelle* basée sur l'isomorphisme de (sous) graphes. En théorie des graphes, l'isomorphisme de sous-graphes induits permet de démontrer qu'un graphe est inclus dans un autre, alors que l'isomorphisme de sous-graphes partiels permet de déterminer l'intersection entre deux graphes. Cependant, la recherche d'isomorphisme de sous-graphes est un problème bien connu pour être combinatoire. Ce problème combinatoire rend la plupart des approches limitées à des graphes de petite taille. Pour réduire le coût combinatoire, nous proposons de considérer un graphe comme un ensemble de chemins. Comparer deux graphes revient donc à comparer les chemins qui les composent, en utilisant un modèle à alignement structurel. La mesure proposée reflète la structure des

graphes comparés dans le sens où l'on tient compte à la fois de la position des nœuds, de l'ordre des nœuds frères et des liens entre ces nœuds.

Pour valider les propositions présentées dans ce mémoire de thèse, nous avons développé un outilde classification automatique non-supervisée d'un corpus de documents à structures multiples. Ainsi, la classification structurelle au sens où nous l'entendons permet de créer des classes appelées *vues génériques* représentant des documents structurellement proches (thèses, articles scientifiques, documentaires, rapports médicaux, etc).Les classes ne sont pas connues a priori, elles sont calculées automatiquement lors de l'intégration des documents, à partir de la mesure de similarité proposée. Le processus de classification doit permettre l'évaluation de la similarité d'une structure quelconque d'un document donné, avec chacune des *vues génériques* de l'entrepôt documentaire. Ceci pose des problèmes de temps de calculs et d'efficience de nos algorithmes. Nous avons proposé de restreindre l'espace de comparaison en utilisant une *présélection* des vues génériques susceptibles d'être similaires avec la structure du document à intégrer. Cela permet d'optimiser le temps de réponse de nos algorithmes de comparaison sans pour autant altérer la qualité de la classification.

Dans nos expérimentations, nous avons étudié les performances de nos algorithmes de classification en termes de qualité des classes générées et de temps de réponse.

Mots-clés : *document multimédia, multi-strcturalité, isomorphisme de sous-graphes, similarité structurelle, classification*.

A mon père que son âme repose en paix

A ma chère mère à qui je dois mon amour

A ma famille à qui je dois tout

A tous mes amis

Remerciements

Je suis particulièrement heureux de saisir cette occasion pour exprimer ma gratitude à toutes les personnes dont l'apport a été réel pour ma thèse. J'espère que les quelques mots que je m'apprête à écrire réussiront à retranscrire mes sentiments à l'égard de ces personnes.

J'adresse mes remerciements les plus sincères à mes directeurs de thèse Madame Chantal SOULE-DUPUY, Professeur à l'Université Toulouse I Capitole France et Monsieur Driss MAMMASS, Professeur à l'université Ibn Zohr Agadir Maroc, pour avoir dirigé et encadré cette thèse. Qu'ils soient assurés de ma profonde gratitude. Je suis notamment très reconnaissant à Monsieur Driss MAMMASS pour ses qualités humaines, ses orientations et ses encouragements et sa disponibilité en tant que responsable de l'unité de recherche et en tant que directeur de thèse. Je suis, de même, très reconnaissant à Mme Chantal SOULE-DUPUY. En effet, c'est d'abord elle qui m'a fait découvrir le domaine de la recherche dans le cadre de mes travaux de Master, à l'université de Paul Sabatier Toulouse France, avant de diriger ma thèse. Qu'elle soit assurée de ma profonde reconnaissance pour la qualité de la formation dont elle m'a fait bénéficie.

Je tiens à remercier sincèrement, par ailleurs, Madame Nathalie VALLES-PARLANGEAU, maître de conférences à l'Université Toulouse I Capitole pour avoir encadré et suivi cette thèse, pour ses encouragements, sa disponibilité, sa contribution fructueuse et ses remarques de qualités. C'est d'ailleurs elle qui a encadré mes travaux de Master avant d'encadrer cette thèse. Ses conseils précieux et ses nombreux commentaires m'ont permis non seulement d'améliorer mon travail mais aussi d'élargir mon horizon. Qu'elle soit assurée de mon très grand respect.

Mes remerciements les plus respectueux vont aussi à Monsieur Jean-Marie PINON, professeur à l'INSA de Lyon France et à Monsieur Abderrahim SEKKAKI professeur à l'Université Hassan II Casablanca Maroc qui ont accepté d'évaluer ce travail et d'en être rapporteurs. Je les remercie également pour l'honneur qu'ils m'ont fait en participant au jury de ma thèse.

Je remercie sincèrement Monsieur Mohamed WAKRIM, Professeur à l'université Ibn Zohr Agadir Maroc, d'avoir rapporté mes travaux et d'avoir présidé le Jury de ma soutenance de thèse.

Je remercie vivement Monsieur Driss MAMMASS, responsable de l'unité de recherche à la Faculté des sciences d'Agadir Maroc pour m'avoir accueilli. Je tiens à remercier de même Mme Josiane MOTHE responsable de l'équipe SIG au laboratoire IRIT (Institut de Recherche en Informatique de Toulouse), pour m'avoir accueilli au sein de cette équipe afin de mener à bien cette Thèse. Je remercie Monsieur le Doyen, de la faculté des sciences d'Agadir Maroc.

Je remercie, également, tous les membres du centre doctoral de l'Université d'Ibn Zohr Agadir Maroc. Je remercie de même tous les membres des écoles doctorales MITT : Mathématiques, Informatique et Télécommunications de Toulouse France, en particulier Mme Virgine MANGION.

Je voudrais enfin exprimer ma profonde gratitude à mes professeurs de l'Université de Toulouse 1 Capitole France et de l'Université de Paul Sabatier Toulouse France qui ont contribué à ma formation en Master de Recherche.

Agadir le : 30/03/2013

Ali Idarrou

Table des matières

Liste des tableaux

Chapitre I

Chapitre III

Chapitre IV

Liste des figures

Chapitre I

Chapitre II

Chapitre III

Chapitre IV

Introduction générale

I. Contexte et problématique

L'émergence de l'information multimédia a donné naissance au document multimédia numérique, fédérant plusieurs médias (texte, son, image, animation, graphique 2D/3D, etc) de natures différentes, qui étend le rôle et la nature du document textuel. La diversité des composantes d'un document multimédia constitue la richesse mais aussi la complexité de ce type de document. L'information n'est pas aussi directement accessible que dans des documents textuels par exemple. Le problème est donc d'accéder à l'information fine (des fois non explicite), jugée pertinente par l'utilisateur, noyée dans une grande masse de données hétérogènes et de sources différentes. Il est donc nécessaire de disposer d'outils automatiques permettant de mieux tenir compte à la fois de l'hétérogénéité de ces données et du temps pour les retrouver. Les techniques de classification ont été utilisées comme solution pour permettre l'accès ciblé à des granules spécifiques dans une large collection de documents. La classification est un processus d'organisation de documents visant l'optimisation du processus de restitution [Soulé-Dupuy C., 2001]. Pour classer un corpus de documents multimédias, il est nécessaire d'avoir un modèle de représentation en adéquation avec ce type de documents. Plus la modélisation des documents sera sophistiquée et plus la comparaison de ces documents sera pertinente mais difficile [Sorlin S., 2006].

Les questions classiques qui reviennent souvent en classification documentaire sont : Comment représenter les documents afin de les classer ? En quoi deux documents sont-ils similaires ? Que signifie la similarité de deux documents ? Comment les comparer ? A quel point peut-on dire que deux documents sont similaires ? Comment valider les résultats d'une classification automatique, lorsque le nombre de documents est important ?

L'évaluation de la proximité de deux objets est un problème qui a motivé de nombreux travaux. Dans ce contexte, plusieurs mesures de similarité ou de distance ont été proposées pour des objectifs spécifiques et dans des contextes variés. Cependant, il est difficile de définir a priori une distance universelle [Bisson G., 2008]. Dans [Champclaux Y., 2009], quatre grandes familles de modèles de similarité peuvent être envisagées : (1) les modèles basés sur les caractéristiques (ou attributs). (2) les modèles géométriques, (3) les modèles à alignement structurel et (4) les modèles basés sur la notion de distance transformationnelle.

La diversité des modèles de similarité, suscite une problématique liée au choix du modèle le plus approprié. Généralement, Ce choix dépend à la fois du contexte, de la nature des objets à comparer, des objectifs visés, etc. Par exemple, dans [Humel J.E., 2001], il a été montré que les modèles géométriques et les modèles basés sur les attributs ne permettent pas la comparaison des objets structurés.

Les méthodes classiques de comparaison de documents sont basées sur les similarités dites de « *surface* », elles considèrent le document comme un ensemble de composants. Ces méthodes ne tiennent pas compte de l'information implicite qui est véhiculée par la structure documentaire. Pourtant, les mêmes composantes peuvent ne pas avoir le même rôle ni le même contexte dans deux document différents. Plus précisément, dans un document, le sens ne concerne pas seulement la signification des éléments structurels de ce document mais il concerne également celle des relations, porteuses d'informations implicites, entre ces éléments. Ignorer la structure du document revient à ignorer sa sémantique [Schlieder T., et al. 2002].

Les documents multimédias sont multi-structurés par essence : issus de la composition de plusieurs sous-documents et chaque sous-document a une voire plusieurs structures. La

muti-structuralité induit des relations complexes et multiples entre deux mêmes composantes d'un document. Cela impose de ne plus voir un document multimédia sous forme d'arbre simple mais plus tôt sous forme de graphe.

Les graphes sont des structures de données ayant la capacité de représenter des objets complexes et structurés. La théorie mathématique des graphes mise au point pourrait présenter un grand intérêt quant à l'évaluation de la similarité des documents, aussi bien en recherche d'information qu'en classification documentaire. Dans [Sorlin S., et Solnon S., 2005], l'isomorphisme de sous-graphe peut être utilisé pour montrer l'inclusion ou l'équivalence de deux graphes. Cependant, la recherche d'isomorphisme de sous-graphes est un problème combinatoire. L'enjeu majeur à ce niveau est de réduire cette combinatoire pour faire avancer cet axe : un problème ouvert qui a motivé un nombre important de travaux.

II. Propositions et contributions

Nous avons utilisé le modèle *MVDM* (*Multi Views Document Model,* chapitre I, section VII.2.3 page 40) pour décrire les documents multimédias à structures multiples. Ce méta-modèle est composé de deux niveaux : un niveau générique et un niveau spécifique. Une vue est dite « spécifique » lorsqu'elle décrit un document particulier, alors qu'une vue générique représente un ensemble de vues spécifiques pour des documents structurellement proches. Nous nous sommes intéressés à la représentation des documents multimédias à l'aide des graphes afin de comparer leurs structures.

Pour évaluer la proximité entre deux graphes, nous avons proposé une *nouvelle mesure* de similarité structurelle basée sur l'isomorphisme de (sous) graphes et reposant sur une fonction de pondération des graphes que nous avons introduit. Cette dernière permet d'exprimer des contraintes liées aux aspects hiérarchique et contextuel, des composantes (nœuds et arcs), dans la mesure où elle tient compte de la répartition de ces composantes dans le graphe et de la nature des relations entre ces composantes. Cela permet de refléter à la fois la structure et la signification des documents comparés. La mesure proposée est structurelle et non une mesure de *surface.* Dans [Gentner D. et al., 1989] la similarité dite de *surface* est basée sur les propriétés descriptives des objets alors que la similarité structurelle entre objets est évaluée sur la base des relations entre ces objets. Les mesures de *surface* ne tiennent pas compte de l'information implicite véhiculée par la structure documentaire et qu'on ne peut pas ignorer dans un processus de comparaison.

Ainsi, les mesures standards existantes ne peuvent pas répondre efficacement à notre problématique.

Deux documents sont structurellement similaires si leurs structures présentent un degré de ressemblance qui dépend d'un *seuil de similarité ;* un paramètre fixé a priori par l'utilisateur.

Nous avons choisi de considérer un graphe comme un ensemble de chemins. Cela permet de réduire le coût combinatoire engendré par la recherche d'isomorphisme de graphes. La comparaison de deux graphes revient donc à comparer les chemins qui les composent.

L'intégration d'un document dans l'entrepôt documentaire passe par un processus de comparaison de la structure du document avec chacune des *vues génériques* de l'entrepôt. Ceci pose des problèmes de temps de calculs et d'efficience de nos algorithmes. Ainsi, nous avons proposé un *sous-processus de filtrage* (*présélection*) des *vues génériques* de

l'entrepôt susceptibles d'être similaires à la structure du nouveau document. Cette restriction de l'espace de comparaison permet d'optimiser le temps de réponses de nos algorithmes tout en conservant la qualité des classes générées.

Les représentants de classes (*vues génériques*) peuvent être enrichis (ajout des fragments) au fur et à mesure de la classification : *transformation des vues*. Cela permet d'augmenter la représentativité des classes et par conséquent optimiser le volume de stockage de l'entrepôt documentaire. Cependant, cette transformation peut engendrer un autre problème : rapprochement des classes (diminuer la distance inter-classe) et par conséquent perturber la classification. En effet, le rapprochement des classes peut susciter un autre problème : l'appartenance d'un même document à deux classes différentes. Pour maintenir la stabilité des classes, nous proposons de fixer a priori une *distance minimale inter-classe*.

La validation des résultats d'une classification automatique est un problème qui a attiré l'attention de plusieurs travaux. Nous citons par exemple les travaux de [Genane Y., 2004], [Beney J., 2006] et [Yosr N. et Sinaoui 2009].

Dans les travaux de [Oh,Il-Seok et al., 1999], une séparation plus large des classes implique un meilleur pouvoir discriminant. Deux objets lointains représentent des données qui appartiennent à des groupes différents [Bisson G., 2000]. En effet, la séparation des classes est l'un des critères d'une classification de qualité. Augmenter la distance entre classes permet de diminuer le bruit et augmenter la précision de la classification. Dans ce contexte, la prise en compte d'une *distance minimale inter-classe* permet de conserver une distance entre les classes et par conséquent éviter le rapprochement des classes, qui peut être engendré par la transformation des vues génériques.

III. Organisation du mémoire

Ce mémoire se compose de quatre chapitres regroupés en deux parties. La première partie est consacrée à l'état de l'art, elle permet de décrire le contexte de nos travaux et présente également un panorama des travaux liés aux documents multimédias à structures multiples. Cette partie se compose de deux chapitres.

Nous proposons au chapitre I un état de l'art sur les documents multimédias à structures multiples dans lequel nous exposons les concepts de base, liés au document multimédia numérique. Ce chapitre permet également de présenter les structures documentaires les plus connues dans la littérature, les standards de structuration documentaire, les modèles de représentation des structures documentaires, la problématique de la multi-structuralié et les solutions de la représentation des documents à structures multiples.

Le chapitre II est composé de trois parties : la première partie est consacrée à des notions de base sur la comparaison des graphes. Dans la deuxième partie de ce chapitre, nous faisons un bref aperçu sur l'état de l'art concernant la similarité des objets. La troisième partie de ce chapitre est dédiée à la classification en général. Nous présentons ensuite un panorama de travaux connexes sur la classification documentaire. En fin, nous donnons un bref aperçu sur quelques techniques de validation des résultats d'une classification automatique.

La deuxième partie de ce mémoire est réservée essentiellement à notre approche, elle est composée de deux chapitres :

Dans le chapitre III, nous exposons la mise en œuvre d'une méthode de classification structurelle des documents multimédias. Nous présentons dans la première partie de ce chapitre notre mesure de similarité basée sur l'isomorphisme de sous-graphes. Dans la

deuxième partie, nous détaillons notre processus de classification structurelle de documents reposant sur la mesure de similarité que nous avons proposé.

Le chapitre IV présente l'évaluation de nos propositions. L'objectif de ce chapitre est la validation de notre approche. Il présente l'implantation des propositions de ce mémoire de thèse au travers du prototype *MMDocRep « Multistructured Multimedia document Repository »*. Ce prototype est constitué d'une application Java qui interagit avec le Système de Gestion de Base de Données objet relationnel Oracle 10g2. Au travers de ce prototype, nous validons notre classification et nous montrons la faisabilité des démarches d'intégration et de comparaison de documents multimédias à structures multiples. Nous présentons un ensemble d'expérimentations qui va permettre d'évaluer nos propositions.

Première Partie - État de l’art

Chapitre I : Document multimédia : Notions et concepts - Etat de l'art

Résumé du chapitre **:**

Dans ce chapitre, nous abordons une recherche bibliographique et nous présentons un état de l'art au travers du quel nous exposons, dans la deuxième section, les concepts de bases liés au document multimédia numérique, que nous allons utiliser dans ce travail. Dans la troisième section de ce chapitre, nous décrivons les structures documentaires les plus connues dans la littérature. Dans la quatrième section, nous exposons brièvement les standards de structuration documentaire. La cinquième section de ce chapitre est consacrée aux modèles de représentation des structures de documents. La sixième section vise à présenter la problématique de la multi-structuralité documentaire. En fin, la septième section présente quelques solutions de la représentation des documents à structures multiples.

Sommaire du chapitre I

I. Introduction

Depuis bien longtemps, l'objet document est considéré comme moyen instructif ; c'est un vecteur de connaissance et porteur d'informations. Selon [EGYED Z.E., 2003], un document véhicule de l'information qui peut être représentée sous forme de texte, d'image, de son, etc.

Avec l'essor technologique, le document a connu un changement radical en passant de l'ère du papier à l'ère du numérique. Selon [Caro S., 2003], les principales différences entre les deux supports, papier et numérique, se déclinent selon quatre points de vue : matériel (propriétés et caractéristiques du support), cognitif (représentation de la structure, l'accès, etc), physique (parcourir le document) et usage. Ce changement a révolutionné la manière de communiquer, d'échanger, de partager, de stocker et d'accéder à l'information numérique. Il a contribué à la disponibilité d'une immense quantité de documents dans les archives personnelles et professionnelles, en ligne (Internet et Intranet), etc. Le format numérique du document le rend plus pratique, plus souple et plus facile à échanger, manipuler et à réutiliser. Ce qui s'explique par son utilisation massive jusqu'à faire de lui un objet inéluctable dans tous les domaines d'applications : académiques, industriels et personnels. En revanche, de nouvelles problématiques liées à la grande diversité de cet usage ont émergés. Par exemple, la conservation du document, son évolution, son traitement par l'agent humain ou automatique, son exécution sur différentes plates-formes : écrans d'ordinateurs, PDAs, IPADs, téléphones, lecteurs personnels, écrans TV, etc. D'où la nécessité adapter le document aux nouveaux contextes. En parallèle, avec l'évolution des documents multimédia, les standards documentaires évoluent et prolifèrent afin de s'adapter et de répondre à ces besoins.

Nous présentons, dans ce chapitre, un certain nombre de notions de base, essentielles liées aux documents multimédia au niveau de : sa composition et sa de structuration. Plusieurs types de structures peuvent être utilisés pour un même document. Ces structures peuvent être de même nature ou de natures différentes (logique, physique, sémantique, etc). Nous présentons aussi les structures documentaires les plus connues dans la littérature. Nous évoquons également la problématique de la multi-structuralité. Nous faisons un bref tour d'horizon sur les standards de structuration et nous discutons de quelques solutions concernant la représentation des documents à structures multiples. Enfin, nous présentons les structures de données (vecteur, arbre, graphe, etc) les plus utilisées pour représenter les structures documentaires.

II. Document multimédia

II.1. Media, multimédia et hypermédia

En latin, « *media* » est le pluriel du terme *medium*. Les puristes utilisent, « *medias* » pour désigner plusieurs supports et « *media* » pour désigner un support unique. Le terme « *media* » peut être utilisé pour désigner tout moyen de diffusion massive. Dans [Bui Thi M.P., 2003], un média est une forme logique qui est capable de véhiculer de l'information. [Scuturici M., 2002] définit le terme « média » comme moyen de : diffusion massive de l'information, de représentation de l'information, de perception de l'information (vue, ouïe, etc) et de stockage de l'information.

Le terme « multimédia » désigne plus d'un « medium ». Multimédia désigne aussi une technique apparue à la fin des années 80, permettant de combiner plusieurs media (texte, image, son, etc) pour représenter un ensemble cohérent d'informations. Le terme *multimédia* se rapporte à une communication à travers plusieurs types de média [Thuong T.T., 2003].

L'*hypermédia* est un concept qui découle du multimédia auquel s'ajoute de l'interactivité et par conséquent augmente la richesse des documents multimédias. Les composants des documents multimédias sont liés de façon à assurer une navigation aisée et interactive dans un espace diversifié d'informations. Cette technique permet d'augmenter le sens, la qualité, l'accessibilité des documents multimédias et les rend facilement manipulables.

II.2. Document, document numérique

Le terme document vient du mot latin "*documentum*", qui veut dire « *ce qui sert à instruire* ». Pour [Bachimont B., 1998], un document est un objet matériel exprimant un contenu. Ce contenu est exprimé sur un support d'inscription (l'objet matériel). Un document est un ensemble constitué d'un support d'information, des données existant sur ce support et de leur signification [Afnor, 1987]. D'après le Larousse, un document est *« tout renseignement écrit ou objet servant de preuve ou d'information »*. L'ISO (International Standard Organisation) définit le document comme l'ensemble d'un support d'informations et de données enregistrées sur celui-ci sous une forme en général permanente et lisible par l'homme ou par une machine. Le Petit Robert définit le document comme étant un renseignement écrit ou objet servant de preuve, d'information ou de témoignage. Dans [Roisin C., 1999], un document est porteur du sens : transmission de connaissance, de savoir, d'information. Il est conçu par un auteur dont le but est d'être reçu par un ou plusieurs lecteurs. C'est une forme pérenne de l'information. Pour [Ramel J.Y., 2006], un document est un ensemble organisé d'éléments de contenu : Structure logique et sémantique.

Dans [Bachimont B., 1998], un document est dit numérique quand le support matériel d'inscription devient numérique. Dans [Thuong T.T., 2003], le document numérique est l'un des principaux fondements caractérisant l'ère de l'information. Pour [Bachimont B., 1998] et [Bachimont B., 2004], un document sur un support numérique est une source à partir de laquelle peuvent être calculés autant de documents. Cela s'explique par le fait que la source en question ne peut être consultée qu'à partir d'un programme pouvant calculer à partir de cette source une forme intelligible du document. Le document numérique n'est pas un document, c'est une source à partir de laquelle un document peut être reconstruit sur un support d'appropriation. C'est une entité dynamique qui n'existe que lors de la projection. Selon [Bachimont B., 1999] un document est un objet matériel exprimant un *contenu*. Il est indissociable d'un *support matériel* (un écran, une feuille de papier, etc.), support d'inscription où un contenu est exprimé. Un document numérique peut être considéré comme un ensemble de données et des méthodes de présentation [Chaudiron et al., 2000]. Dans [Roger T., 2003], l'instrumentation numérique des contenus modifie profondément la nature des documents.

II.3. Document multimédia

Le document multimédia étend le document textuel (mono-média). Il combine plusieurs technologies : le texte, l'image, le son (exemple figure I.1), etc. Ce type de document est caractérisé essentiellement par les composants de différentes natures qu'il contient et des possibilités de les gérer de façon interactive et cohérente à l'aide des liens sémantiques inter et intra document. Selon [Ramel J.Y., 2006], un document multimédia est un enchaînement temporel d'éléments de contenu. Pour [Mbarki M., 2008], un document multimédia est l'agencement interactif dans le temps et dans l'espace d'éléments distincts de natures différentes (issus de plusieurs types de média) afin d'enrichir le contenu de l'information diffusée. Pour [Bachimont B., 1988], il s'agit d'un document mêlant d'une part différents médias et d'autre part construit à partir de différents sous-documents unis par un réseau de liens. Pour définir une présentation multimédia dans laquelle des objets médias sont organisés temporellement, [Ramel J-Y., 2006], utilise la définition suivante : un document est un enchaînement temporel d'éléments de contenu. Pour cet auteur, la plupart des définitions existantes ne couvrent que partiellement la notion de document.

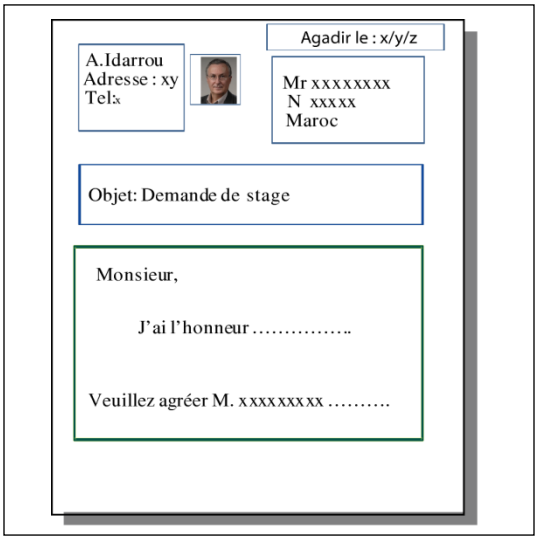


Figure I.1 - Exemple de document multimédia

II.4. Métadonnée

Une métadonnée est une composante cruciale dans la gestion du contenu numérique. C'est une donnée sur une donnée sous forme d'un ensemble structuré d'informations. Elle permet de décrire une autre donnée en lui donnant plus de sens afin d'obtenir une description plus complète de celle-ci. Dans [Vellucci L., 1998], les métadonnées sont les informations utilisées pour la description et la gestion des ressources. Pour [Ercegovac Z., 1999], les métadonnées sont des informations qui permettent de décrire, d'identifier et de définir une ressource. Plus généralement, les métadonnées peuvent être utilisées pour décrire les ressources numériques, les enrichir, faciliter leur accès, les partager, les réutiliser, etc.
Une métadonnée est un indicateur porteur de sens qui est rajouté au sein d'un document pour souligner une idée, une information [Abascal R., et al. 2007].

Après avoir donné une introduction sur le document multimédia numérique, il est judicieux de s'interroger sur les objets qui composent ce type de documents. C'est ce que nous allons aborder dans la section suivante.

II.5. Composants d'un document multimédia

La complexité des documents multimédia vient du fait qu'ils mêlent des objets de natures différentes, et ayant des caractéristiques à la fois spécifiques et diverses d'un objet à l'autre (figure I.2). La façon dont un document multimédia a été composé, l'organisation de ses composants eux même simples ou composés, leurs agencements interactifs dans le temps et dans l'espace, les règles posées par l'auteur et enfin les contraintes supposées pour son exploitation selon différents objectifs ont fait de ce type de document un objet complexe et difficile à appréhender. Selon [Laborie S., 2008], un document multimédia est composé d'un ensemble d'objets multimédia et un objet multimédia est une entité faisant référence à une ressource pouvant être : texte, audio, image ou vidéo.

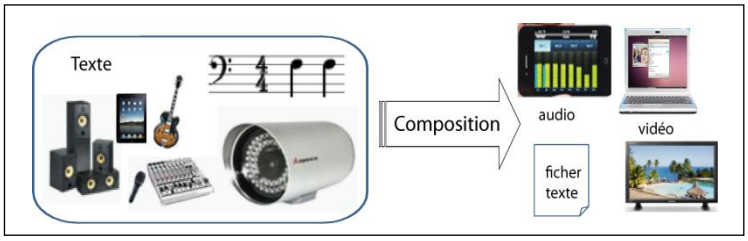


Figure I.2 - Exemple de composants d'un document multimédia

Nous présentons ci-dessous quelques définitions des composants de base les plus utilisés d'un document.

a) Le texte

C'est le medium le plus connu. Un texte est un assemblage de caractères organisés pour exprimer une idée qui dépend de l'intention de son auteur. Pour [Marcoux Y., 1994], un texte désigne un ensemble d'informations qui représente une unité pouvant être considérée comme indivisible et complète. La représentation et le traitement du texte sont assurés par plusieurs logiciels standards tels que RTF (Rich Text Format), PDF (Portable Document Format), PS (Postscript), LaTeX (Lamport TEX), etc. Ces standards permettent de faire des opérations (fastidieuses à effectuer sur le document papier) sur le document numérique telles que la mise en forme, la mise à jour, la recherche et l'accès rapide, etc. Ils permettent d'assurer l'évolution du document électronique, son intégrité, sa réutilisabilité, sa portabilité, etc. La notion d'hypertexte permet d'assurer la navigation, prévue par l'auteur, au sein des documents et inter documents. Elle permet au lecteur un parcourt aisé pour faciliter l'accès direct à des passages d'un document. L'hypertexte permet de franchir les limites physiques fixées par le document papier concernant le parcours de celui-ci.

b) L'image

Selon le dictionnaire Larousse 2009, le terme *image* est définit comme suit : « *Représentation d'un être ou d'une chose par les arts graphiques* ». C'est une composante prépondérante dans les documents multimédias, elle est indépendante de la langue et parfois plus riche que le texte. L'origine des documents contenant des images remonte aux années 1950 [Bres S. et al. 1999]. Les images sont utilisées, sous forme de dessins, depuis

plus de trente mille ans. Elles étaient utilisées comme moyen de communication et d'expression, pour illustrer des personnes préhistoriques, ou pour concrétiser une idée. Une image désigne la représentation visuelle d'un objet par différents moyens ou supports. C'est un medium de grande importance utilisé dans de nombreux domaines tels que l'enseignement, la publication, la médecine, etc. Une image numérique, est une image qui se présente sous la forme d'une matrice de pixels ou, sous forme vectorielle. Elle peut être présentée et codée selon plusieurs formats [Roxin I. et Mercier D., 2004]. Parmi les plus répandus, nous pouvons citer les formats :

- BMP (BitMap image matricielle : crée par Microsoft),
- GIF (Graphics Interchange Format : format d'échange d'images mis au point en 1987/1989 pour permettre le téléchargement d'images en couleur : développé par Compuerve).
- JPEG ISO/CEI 10918-1 (Joint Photographic Expert Group), ayant les mêmes avantages que le format GIF. Cependant il est mieux adapté aux images en couleurs grâce à sa technique de compression,
- TIFF (Tagged Image File Format),
- PNG (Portable Network Graphics),
- JPEG2000 ISO/CEI 15444-1,
- SPIFF (Still Picture Interfrange File Format),
- MNG (Multiple-image Network Graphics),
- SVG (Scalable Vector Graphic),
- etc.

Une vidéo peut être considérée comme une succession d'images fixes (avec éventuellement du son) à une certaine cadence. L'animation consiste à donner l'illusion d'un mouvement à l'aide d'une suite d'images.

c) Le son

C'est un médium très populaire qui permet de véhiculer une information sous la forme sonore. Cette information est le résultat d'un phénomène physique : une onde produite par une vibration rapide de compression et de dépression du milieu dans lequel il se propage. Il se caractérise par sa fréquence, son amplitude et son timbre. Ce medium requiert de gros volumes ce qui explique pourquoi nous le trouvons généralement sous un format compressé. Parmi les formats audio les plus connus, nous citons le format Wave (WAREform audio format) et le format MP3 (Mpeg-1 audio Layer 3), 3GP une version simplifiée de MPEG-4 utilisée dans beaucoup de téléphones portables, etc.

Dans un document multimédia, le son est une source d'information pouvant jouer le rôle complémentaire (compléter le geste par exemple) ou supplémentaire d'autres modalités composant ce document.

Après un aperçu global sur les composants d'un document, dans la section suivante, nous allons aborder la structuration documentaire.

III. Les structures documentaires

III.1. Introduction

Le concept structure vient du latin « *struere* » qui signifie construire et agencer. L'agencement entre les composants d'un document peut être assuré par plusieurs types de relations (hiérarchiques, spatiales, temporelles, etc). Dans le dictionnaire le Larouse, une structure est la manière dont les parties d'un ensemble concret ou abstrait sont organisées entre elles. Dans [Abascal R. et al. 2004], une structure documentaire est une description d'un document par un ensemble d'éléments en relation les uns avec les autres, au cours ou en vue d'un usage. Les travaux sur les documents ont envisagé trois classes de documents :

III.1.1. Document plat

C'est un document qui ne dispose pas de structures explicites à part la ponctuation. [Tannier X., 2006] considère que tout texte ne comportant pas plus que des marquages de ponctuations et/ou de présentation (passages en lignes, espacements divers, énumérations, etc) est un document plat. Selon [Bringay S. et al., 2004], un document plat est un document pour lequel on ne peut pas détecter la structure. Dans [Portier P.E., 2010], les données non structurées sont des données qui peuvent être de n'importe quel type : qui ne semblent pas suivre quelques formats ou règles.

III.1.2. Document structuré et semi-structuré

Un document structuré est un document ayant une structure régulière, il contient des informations dont on connaît le contenu et l'emplacement. Il s'agit par exemple de CVs, de questionnaires, de formulaires, etc. Plus précisément un document structuré est un document dont la structure est connue a priori.

Le document semi-structuré contient des informations dont on connaît le contexte, mais pas exactement l'emplacement. Sa structure n'est pas connue a priori mais elle peut être définie a posteriori. Un document semi-structuré contient donc des informations implicites sur la structure. Les documents semi-structurés ne respectent pas un schéma fixe et ils sont caractérisés par des structures qui peuvent être :

- flexibles,
- irrégulières : des données manquantes ou en plus,
- implicites : les structures sont définies dans les données, on doit parser les documents pour les découvrir,
- partielles : une partie des données est sa structure (texte, images, son, …),
- etc.

Les documents semi-structurés permettent une saisie plus naturelle et un échange plus simple de l'information [Bryan M., 1998]. Dans la littérature, les documents XML (eXtensible Markup Language) sont souvent qualifiés de documents semi-structurés.

Dans la section suivante, nous présentons les structures documentaires les plus citées dans la littérature.

III.2. Typologie des structures documentaires

Les documents textuels sont caractérisés par une organisation logique et hiérarchique, un agencement dans l'espace entre ses composants et peuvent être exploités manuellement ou d'une manière automatique. Les documents multimédia ont en plus des spécificités particulières concernant leurs présentations multimédia, dans laquelle les composants (texte, son, image, etc) sont organisés dans l'espace et dans le temps pour pouvoir répondre à des objectifs généralement particuliers. Selon [Roisin C., 1998], il existe quatre dimensions principales : (1) temporelle (synchronisation des objets multimédia dans le temps), (2) spatiale (disposition des objets multimédia dans l'espace), (3) logique (regroupement de certains objets multimédia sous une même entité) et (4) hypermédia (liens sémantiques pour naviguer dans le document multimédia). L'utilisation multiple d'un document nécessite plusieurs présentations de ce même document (figure I.3).

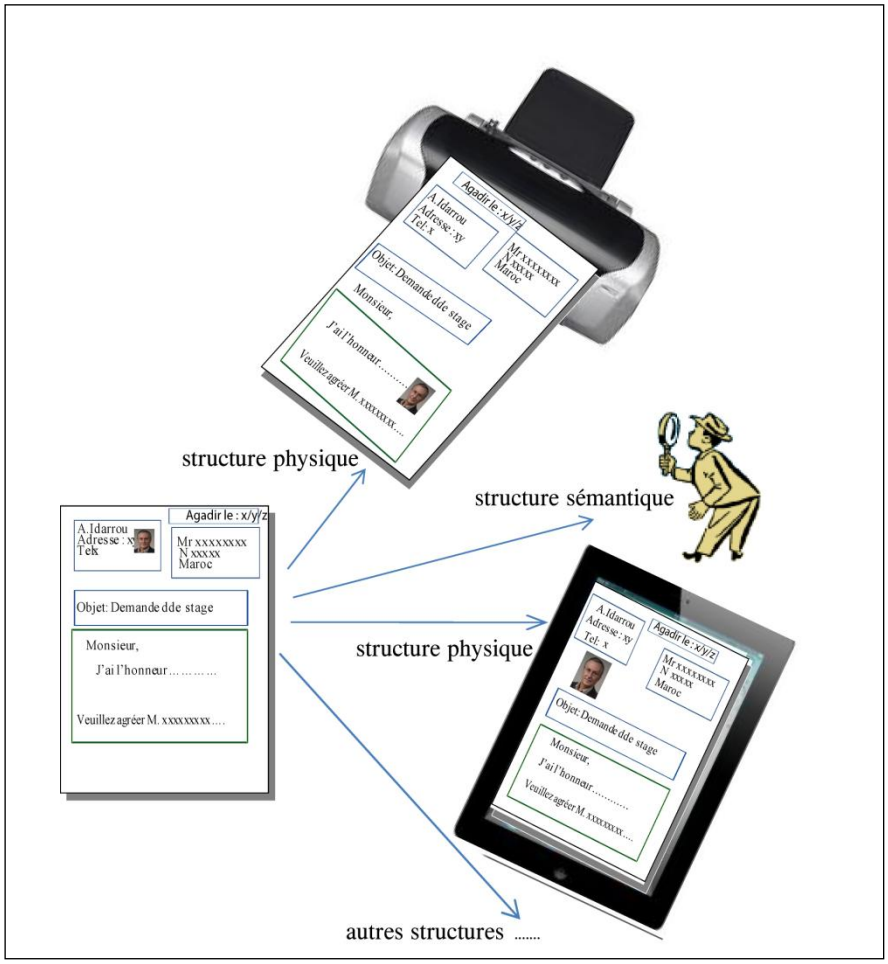


Figure I.3 - Utilisation multiple du même document

A notre connaissance, il n'existe pas une topologie exhaustive et normative de tous les types de structures documentaires, généralement la structure d'un document répond à l'objectif pour lequel elle a était définie.

Dans cette section, nous allons évoquer les structures documentaires les plus connues dans la littérature et, pour faciliter la compréhension du modèle de document multimédia,

nous allons mettre en évidence à travers un exemple les spécificités de chacune de ces structures.

III.2.1. La structure logique

La structure logique d'un document décrit l'organisation hiérarchique du contenu de ce document (exemple figure I.4). Elle permet de lier, de façon sémantique, l'ensemble des composants de ce document. La structure logique permet de refléter l'organisation explicite (par exemple : titres, chapitres, sections, etc) d'abstractions logiques représentant des parties de document [Fourel F., 1998].

Dans [Roisin C.,1999], un document est un ensemble d'objets organisés hiérarchiquement dont la structure logique s'appuie sur trois entités : (1) les éléments de base non décomposables qui constituent le contenu, (2) les éléments composites obtenus par composition d'éléments de base ou d'autres éléments composites et (3) les attributs qui peuvent être associés aux éléments pour leur adjoindre des informations supplémentaires.

Pour [Mbarki M., 2008], « La structure logique permet un découpage de l'information d'un point de vue hiérarchique selon un principe de décomposition plus ou moins fin. Ce mécanisme impose d'identifier de façon non ambiguë les granules d'information composant le document ».

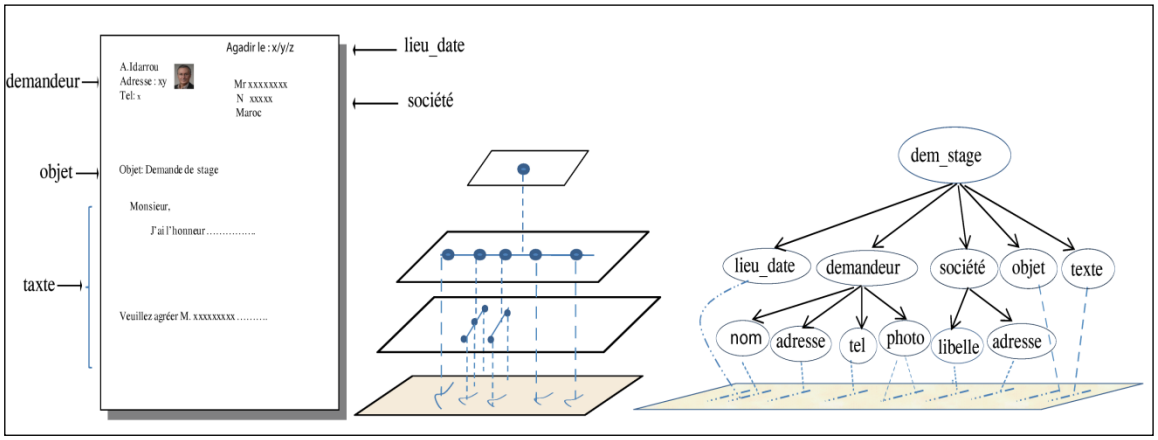


Figure I.4 - Exemple de structure logique

III.2.2. La structure physique

La structure physique d'un document définit le positionnement de ses composants (éléments physiques : lignes, colonnes, pages, blocs, etc) ainsi que leur agencement les uns par rapport aux autres (exemple figure I.5). Elle traduit l'organisation des éléments de la structure logique sur un support de présentation en respectant un certain ensemble de règles tout en tenant compte des média de restitution (papier, écran du terminal, etc). La structure physique peut le plus souvent se déduire de la structure logique [Chatti N., 2007].

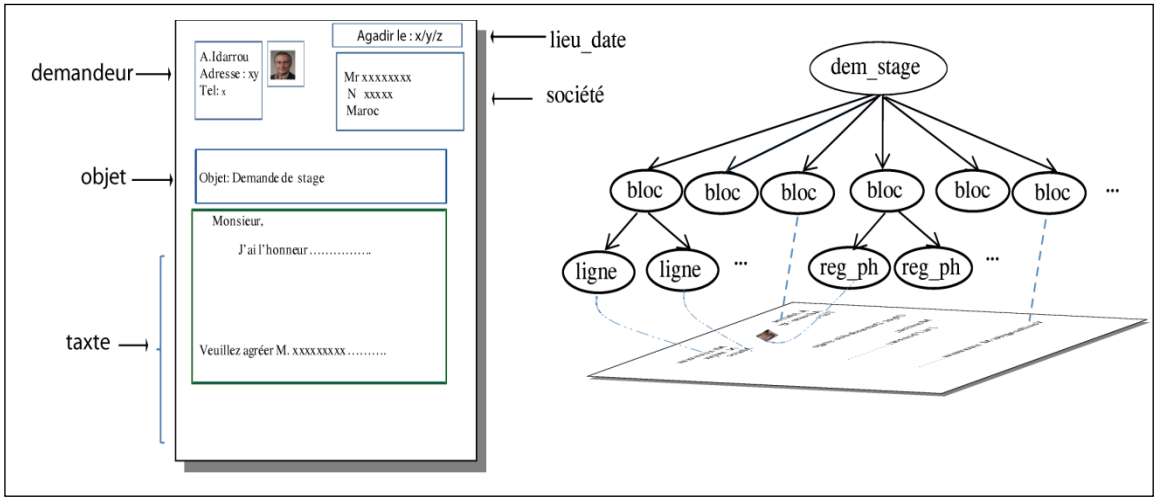
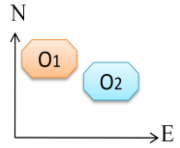
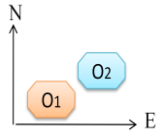
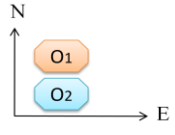


Figure I.5 - Exemple de structure physique

III.2.3. La structure spatiale

La structure spatiale, appelée également composition graphique, concerne l'organisation des composants du document dans l'espace. Elle exprime les contraintes d'ordonnancement des différentes parties de ce document sur le support de présentation. Elle doit définir, par exemple, leur taille, leurs superpositions, leurs juxtapositions, etc (par exemple, l'objet « texte » de la figure I.5 occupe 28% de la page en cours). Les composants d'un document sont reliés par des relations spatiales (tableau I.1). Ces relations permettent d'organiser les objets d'un document et les situer les uns par rapport aux autres. Par exemple, les relations spatiales sont utilisées pour lier deux points ou deux régions d'une image [Papadias D. et al., 1997]. Les relations spatiales sont aussi utilisées dans des requêtes spatiales pour localiser les objets par rapport à un repère. Elles peuvent être classées selon deux catégories : les relations cardinales, appelées encore directionnelles, pour décrire les objets dans un espace planaire 2D (nord, sud, ouest, etc) et les relations topologiques (graphiques) pour décrire les voisinages entre les objets : recouvrement, inclusion, etc. Un ensemble de relations topologiques a été proposé par [Lementini E.C. et al., 1993].

Relations	Représentations graphiques
- Nord-Oest(O_1,O_2) - Sud_Est(O_2,O_1)	
- Sud-Ouest(O_1,O_2) - Nord-Est(O_2,O_1)	
- Nord(O_1,O_2) - Sud(O_2,O_1)	

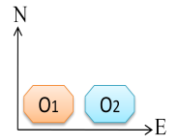
- Ouest(O_1,O_2) - Est(O_2,O_1)	

Tableau I.1- Relations cardinales spatiales

Les relations spatiales les plus connues sont relatives à un espace de dimension 2 (plan). Elles doivent être adaptées pour pouvoir définir les positions inter et intra objets dans un espace de plus de deux dimensions (dans un univers 3D par exemple).

III.2.4. La structure temporelle

Un document multimédia est caractérisé par une dimension temporelle qui permet la description de l'enchaînement des objets de ce document dans le temps. En effet, quelle que soit leur granularité, les objets multimédias sont reliés temporellement entre eux. La structure temporelle d'un document est fondée sur la description des évènements. Pour assurer la cohérence globale du document, il est nécessaire de respecter une organisation suivant un ordre chronologique de l'ensemble de ces évènements. Ceci définit un ordre global de la présentation qu'on pourra modéliser sous forme d'un scénario bien précis (exemple de la figure I.6). Plusieurs travaux ont abordé la structure temporelle des documents. Citons par exemple les travaux de [Allen J. F., 1983], [Allen J. F., 1991], [Vilain M., et al., 1986], [Vazirgiannis et al., 1998], [Beigbeder M., 2004] et [Metzger J.P. et al., 2004]. Les notions d'élément temporel, d'attribut temporel et de relation temporelle ont été définies dans [Bruno E. et al., 2004]. Les relations temporelles, entre objets d'un document, permettent d'assurer une cohérence inter et intra-objet de ce document.

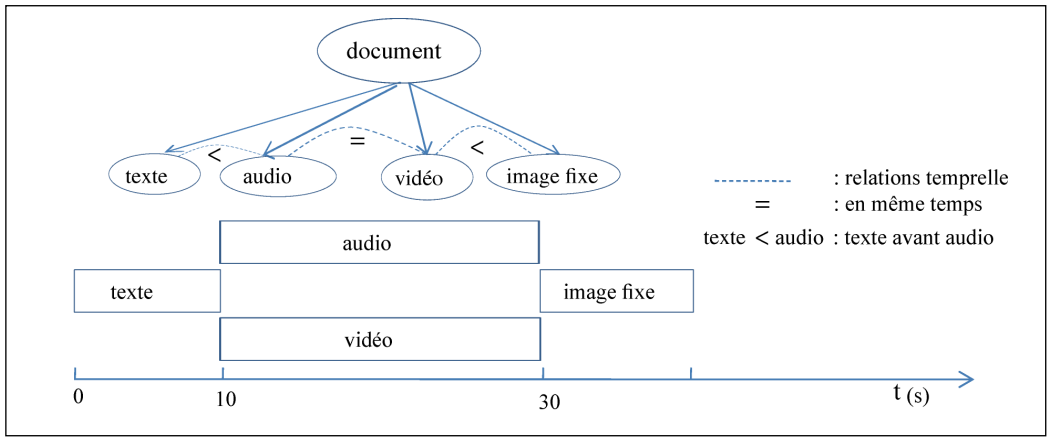


Figure I.6 - Exemple de structure temporelle (synchronisation inter objets)

Dans l'exemple de la figue I.6, nous avons quatre objets multimédias : texte, audio, vidéo et image fixe. Ces objets sont organisés comme suit : l'objet texte est joué de 0 à 10s, les objets audio et vidéo sont joués de 10 à 30s et l'objet image fixe est joué de 30 à 40s. Il faut noter également les relations entre les composants de ce document. Par exemple le texte est joué seul et sa fin doit déclencher en même temps les objets audio et vidéo, etc.

Les relations d'Allen [Allen J.F., 1991] permettent de structurer le contenu d'une séquence audiovisuelle en se basant sur les informations temporelles (tableau I.2).

Relations		Représentations graphiques
X < Y	Précéde(X, Y)	
X m Y	Rencontre(X, Y)	
X o Y	chevauche(X,Y)	
X s Y	Débute(X, Y)	
X d Y	Pendant(X, Y)	
X f Y	Termine(X, Y)	
X = Y	Equivalent(X,Y)	

Tableau I.2 - Relations d'Allen

III.2.5. La structure sémantique

La structure sémantique a été abordée par plusieurs travaux de recherche, nous citons par exemple [Nanard M. et al., 1996], [Poullet L. et al ., 1997], [Chabbat, 1997], [Fourel F., 1998], [Abascal R. et al., 2005] et [Chatti N., 2007]. La structure sémantique est définie au travers de la composition sémantique, représentant le sens d'un ou plusieurs éléments de la structure logique [Poullet L., 1997]. Pour [Fourel F., 1998], la structure sémantique est l'organisation des entités d'informations qui représentent des idées, des connaissances décrites dans le document. Dans la structure sémantique, les données sont organisées selon leur sens et leur définition respective [Abascal R. et al, 2005]. Dans [Chatti N., 2007], la structure sémantique est une désignation fréquemment utilisée pour identifier les structures explicitant des informations relatives au sens véhiculé par différentes parties d'un contenu.

Généralement, cette composition est traduite par des métadonnées décrivant les éléments de la structure logique. La structure sémantique permet donc d'enrichir les éléments de la structure logique en leur donnant plus de sens (figure I.7).

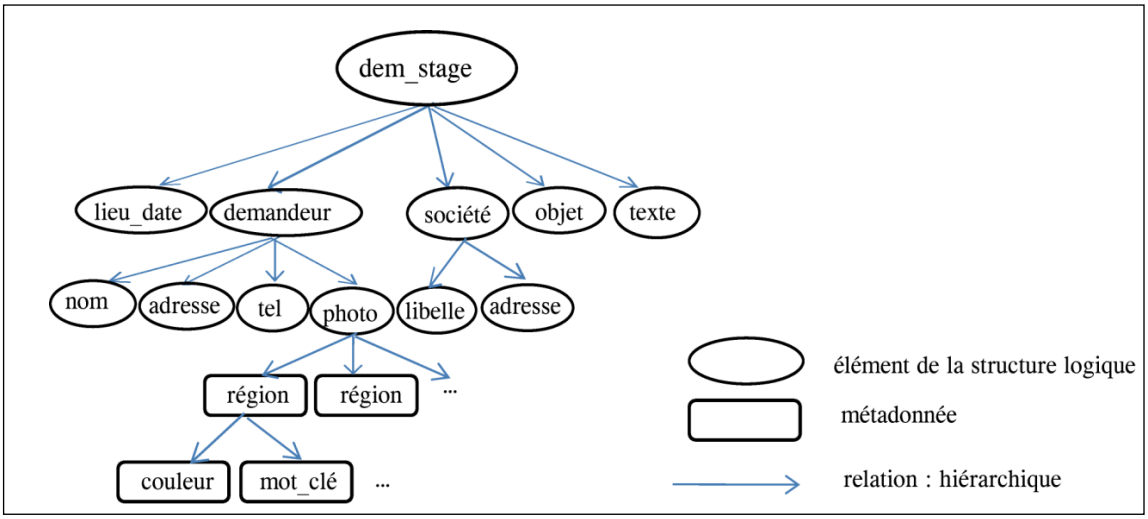


Figure I.7 - Exemple de structure sémantique

III.2.6. La structure hypertexte

La structure hypertexte permet de définir des liens typés non-hiérarchiques entre parties d'un même document ou entre des documents différents. Un lien hypertexte permet au lecteur de quitter la séquence normale d'un texte et d'accéder directement à la partie, souhaitée, du document. Par exemple, un lien pourra être utile pour le renvoi à une note ou à une référence bibliographique ou à un autre document. Pour différencier texte et hypertexte, [Laufer R. et Scavetta D., 1992] définissent un texte comme étant une structure linéaire plus ou moins fortement hiérarchisée : les éléments textuels, plus ou moins autonomes, sont reliés par des relations d'ordre. Un hypertexte est une structure en réseau : les éléments textuels sont des nœuds reliés par des relations non linéaires et faiblement hiérarchisées.

III.2.7. La structure hypermédia

Le préfixe « hyper » est employé pour montrer que la structure est importante [Woodhead N., 1991]. La structure hypermédia décrit les relations entre documents ou entre composantes d'un même document [Jedidi A., 2005]. Ces liens permettent à l'utilisateur une navigation interactive dans un réseau d'informations multimédias sémantiquement riche. La différence entre lien hypertexte et lien hypermédia réside dans les types de données auxquelles l'utilisateur a accès. Un lien hypermédia peut être considéré comme un lien hypertexte multimédia. Plus précisément, un lien hypermédia permet l'accès à des textes, images et sons dans un document ou dans des documents différents. Dans [Landow G.P., 1992], Hypermédia est l'hypertexte auquel on ajoute le multimédia.

Il est à signaler que pour étendre la typologie des structures documentaires et répondre à des besoins spécifique, d'autre structures sont utilisées telles que :

- La structure linguistique qui permet de fournir des informations d'ordre lexical, syntaxique, morphologique, etc entre les composantes du document,
- La structure rhétorique [Mann W.C. et Thompson S.A. 1988], [Luc C, 2001] permet de représenter les relations reliant les segments de texte adjacents entre eux,

- etc.

Dans la section suivante, nous abordons quelques notions sur les standards de structuration documentaire.

IV. Notions générales sur les standards de structuration documentaire

L'une des préoccupations des entreprises et des institutions est la gestion de l'information car l'échange et donc l'interopérabilité de celle-ci posent problématique. Un besoin accru des normes et des standards s'impose afin d'assurer l'interopérabilité et faire circuler l'information de manière sécurisée, transparente, plus efficace, mieux organisée et facile à repérer. Dans [Chatti N., 2007], Les technologies liées à la structuration documentaire constituent le fondement des systèmes d'échange et d'accès à l'information.

Pour répondre à des besoins en matière de structuration, de représentation de l'information et faciliter l'échange des documents tout en évitant les formats propriétaires, des langages standards documentaires ont été proposés et la notion de documents structurés a fait son émergence. Vers la fin des années 80, SGML (Standard Generalized Markup Language) une norme (ISO 1986) a été proposée. Ce méta-langage, adopté par l'organisation internationale de standardisation sous le nom d'ISO 8879, permet la description structurelle des documents. Son avènement suscite une grande révolution dans le monde documentaire. Il permet la structuration de l'information à l'aide d'un système de balises. SGML a connu un grand succès dans le monde industriel où il a été utilisé afin de rédiger des documents techniques assez volumineux. La lourdeur et la complexité de la spécification de SGML et la difficulté de sa mise en œuvre a fait que celle-ci n'est plus utilisée, après l'apparition de son successeur XML.

XML est un méta langage de description et d'échange de documents, indépendamment de la plate forme utilisée, des langages de programmation et des formats d'affichage. Recommandé par le Consortium W3C, XML permet de structurer les documents à l'aide d'un système de balisage. Ces balises ne sont pas spécifiées par le langage, ce qui permet à l'auteur du document de définir ses propres balises. Il permet de séparer contenu et structure des documents numériques. XML a profité des avantages de son ancêtre SGML en bénéficiant de plusieurs concepts visant à obtenir une formalisation simplifiée, avec une souplesse syntaxique, une possibilité de définir des structures de lien, une possibilité de modularité grâce à la spécificité d'espaces de noms « Namespaces 1999 », etc. Depuis son apparition, XML n'a cessé d'évoluer et de s'enrichir pour répondre à de nouveaux besoins, notamment aux besoins spécifiques du web sémantique. Dans [Roisin C., 1999], XML peut être considéré comme le standard pivot du web à partir duquel et autour duquel les autres standards de représentation des informations du web sont définis. Dans ce contexte, d'autres standards basés sur XML ont été principalement définis :

- WML (Wireless Markup Language) est conçu spécifiquement pour les applications WAP (Wireless Application Protocol), il permet de faciliter l'affichage sur des terminaux d'appareils mobiles
- XSL (eXtended Stylesheet Language) est un langage extensible de feuilles de style qui permet de décrire la mise en page et le format d'affichage d'un document XML. En effet, XML permet de séparer le contenu d'un document de sa structure et de son

format d'affichage mais il ne permet pas de décrire le format d'affichage ou d'impression de ce document. Pour palier à ce problème, un standard de mise en page est nécessaire. Ainsi, une description XSL, utilisant la syntaxe XML, peut être appliquée à un document pour définir l'affichage des divers éléments de ce document.

- Le Voice XML (Voice eXtensible Markup Language ou langage de balisage extensible vocal) est un langage normalisé permettant la création des applications vocales avec des interactivités sonores, de la reconnaissance de parole, etc,
- Le VRML (Virtual Reality Modeling Language) est une norme ISO. C'est un langage qui permet de concevoir des simulations interactives dans un environnement multi-utilisateurs et en 3D. C'est un langage de présentation qui permet la représentation d'univers interactifs 3D virtuels,
- Le langage SVG (Scalable Vector Graphics) est un standard, proposé par W3C et est conçu pour décrire des objets graphiques en 2D. Le Langage SVG supporte les objets : les formes graphiques vectorielles, les images, le texte, etc,
- SMIL (Synchronized Multimedia Integration Language) est un langage d'intégration multimédia synchronisé, il est recommandé par le Consortium W3C. Il permet la description des présentations multimédia.
- etc.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE dem_stage SYSTEM "dem_stage.dtd">
<dem_stage>
    <lieu_date>     ……..……. </lieu_date>
    <demandeur>
      <nom> ……. </nom>
      <adresse> …. .</adresse>
      <tel> ……. </tel>
      <photo>
        <région> …  </région>
        <région> …  </région>
        ……..
      </photo>
    </demandeur>
    <société>
      <libelle> …. ……. </libelle>
      <adresse>            </adresse>
      </société>
    <objet> ……. </objet>
    <texte> ….. ……</texte>
    </dem_stage>
```

Figure I.8 - Exemple de document XML (dem_stage.xml)

Un document en format XML (exemple, Figure I.8) doit vérifier deux niveaux de conformité : (1) il doit être bien formé (respecte les règles syntaxiques de XML), (2) : il doit être valide (conforme aux règles d'un modèle ou schéma). Pour cela, une multitude de parseurs ont été proposés. Généralement, un parseur est un outil logiciel permettant de parcourir un document et d'en extraire des informations.

Les parseurs se distinguent par l'approche utilisée pour traiter un document. Ces approches peuvent être groupées en deux catégories : (1) Une catégorie basée sur les API (*Application Programming Interface*) utilisant l'approche hiérarchique dont le principe est de construire une structure hiérarchique composée d'objets représentant les éléments du document. DOM (*Document Object Model*) est l'approche la plus connue dans cette catégorie, elle permet de construire une représentation globale du document en mémoire sous forme d'arbre. (2) Une catégorie basée sur les API dont le principe est basé sur l'aspect événementiel. L'approche SAX (*Simple API for XML*) est l'interface la plus connue de cette catégorie. Elle permet de parcourir le document en déclenchant des événements (par exemple début et fin de l'élément, du document, etc) et de produire des résultats pouvant être exploité par des applications utilisant cette API.

Les parseurs utilisant l'approche DOM souffrent du fait que cette API impose la construction d'un arbre contenant l'intégralité des éléments du document en mémoire. C'est une contrainte quand il s'agit de traiter de gros documents. Ainsi, il est plus pratique d'utiliser des API événementielles (comme SAX), permettant de traiter uniquement ce qui est nécessaire.

Malgré ses les limites, il a connu un succès important surtout dans le monde industriel où il a été utilisé pour décrire une lourde documentation technique. Après avoir donné naissance à son successeur XML, la spécification SGML n'est presque plus utilisée. XML est plus souple que son ancêtre SGML, il a été conçu pour la description des documents, leurs stockages dans des environnements répartis et leurs échanges en assurant leur interopérabilité.

XML a bénéficié des avantages de SGML, à savoir par exemple la possibilité de définir des modèles de documents respectant une DTD (Document Type Definition). Les atouts de XML sont nombreux en voici quelques uns : flexibilité, lisibilité, extensibilité, portabilité, structure hiérarchique, etc. Depuis 1998, XML est devenu le langage standard pour la description de documents structurés. XML a introduit des balises, dites sémantiques, car elles structurent le sens des documents. La figure I.9 représente la DTD du document « dem_stage.xml » de la figure I.8.

```
< !DOCTYPE dem_stage [
<!ELEMENT dem_stage(lieu_date,demandeur,société,objet,texte)>
<!ELEMENT lieu_date (#PCDATA)>
<!ELEMENT demandeur(nom,adresse,tel,photo)>
<!ELEMENT nom(#PCDATA)>
<!ELEMENT adresse(#PCDATA)>
<!ELEMENT tel(#PCDATA)>
<!ELEMENT photo (région)+>
<!ELEMENT région(couleur,mot_cle)>
<!ELEMENT couleur(#PCDATA)>
<!ELEMENT mot_cle(#PCDATA)>
<!ELEMENT société(libelle,adresse)
<!ELEMENT libelle(#PCDATA)>
<!ELEMENT adresse(#PCDATA)>
<!ELEMENT objet(#PCDATA)>
<!ELEMENT texte(#PCDATA)>
] >
```

Figure I.9 - La DTD du document dem_stage.xml (figure I.8)

Depuis l'apparition de ces standards documentaires, les technologies liées à la structuration documentaire continuent à se diversifier et proliférer pour s'adapter à des contraintes de diffusion, d'échange et d'exploitation documentaire.

Dans la section suivante, nous abordons quelques notions sur la multi-structuralté documentaire.

Après avoir abordé les concepts de base des documents multimédia, nous allons aborder, dans la section suivante, les représentations structurelles documentaires les plus utilisées.

V. Représentations structurelles documentaires

La représentation d'une structure documentaire est une tâche primordiale. En effet, face à des besoins accrus de l'évolution technologique et de l'utilisation massive des documents multi-structurés, il convient de conserver le maximum d'informations pertinentes sur les documents. Ces informations doivent être organisées afin de pouvoir les traiter efficacement, de façon automatique ou manuelle. Cette organisation doit prendre en compte toutes les composantes des documents et les relations entre ces composantes. Plus précisément, les résultats des systèmes traitant les documents multimédia dépendent fortement de la richesse de la représentation de ces documents.

En général, une structure de données est une organisation des informations permettant de faciliter leur traitement le plus efficacement possible.

Les vecteurs numériques, les listes, les arbres, les forêts et les graphes sont des structures de données couramment utilisés pour représenter les structures des documents.

V.1. Représentation sous forme de vecteur

Les vecteurs sont des structures de données qui ont été utilisés dans plusieurs travaux pour représenter les documents. Pour cela, un ensemble de composants jugés pertinents par

rapport au domaine d'application (en recherche d'information, en classification documentaire, etc) est conservé. Les composants d'un vecteur sont les éléments structurels du document. L'avantage de cette méthode de modélisation est la souplesse de manipulation. Par exemple, les distances entre vecteurs représentant des documents sont facilement calculables. Dans [Salton G., 1971], le modèle vectoriel repose sur la théorie mathématique des espaces vectoriels. Selon [Nassr N., 1999], dans le modèle vectoriel, les requêtes et les documents sont vus comme des vecteurs dans un espace euclidien engendré par tous les termes d'indexation.

Exemple

Soient les documents d_1 = {document multimédia} et d_2 = {document multimédia numérique}. Le lexique de ces documents (la base de l'espace vectoriel) est L = {document, multimédia, numérique}.

Les vecteurs v_1 *et* v_2 représentant respectivement d_1 et d_2 sont $v_1(1,1,0)$ et $v_2(1,1,1)$.

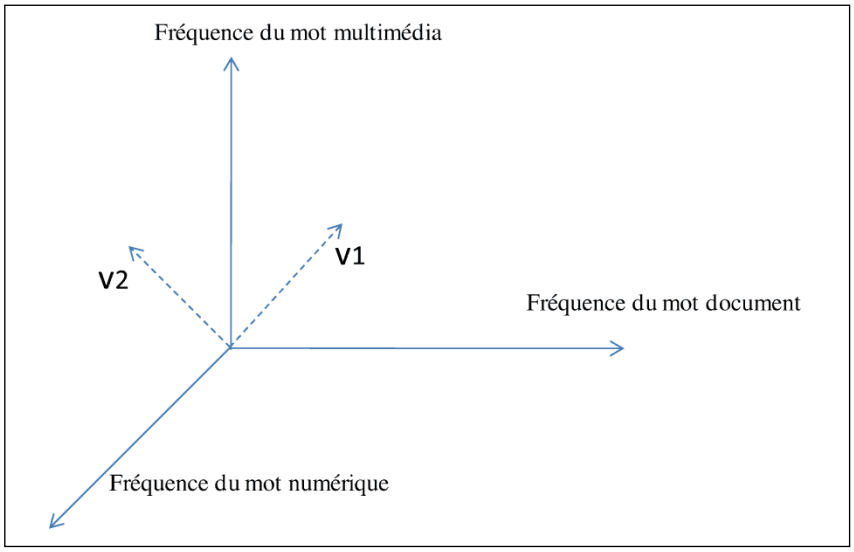


Figure I.10 - Exemple de représentation d'un document en vecteur

Dans cet exemple, un document est une suite de mots, il est représenté par un vecteur élément de R^3 (figure I.10).

Les vecteurs ont été utilisés par [Doucet et al., 2002] et [Yi J. et Sundaresan N., 2000] pour représenter les documents textuels afin d'évaluer leur similarité. [Del Razo et al., 2006] utilisent l'approche proposée par [Weiss M.A., 1998] pour représenter les schémas XML à l'aide des vecteurs.

Si les vecteurs sont des structures de données faciles à manipuler, en revanche, ils ne permettent pas de modéliser les objets complexes et structurés. Par exemple, ils ne permettent pas une représentation explicite des relations entre les composants de ces objets. Dans [Sorlin S., 2006], les vecteurs ont un pouvoir d'expression assez pauvre. Ils ne permettent pas d'exprimer les relations multiples qui peuvent exister entre les composantes d'un document.

V.2. Représentation sous forme de liste

Une liste est une structure de données qui consiste en un ensemble d'éléments ordonnés. Elle permet d'établir un ordre total entre les objets qu'elle représente (exemple, figure I.11). Les listes sont utilisées, par exemple, pour représenter les documents en recherche d'informations lorsqu'il s'agit d'ordonner des éléments. Les listes chaînées sont utilisées par [Ros J. et al. 2005] pour comparer des images.

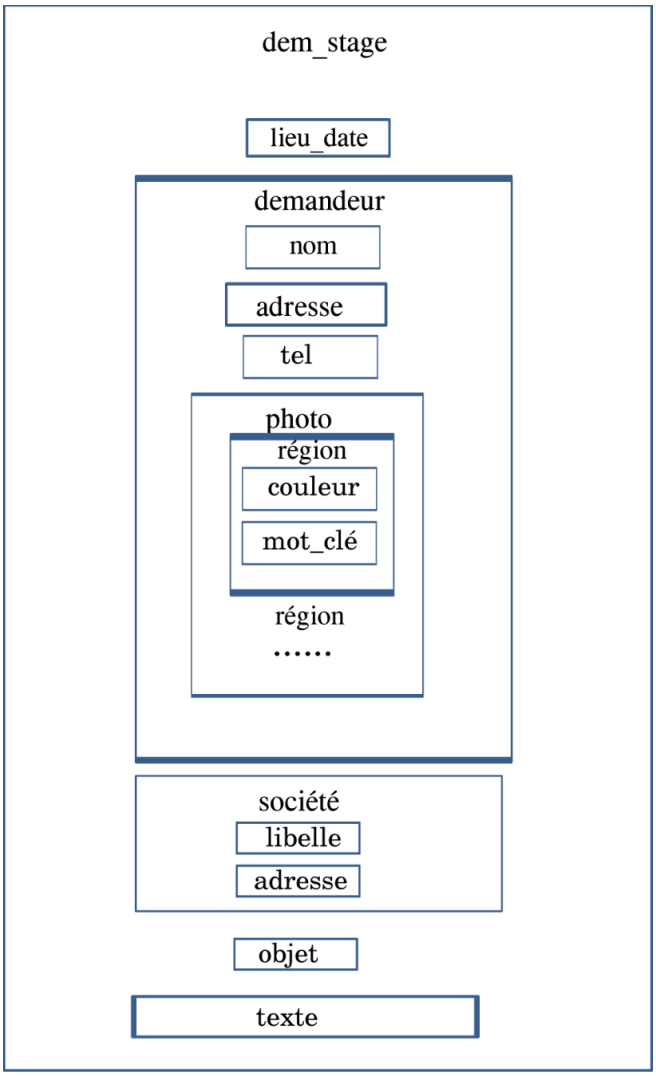


Figure I.11 - Exemple de structure de liste

V.3. Représentation sous forme d'arbre

Les arbres permettent de représenter des objets structurés et sont fréquemment utilisés pour représenter des documents. La structure d'arbre permet de représenter les relations hiérarchiques qui s'ajoutent à la relation d'ordre. Ces relations traduisent les inclusions entre les composants du document. Un nœud de l'arbre peut être considéré comme un sous-document (sous-arbre). La granularité de la représentation dépend de la profondeur de

l'arbre. La structure sémantique de l'exemple de la figure I.7 est représentée sous forme d'arbre.

En classification des documents XML, les arbres ont été utilisés par [Termier A. et al., 2002], [Costa G. et al., 2004], [Vercoustre A. M. et al., 2006], [Mbarki M., 2008], [Aït elhadj A., et al., 2009] et [Tagarelli A., 2010]. En recherche d'information (RI), [Ben Aouicha M., 2009] a utilisé les arbres pour représenter les documents et les requêtes.

V.4. Représentation sous forme de forêt

Les structures de listes et d'arbres induisent un ordre total entre les éléments du document. Cependant, il existe des éléments dont la position n'est pas déterminée par rapport à la structure logique. C'est le cas des figures ou notes de bas de pages, par exemple. Il est donc plus approprié de considérer la structure d'un document comme étant une forêt. Une structure en forêt peut être vue comme un ensemble d'arbres (exemple figure I.12).

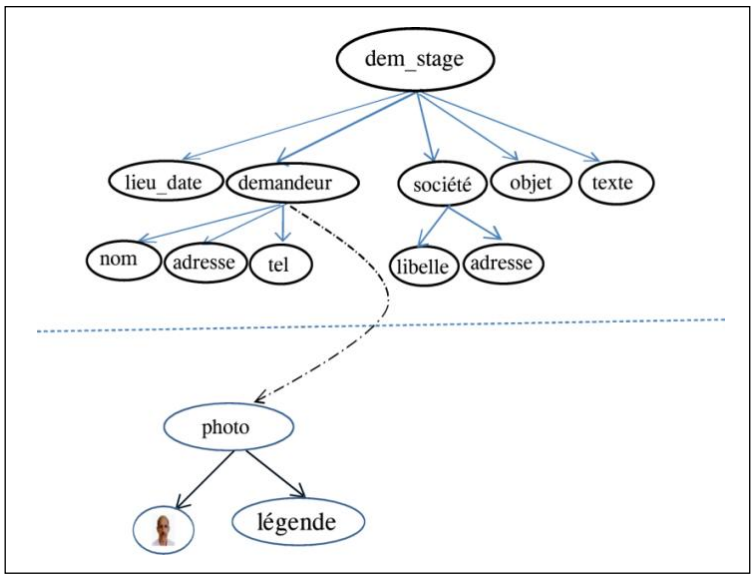


Figure I.12 - Exemple de structure en forêt

V.5. Représentation sous forme de graphe

Les approches utilisant les arbres pour représenter les documents multimédias sont confrontées au problème de la limite de représentation des relations multiples entre deux mêmes nœuds d'un document (exemple figure I.13).

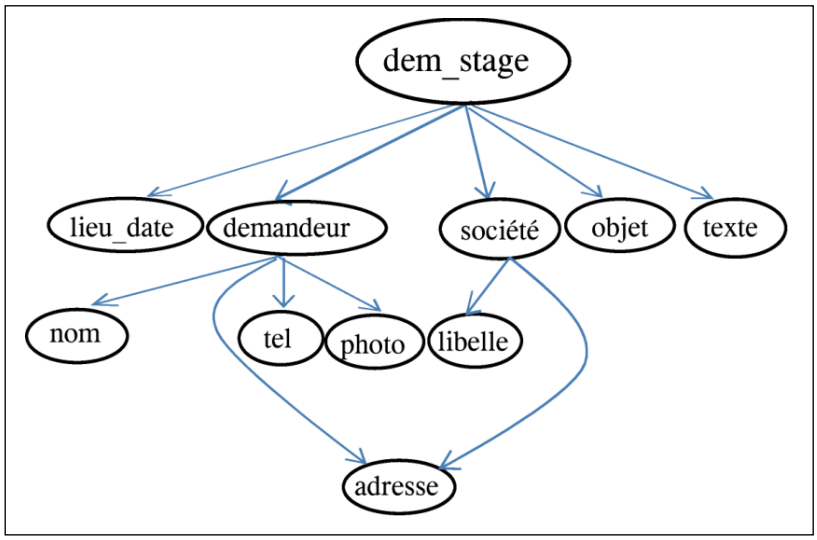


Figure I.13 - Exemple de structure de graphe orienté

Les graphes sont souvent utilisés pour représenter des objets complexes et structurés : les sommets représentent les composants de l'objet et les arcs représentent les relations binaires entre ces composants. Les graphes orientés sont une généralisation des arbres. Ils permettent une modélisation riche des documents et de leurs structures. Les composants et relations peuvent être étiquetés. Leur très forte expressivité fait que les graphes sont utilisés dans de nombreuses applications.

V.6. Conclusion

Dans la section précédente, nous avons donné quelques définitions des structures de données les plus connues. Le choix d'une structure dépend du contexte et du problème à résoudre. Par exemple, les listes permettent de modéliser les documents textuels simples, mais la représentation de ces mêmes documents, à l'aide des arbres, donnera plus de richesse à cette modélisation. En générale, les listes ne permettent pas de représenter toutes les relations existantes entre les composantes d'un objet. Les vecteurs sont faciles à manipuler mais ils ont un pouvoir d'expression assez pauvre. Ils ne permettent pas de représenter les objets complexes.

Les arbres permettent de modéliser des objets ayant une hiérarchie simple. Quand les objets à modéliser ont une hiérarchie complexe on pourra avoir recours à la structure en forêt qui donnera plus de possibilités. En effet, une forêt est une structure constituée d'un ou de plusieurs arbres. Cependant, les approches utilisant les arbres pour représenter les objets complexes et structurés sont confrontées au problème de la limite de représentation des relations multiples entre deux composants d'un même objet.

Les graphes orientés sont une généralisation des arbres. Ils permettent une modélisation riche des documents et de leurs structures : plus d'opérations et beaucoup plus d'expressivités en matière de modélisation avec une théorie sous-jacente pouvant être un appui pour le traitement des structures.

VI. Notions de bases sur la multi-structuralité

VI.1. Introduction

Un document multimédia est caractérisé par une complexité structurelle, issue de plusieurs sous-documents eux mêmes plus au moins complexes. Selon [Djemal K., 2010], un document est multi-structuré, soit parce que plusieurs structures le décrivent, soit parce que les documents qui le composent sont eux-mêmes structurés ou multi-structurés. La notion de document devient complexe et difficile à appréhender. Plusieurs caractéristiques peuvent être considérées pour définir et décrire un document en tenant compte de plusieurs paramètres à la fois de l'utilisateur, du dispositif de restitution, des contraintes du matériel et des plates-formes, des réseaux comme moyens d'échange, etc. Par exemple, la représentation d'un document peut ne pas se présenter de la même façon sur un support de papier, sur un écran PDA ou sur un téléphone portable (exemple Figure I.3). Cette diversité implique un besoin d'adaptation des documents à leurs contextes. Selon [Bruno E. et al., 2007], certaines applications nécessitent la description d'un texte selon différents niveaux d'analyse qui correspondent à des usages différents de ce texte. Ainsi un document peut posséder plusieurs structures hiérarchiques. La définition simultanée de plusieurs structures pour un même document suscite la problématique dite des documents multi-structurés [Bruno E. et al., 2007]. Le problème de la multi-structuralité n'est pas nouveau, déjà depuis SGML, fin des années 80, on parle de la structure *logique* et de la structure *physique* d'un même document. Une première structure peut être définie pour organiser logiquement le contenu d'un document tandis qu'une deuxième explicitera les règles de sa mise en forme sur un support physique [Chatti N., 2007]. D'autres types de structures, liées plutôt à la nature du document, ont également été définis. C'est le cas par exemple des structures spatiales, temporelles ou encore spatio-temporelles, associées aux documents multimédias [Chrisment C., et al., 2002].

Dans ce qui suit, nous citons quelques définitions de la multi-structuralité.

VI.2. Définitions

Pour [Durusau P. et al., 2002] et [Tennison et al., 2002], le concept de multi-structuralité est apparu du fait qu'il est souvent très difficile de réduire la structure d'un document à un arbre unique. Ils supposent que les documents textuels ont souvent plusieurs structures. Pour [Abascal R. et al., 2003], la multi-structuralité est une description d'un document par un ensemble d'éléments en relation les uns avec les autres, au cours ou en vue d'un usage. Selon [Chatti et al., 2007], la multi-structuralité peut être considérée comme étant la définition simultanée de plusieurs structures pour un même document de base. Ainsi, un document multi-structuré est décrit par un ensemble de structures mises en correspondance. L'une de ces structures est constitutive du document et toute autre structure doit être attachée à cette structure pivot par le biais d'une correspondance. La diversité de ces structures dépend de plusieurs paramètres : contexte, usage du document, etc. Les auteurs de [Abascal et al. 2004] proposent une définition plus large de la multi-structuralité. Ils définissent le document multi-structuré comme étant une entité unique dans laquelle sont englobées des structures différentes.

Après avoir faire un tour d'horizon sur les concepts des documents, leurs composants, leurs structures, dans la section suivante nous allons aborder le problème de la représentation des documents à structures multiples.

VII. Représentation des documents à structures multiples

VII.1. Introduction

En général, les structures sont exploitées d'une façon individuelle et indépendamment les unes des autres. Cependant des avantages peuvent être tirés de l'exploitation conjointe de plusieurs structures d'un même document. Le fait de combiner plusieurs structures d'un même document pourra donner une exploitation plus riche et plus efficace de celui-ci. Par exemple en recherche d'information, on pourra demander de restituer les cinq premières lignes de la deuxième page de la première section de chaque chapitre d'un document donné. Pour répondre à ce besoin, les structures logique(s) et physique(s) de ce document, doivent être utilisées simultanément.

Les langages standards de structuration comme XML et ses dérivés permettent de sérialiser dans un même fichier une structure arborescente d'un document, mais leur utilisation pour représenter les structures concurrentes (d'un même contenu) pose un certain nombre de problèmes. Plusieurs propositions, visant à définir ou à adapter des formalismes pour représenter ces documents, ont été faites.

Nous présentons dans ce qui suit un panorama non exhaustif mais représentatif des solutions concernant la problématique de la représentation de la mult-structuralité.

VII.2. Solutions sur la représentation des documents multi-structurées

Pour exploiter efficacement les documents à structures multiples, il est nécessaire d'avoir un formalisme d'encodage approprié. Les langages standards de structuration ne s'adaptent pas facilement aux spécificités de ce type de documents. Par exemple, avec le langage standard XML, l'encodage de plusieurs structures (dans un même ficher en format XML) peut engendrer le problème de chevauchement des éléments (un élément d'une structure ne s'imbrique pas dans un élément d'une autre structure). Dans ce cas, le document résultant ne sera pas bien formé.

Nous présentons ci-dessous, deux exemples de solutions : un exemple de solutions standard permettant l'encodage des documents multi-structurés et un exemple de solutions propriétaires permettant la représentation des documents à structures multiples.

VII.2.1. Solutions standards

L'option CONCUR du standard SGML est une solution permettant de décrire plusieurs structures d'un même document, dans un même fichier. Cette technique permet de définir autant de DTD que de structures [Goldfarb C.F. et al., 1990]. L'utilisation d'un préfixe identifiant la DTD, dans laquelle est défini un élément, permet de distinguer entre les éléments de chaque structure. Ainsi, les documents représentés selon cette méthode sont valides par rapport à chacune des DTD définies. Pour cela la TEI (Text Encoding Initiative)

a proposé des solutions pour encoder un document multi-structuré dans un même document en format XML. Si ces solutions sont simples à comprendre par l'opérateur humain, elles ne sont pas faciles à gérer d'une façon automatique. Par conséquent, ces techniques ne peuvent pas être considérées comme de véritables solutions pour l'encodage des structures multiples.

VII.2.3. Modèles propriétaires

Dans ce contexte, d'autres modèles propriétaires ont été proposés pour résoudre la problématique de la représentation des documents multi-structurés :

- Le modèle *MSXD* (*MultiStructured XML Documents*) proposé dans [Bruno E. et Murisasco, E., 2006]. Ce modèle propose de décrire séparément chaque structure dans un document XML. Les relations entre les éléments des structures sont modélisées par un ensemble de contraintes définissant un schéma du document multi-structuré,
- Le modèle *MSDM* (*Mulsi-Structured Document Model*) a été proposé dans [Chatti N. et al., 2006] (figure I.14), ce modèle est basé sur une première proposition définie dans [Abascal R. et al. 2003]. Le formalisme MultiX a été défini dans [Chatti N. et al. 2006], ce formalisme est basé sur le modèle *MSDM*. Il permet l'encodage de structures multiples de documents. MultiX est une application XML qui permet d'expliciter les caractéristiques particulières des documents multi-structurés et d'optimiser ainsi leurs exploitations. En effet un document MultiX est décomposé en trois parties : les structures documentaires (*SD*), la structure de base (*SB*) et les correspondances,

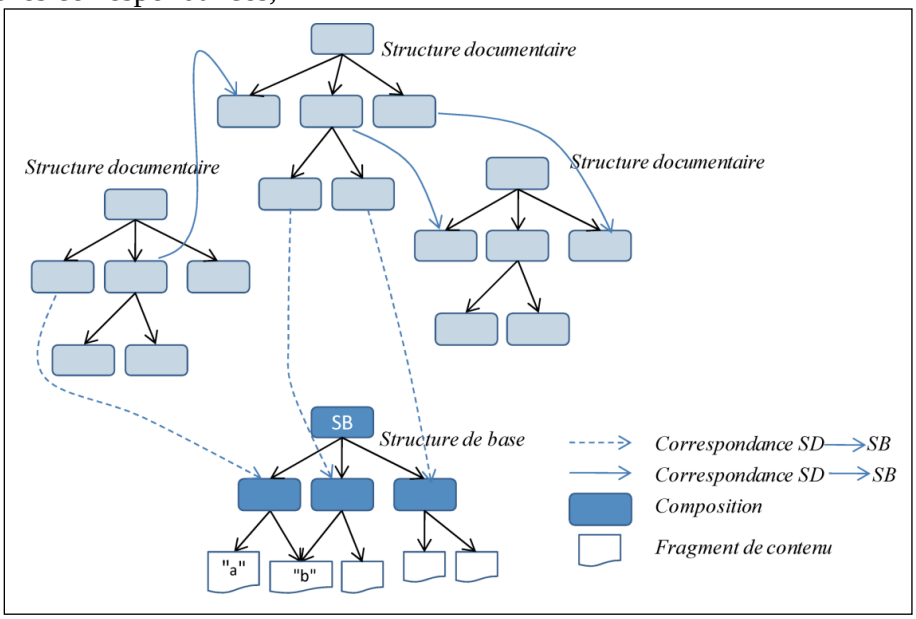


Figure I.14 - Illustration du modèle MSDM [Chatti N., 2006]

- Dans [Mbarki M., 2008], l'auteur a proposé un modèle qui explore les structures logique et sémantique d'un même document tout en assurant une dichotomie entre ces deux structures. Dans ce modèle, un document possède d'une part une structure logique composée des éléments et leurs attributs et d'autre part une ou plusieurs

structures sémantiques composées de composants et de méta-données associées à un élément de la structure logique. Ainsi la structure logique joue un rôle pivot dans ce modèle et la gestion de la multi-structuralité, dans ce travail, se limite uniquement à la structure logique et à la ou les structures sémantiques associées.

- Le modèle *MVDM* (Multi Views Document Model), a été proposé par [Djemal K., 2010]. Il est organisé autour du concept de *vue*. Une vue correspond à une organisation particulière d'un document. Ce méta-modèle est composé de deux niveaux de description : un niveau spécifique et un niveau générique. Le niveau spécifique décrit les documents au travers de leurs différentes structures traduites par des vues spécifiques. Le niveau générique décrit les structures génériques représentant des classes de structures spécifiques similaires (figure I.15).

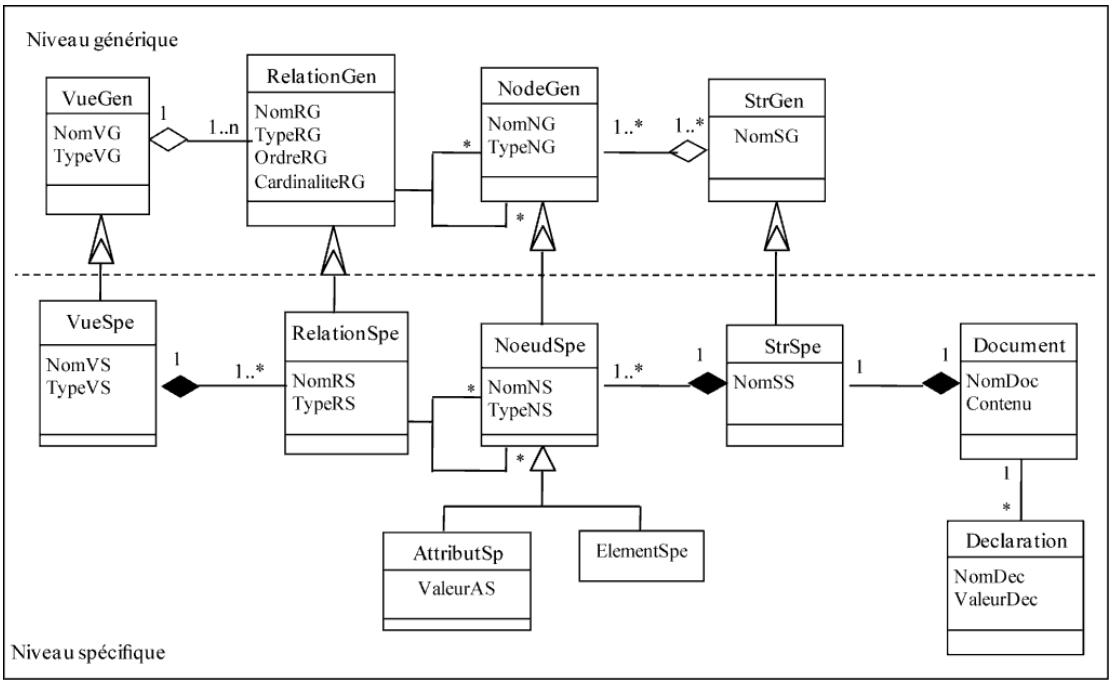


Figure I.15 - Illustration du modèle *MVDM* [Djemal K., 2010]

VII.2.4. Conclusion

Les deux solutions d'encodage basées sur XML que nous avons abordées dans la section VII.2.2, proposent différentes techniques qui permettent de contourner la problématique de la structuration multiple des documents. Malgré l'apport théorique de ces techniques, pratiquement elles sont complexes et difficiles à utiliser.

Parmi les inconvénients du modèle *MSXD*, nous notons le problème de la redondance du contenu et les difficultés de gestion d'évolution qui en découlent (par exemple la mise à jour d'une structure). Par conséquent, pour les structures de taille importante et qui évoluent rapidement, cette technique ne convient pas. Le modèle *MSDM* privilégie une structure (structure de base), la mise à jour de celle-ci nécessite la mise à jour de toutes les relations (entre celle-ci et les autres structures). Dans le modèle proposé par [Mbarki M. 2008], la gestion de la multi-structuralité se limite à la structure logique et la ou les structures sémantiques associées. Ainsi, ce modèle ne traite que des relations de

compositions entre la structure logique et la structure sémantique. *MVDM* permet de représenter les différentes structures d'un document multi-structuré. Il permet une gestion flexible et un stockage efficace de ce type de documents en tenant compte de la gestion du chevauchement entre les structures d'un même contenu.

Une étude sur l'état de l'art des solutions proposées pour représenter les documents multi-structurées, a été faite par [Portier P.E., 2010] qui a monté que le modèle *MVDM* permet une représentation riche des documents à structures multiples et que cette richesse peut être exploitée pour classer les documents multi-structurés.

VIII. Bibliographie

[Abascal R., 2003] Abascal R., Beigbeder M., Benel A., Calabretto S., Chabbat B., Champin P. A., Chatti, N., Jouve, D., Prie, Y., et Rumpler, B. (2003). « Modéliser la structuration multiple des documents » H2PTM, Hermès, Paris, France, 253-258.

[Abascal R. et al., 2004] Abascal R., Beigbeder M., Bénel A., Calabretto S., Chabbat B., Champin P. A., Chatti N., Jouve, D., Prié, Y., et Rumpler, B. (2004). « Documents à structures multiples ». *SETIT 2004*.

[Abascal R. et al., 2005] Abascal R., Rumpler B., Berisha-Bohé S. « Proposition d'une nouvelle structure de document pour améliorer la recherche d'information », *Proceedings of the CORIA'05*, ISBN: 2-9523810-0-3, IMAG, pp. 389-404, 2005.

[Abascal R. et al., 2007] Abascal-Mena R., B. Rumpler, « Accès au contenu des thèses numériques par leur structure sémantique » Document Numérique 10(2/200):9-35, Lavoisier - Hermes, ISBN 978-2-7462-2023, 2007.

[Afnor, 1987] AFNOR, Références bibliographiques : contenu, forme et structure. ISO 690 1987. Paris, 1987.

[Aïtelhadj A. et al., 2009] Ali Aïtelhadj, « Classification de Structures Arborescentes : Cas de Documents XML », *CORIA, 6th French Information Retrieval Conference, Presqu'île* de *Giens*, France, May 5-7, 2009. Proceeding. LSIS-USTV, ISBN 2-9524747-1-0 page 301-317, 2009.

[Allen 1991] Allen J. F. (1991). "Time and time again: The many ways to represent time". International Journal of Intelligent Systems, 6(4).

[Allen 1983] Allen J. F. (1983). "Maintaining knowledge about temporal intervals. Communication ACM", 26(11), 837-843.

[Bachimont B, 1999] Bachimont B., "Bibliothèques numériques audiovisuelles : des enjeux scientifiques et techniques". *Document Numérique, Numéro Spécial "Les Bibliothèques Numériques"*, 2(3-4):219-242, Janvier 1999.

[Bachimont B, 1998] Bachimont B. « Bibliothèques numériques audiovisuelles : Des enjeux scientifiques et techniques. » Revue Document numérique, 2(3), 219-242, 1998.

[Bachimont B, 2004] Bachimont B., et Crozat, S. « Instrumentation numérique des documents : pour une séparation fonds/forme. » Information-Interaction-Intelligence I3, 4(1), 95-104, 2004.

[Beigbeder M., 2004] Beigbeder M. (2004). « Les temps du document et la recherche d'information. » *Document numérique*, (2004/4), 55–64.

[Ben Aouicha M., 2009] Ben Aouicha Mohamed, « Une approche algébrique pour la recherche d'information structurée », Thèse de doctorat de l'Université de Paul Sabatier Toulouse 2009.

[Bres S., 1999] Bres S., Champin P.A., Heraud J.M., Herilier V., Jolion J.M., Loupias E., *TeleSUN A world wide multimedia TELEteaching System for UNiversities*, 1999.

[Bringay S., 2004] Bringay, S., Barry, C., et Charlet, J. (2004), « Les documents et les annotations du dossier patient hospitalier » *Revue I3 : Information-Interaction-Intelligence*, 4(1), 191-211.

[Bruno E. et al., 2007] Bruno E., Calabretto S., Murisasco E., « Documents textuels multistructurés : un état de l'art ». *Revue Information - Interaction - Intelligence (I3)* 7 (1) 2007.

[Bruno E. et Murisasco, 2006] Bruno, E., et Murisasco, E. (2006), "MSXD: A Model and a Schema for Concurrent Structures Defined over the Same Textual Data". *Database and Expert Systems Applications*, 172-181.

[Bruno E. et al., 2004] Bruno E., J. LeMaitre et E.Murisasco, « Temporalisation d'un document XML », In *Document Numérique*, volume 8(4), pages 125–141, 2004.

[Bryan M., 1998] Bryan, 1998 Bryan M. "Guidelines for using XML for Electronic Data Interchange", 1998 http://www.xmledi-group.org/xmledigroup/guide.htm.

[Chabbat B., 1997] Chabbat B. *Modélisation Multiparadigme de textes réglementaires*. Thèse de doctorat, LISI. Lyon, décembre 1997, 392 p.

[Chatti N. et al., 2007] Chatti, N., Calabretto, S., Pinon, J. M., et Kaouk, S. « MultiX: an XML-based formalism to encode multi-structured documents », *Proceedings of Extreme Markup Languages 2007*, 2007.

[Chatti N. et al., 2006] Chatti N., Kaouk S., Calabretto S., and Jean-Marie Pinon. "MultiX : an XML-based formalism to encode multi-structured documents". In Proceedings of Extreme Markup Languages'2006, Montréal (Canada), August 2006.

[Chaudiron et al., 2000] S. Chaudiron, F. Role, M. Ihadjadene, 2000. *CodeX : un système pour la définition de vues multiples guidée par les usages*, CIDE 2000, Lyon, FR, 2000, pp. 71-81.

[Chrisment C. et al., 2002] Chrisment C., F. Sedes. *Media annotations: toward a unified representation.* Chapitre du livre Multimedia mining, octobre 2002. Kluwer Academic Publisher.

[Bruno E. et Murisasco, 2006] Bruno, E., et Murisasco, E. (2006). ―MSXD: A Model and a Schema for Concurrent Structures Defined over the Same Textual Data. Database and Expert Systems Applications, 172-181.

[Bui Thi M-P., 2003] Minh Phung BUI THI, « La structuration sémantique des contenus des documents audiovisuels selon les points de vue de la production », Thèse de doctorat de l'Université de Paris VIII, 2003.

[Caro S., 2003] CARO Stéphane, Manuscrit auteur, publié dans « Techniques de l'ingénieur Documents numériques Gestion de contenu », 2003.

[Costa G. et al., 2004] Costa, G., Manco, G., Ortale, R., et Tagarelli, A. (2004). "A Tree-Based Approach to Clustering XML Documents by Structure". *LECTURE NOTES IN COMPUTER SCIENCE*, 137-148.

[Del Razo L.F. et al., 2006] Del Razo Lopez F., Laurent A., Poncelet P., Teisseire M., « Recherche de sous-structures fréquentes pour l'intégration de schémas XML », In Conférence Extraction et Gestion des Connaissances (EGC 2006), Lille, Janvier 2006, volume II, p 487-498.

[Doucet A. et al., 2002] Doucet A. et Ahonen-Myka H., "Naïve Clustering of a large XML Document Collection", In INEX Workshop, pp 81-87, 2002.

[Durusau P. et al., 2002] Durusau, P., et O'Donnell, M. B. (2002). "Concurrent markup for XML documents". *Proc. XML Europe*.
[Egyed Z.E., 2003] Egyed Zsigmond E., « *Gestion des connaissances dans une base de documents multimédias* », Thèse de doctorat en informatique, INSA de Lyon, octobre 2003.

[Ercegovac Z., 1999] Ercegovac Z., "Introduction to the Special Topic Issue, Integrating Multiple Overlapping Metadata Standards.". dans Journal of the American Society for Information Science, Vol. 50, n°. 13, p. 1165-1170, novembre 1999.

[Fourel, F., 1998] Fourel, F. « Modélisation, indexation et recherche de documents structurés » Thèse de doctorat, Université Joseph Fourier, Grenoble, 1998.

[Goldfarb C.F. et al., 1990] Goldfarb C.F., Rubinsky Y., "*The SGML handbook"*. Clarendon Press, Oxford, 1990.

[Jedidi A., 2005] Jedidi A., « Modélisation générique de documents multimédia par métadonnées : mécanismes d'annotation et d'interrogation », Thèse de Doctorat de L'université de Paul sabatier Toulouse, France. 2005

[Laborie S., 2008] Laborie S., adaptation sémantique de documents multimédia, Doctorat de l'Université de Joseph Fourier-Grenoble 1, 2008.

[Laufer R. et Scavetta D., 1992] Laufer R. et Scavetta D., « Texte, hypertexte, hypermédia, Paris, PUF, 1992.

[Landow G. P., 1992] Landow G. P. (1992): Hypertext: the convergence of contemporary critical theory and technology, *John Hopkins University Press*.

[Lementini E.C. et al., 1993] Lementini E.C., Felice P., Oosterom D. et Van P., "A Small Set of Formal Topological Relationships Suitable for End-User Interaction", dans le Proc. of 3rd International Symposium on Advances in Spatial Databases, Berlin Heidelberg New York, juin 1993, Springer Verlag, LNCS n° 692, p. 277-295.

[Luc C., 2001] LUC Christophe, « Une typologie des énumérations basée sur les structures rhétoriques et architecturales du texte », *Actes de TALN 2001, Université de Tours*, juillet 2001, 263-272.

[Mann W.C. et al., 1998] Mann, W.C., Thompson, S.A. 1988. Rhetorical Structure Theory : Toward a functional theory of text organization. Text, 8(3). 243-281.

[Marcoux, Y., 1994] Marcoux, Y. (1994). « Les formats normalisés de documents électroniques » ICO. Intelligence artificielle et sciences cognitives au Québec, 6(1-2), 56-65.

[Mbarki M., 2008] Mbarki M., Gestion de l'hétérogénéité documentaire : le cas d'un entrepôt de documents multimédias., Thèse de Doctorat de l'Université de Paul Sabatier, Toulouse 3 France, 2008.

[Metzger J. P. et al., 2004] Metzger J. P., et Lallich-Boidin, G. (2004). « Temps et documents numériques ». *Document numérique*, (2004/4), 11–21.
[Nanard M., 1996] Nanard M., and all. La métaphore du généraliste : acquisition et utilization de la connaissance macroscopique sur une base de documents techniques. In Acquisition et Ingénierie des Connaissances - Tendances actuelles. N.Aussenac-Gilles, P. Laublet, C. Reynaud. Toulouse : CEPADUES, pages 285–304, 1996.

[Nassr N., 1999] N. Nassr, "*Oragnisation et Indexation automatique de documents multilingue"*. Rapport de DEA 2IL de l'université Paul Sabatier. 1999.

[Papadias D., et al., 1997] Papadias, D., et Theodoridis , Y. (1997). "Spatial relations, minimum bounding rectangles, and spatial data structures". *International Journal of Geographical Information Science*, 11(2), 111–138.

[Poullet L., 1997] L. Poullet. « Formaliser *la sémantique des documents – Un modèle unificateur »*, Actes de la Xème Conférence INFORSID'1997, Toulouse, juillet 1997. pp. 339-352.

[Portier P.E., 2010] Pierre-Edouard PORTIER « Construction des Documents Multistructurés dans le Contexte des Humanités Numériques », Thèse de Doctorat de l'INSA De Lyon France, 2010.

[Poullet L. et al., 1997] Poullet L., Pinon J.M., Calabretto S.. Semantic Structuring of Documents. Proceedings of the Third Basque International Workshop on Information Technology, BIWIT'97, Biarritz, July 1997, pp. 118–124.

[Ramel J-Y., 2006] Jean-Yves Ramed **:** Habilitation à diriger les recherches UNIVERSITE FRANCOIS RABELAIS DE TOURS, 2006.

[Roxin I. et Mercier D., 2004**]** Roxin Ioan, Mercier D., "Multimédia, les fondamentaux : Introduction à la représentation numérique", Vuibert, 2004.

[Roger T. et al., 2003] Roger T. « Pédauque. Document : forme, signe et médium, les reformulations du numérique ». Working paper. Version 3 du 08 juillet 2003.

[Roisin C., 1999] Roisin Cécile. *Document structurés multimédias*. Habilitation à diriger les recherches. Institut National Polytechnique de Grenoble, septembre 1999.

[Roisin C., 1998] Roisin Cécile : "Authoring structured multimedia documents". In Proceedings of the Conference on Current Trends in Theory and Practice of Informatics, pages 222-239, 1998.

[Ros J. et al, 2005] Julien Ros, Christophe Laurent, Jean-Michel Jolion and Isabelle Simand, "Comparing string representation and distances in a natural images classification task. In Graphbased representations" in Pattern Recognition, pages 72-81, 2005.

[Salton G., 1971] Salton G. (1971). The SMART Retrieval System – experiments. *in automatic document processing. U Perntice-Hall, Inc., Englewood Cliffs, NJ.*

[Scuturici M., 2002] Scuturici M., *Contrinbution aux techniques orientée objet de gestion des séquences vidéo pour les serveurs Web*, PhD, INSA Lyon, 2002, 118 p.

[Tagarelli A., 2010] Andrea Tagarelli, Sergio Greco: "Semantic clustering of XML documents". ACM Trans. Inf. Syst. 28(1): (2010).

[Tannier X., 2006] Tannier X., « Traitement automatique du langage naturel pour l'extraction et la recherche d'information ». Technical report, Ecole Nationale Sup_erieure des Mines de Saint-Etienne, July 2006. 16, 17.

[Thuong T.T., 2003] Tien TRAN THUONG, « *Modélisation et traitement du contenu des médias pour l'édition et la présentation de documents multimédias »*.Thèse de Doctorat de l'Institut National Polytechnique de Grenoble, 2003.

[Tennison J. et al., 2002] Tennison, J., et Piez, W. (2002). « The Layered Markup and Annotation Language » (LMNL). Extreme Markup, Montreal.

[Termier A. et al., 2002] Termier A., Rousset, M. C., et Sebag, M. (2002). "TreeFinder: a First Step towards XML Data Mining", Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02), IEEE Computer Society Washington, DC, USA, 450.

[Vazirgiannis M. et al., 1998] Vazirgiannis M., Theodoridis, Y., and Sellis, T. K., "Spatio-Temporal Composition and Indexing for Large Multimedia Applications Multimedia Systems", 6(4), pp. 284-298, 1998.

[Vilain M. et al., 1986] Vilain M., Kautz H. A. (1986). Constraint propagation algorithms for temporal reasoning. In AAAI-86. p. 132-144.

[Vellucci L., 1998] Vellucci L, "Metadata", Annual Review of Information Science and Technology, Vol. 33, 1998, p 187-222.

[Vercoustre A.M. et al., 2006] Vercoustre, A. M., Fegas, M., Lechevallier, Y., Despeyroux, T., et Rocquencourt, I. (2006). « Classification de documents XML à partir d'une représentation linéaire des arbres de ces documents ». Paris, France, 433–444.

[Yi J. et Sundaresan N., 2002] Yi, J., et Sundaresan, N. (2000). "A classifier for semi-structured documents". *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM New York, NY, USA, 340-344.

[Weiss M.A., 1998] Weiss M. A. *Data Structures And Algorithm Analysis In C*, 1998.

[Woodhead N., 1991] Woodhead N.: "Hypertext and hypermedia: theory and applications", London: Sigma Press and Addison-Wesley Co, 231 p, 1991.

…. Mais d'abord, Pourquoi classer?

Diviser pour mieux régner.

Organiser pour mieux gérer.

Chapitre II - Classification structurelle des documents multimédias : état de l'art

Résumé du chapitre :

La première section de ce chapitre est consacrée à une présentation globale de la comparaison des graphes afin de faciliter la lecture de ce document. Dans la deuxième section nous abordons quelques mesures standards de distance/similarité et dans la troisième section, nous exposons un état de l'art sur la classification en général. Nous présentons aussi quelques travaux connexes sur la classification documentaire. En fin, nous donnons un bref aperçu sur quelques techniques de validation des résultats d'une classification automatique.

Sommaire chapitre II

I. Comparaison de graphes

I.1. Introduction

Un grand nombre de travaux utilisent les graphes pour représenter les entités structurées : les nœuds représentent les composants des entités et les arêtes représentent les relations entre ces composants. Cette représentation permet de manipuler plus facilement les objets et leurs relations. En général, la représentation graphique des objets permet de faciliter leur lecture, leur analyse, leur interprétation, etc. L'ensemble des outils mathématiques mis au point en théorie des graphes présente un grand intérêt puisqu'il permet de démontrer des propriétés, déduire certaines caractéristiques d'un objet à partir d'un autre, etc.

De nombreuses applications nécessitent de comparer les graphes afin de déterminer si leurs structures sont identiques (ou similaires) ou si une structure est incluse dans l'autre. La comparaison des graphes, nécessite d'établir une correspondance entre leurs composants (nœuds et arêtes). En théorie des graphes, le problème de comparaison de graphes se traduit par la recherche d'un isomorphisme de (sous) graphes permettant de montrer l'inclusion ou l'équivalence entre les graphes à comparer. Plus généralement, il s'agit de trouver le meilleur appariement « best matching » possible entre les graphes à comparer : un appariement qui met en correspondance le maximum de nœuds et d'arêtes similaires. Cela permet d'évaluer les caractéristiques qu'ils ont en commun et de pouvoir en conséquent qualifier la similarité des deux graphes, c'est-à-dire de mettre en évidence la ressemblance et la différence entre les objets représentés par ces graphes.

Avant d'aborder l'objet central de cette section, à savoir la comparaison de graphes, nous rappelons quelques généralités et notions de bases sur les graphes.

I.2. Concepts de bases sur les graphes

I.2.1. Relation

Soient A et B deux ensembles non vides. On dit que $\mathcal{R}$ est une *relation binaire* définie de A vers B si et seulement s'il existe des éléments de A auxquels on peut associer, par une règle précise $\mathcal{R}$ (non ambiguë) des éléments de B. On écrit ensuite :

$$\mathcal{R} : A \rightarrow B$$
$$x \rightarrow y \text{ ou } x \,\mathcal{R}\, y$$

Autrement dit, une relation binaire $\mathcal{R}$ de A vers B est définie par une partie G de $A \times B$ telle que : si $(x,y) \in G$ alors x est en relation avec y et on note : $x \,\mathcal{R}\, y$. G est appelé le *graphe de la relation* $\mathcal{R}$.

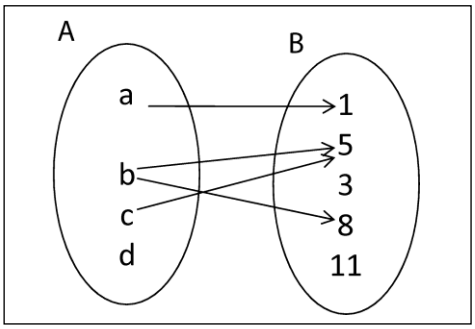


Figure II.1 - Exemple de relation entre deux ensembles

Dans l'exemple de la figure II.1, le graphe de la relation est : $G = \{(a,1),(b,5),(b,8),(c,5)\}$

I.2.2. Graphe

Formellement, *un graphe G* est défini par un couple (*V,E*) tel que :

- *V* est un ensemble fini de nœuds,
- et $E \subseteq V \times V$ est un ensemble d'arêtes (relations entre les nœuds de *G*).

- *Graphe orienté*

Un graphe *G* = (*V,E*) est dit *orienté* (ou d*igraphe*) si les arêtes sont orientées. On parle alors d'arcs (exemple, figure II.2).

Un *arc* e ($e \in E$) est un lien entre deux nœuds. L'arc (u,v) ($(u,v) \in V \times V$) est caractérisé par un nœud initial *u* et un nœud terminal *v*. Un arc possède une direction souvent symbolisée par une flèche.

Les nœuds *u* et *v* sont *adjacents* si (u,v) $\in E$, dans ce cas on dit que *u* est le *précédant* (ou *père*) de *v* et que *v* est le *suivant* (ou *fils*) de *u*. Un nœud *feuille* est un nœud qui ne possède aucun *fils*.

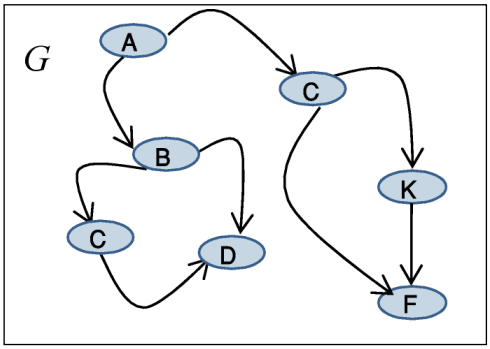


Figure II.2 - Exemple de graphe orienté

- *Arcs adjacents*

Deux arcs d'un graphe *G* sont adjacents s'ils ont au moins une extrémité commune.

- *Chaîne*

Dans un graphe, une chaîne est définie par une suite de sommets adjacents.

- *Chemin*

Soit $G = (V,E)$ un graphe orienté. Un *chemin de longueur k* est une suite de nœuds $u_0,u_1, ...,u_k$ de V tels que :

$\forall i \in [0,k\text{-}1]$, u_i et u_{i+1} sont adjacents, u_0 est l'origine du chemin u_k l'extrémité.

- *Cycle*

Dans un graphe, un *cycle* est un chemin dont l'origine est égale à l'extrémité.
Un graphe *acyclique* est un graphe qui ne contient aucun cycle (exemple, figure II.3).

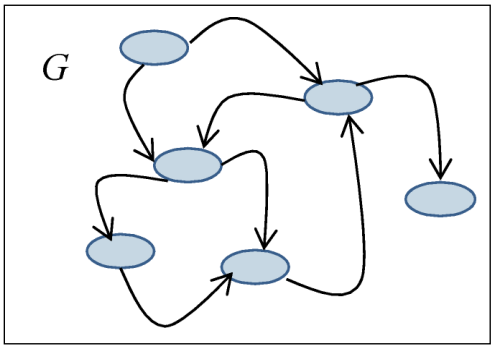


Figure II.3 - Exemple de graphe avec cyclique

Dans l'exemple de la figure II.2, *G* est un graphe acyclique (sans cycle).

- *Graphe connexe*

Un graphe est connexe lorsque, pour tout couple de sommets, il existe une chaîne ou chemin qui les relie.

- *Arbre*

Un arbre est un cas particulier de graphe. C'est un graphe connexe et sans cycle.

- *Chemin simple*

Un *chemin simple* est un chemin qui ne contient pas plus d'une fois le même nœud. Dans l'exemple de la figure II.2, tous les chemins sont simples.

- *Profondeur*

La *profondeur* d'un nœud *u* est la longueur du chemin qui va de la racine à ce nœud. Dans l'exemple de la figure II.2, la profondeur du nœud *K* est 2, la profondeur de *F* dépend du chemin parcouru. Si l'on considère le chemin *A/C/K/F*, la profondeur *F est* 3.

- *Graphe étiqueté*

Un *graphe étiqueté* est un graphe où les nœuds et les arêtes sont affectés d'une valeur (nombre, lettre, mot, etc). Ces valeurs sont appelées étiquettes (exemple, figure II.4).

Un graphe *G* étiqueté peut être défini par le quadruplet (V,E,f,g) où f (resp g) est une fonction d'étiquetage des nœuds (resp. des arêtes). Formellement :

$$f : V \rightarrow L_V \quad \text{et} \quad g : E \rightarrow L_E$$
$$u \rightarrow f(u) \qquad\qquad e \rightarrow g(e)$$

où L_V (resp. L_E) : est l'ensemble des étiquettes des nœuds (resp. de arêtes).

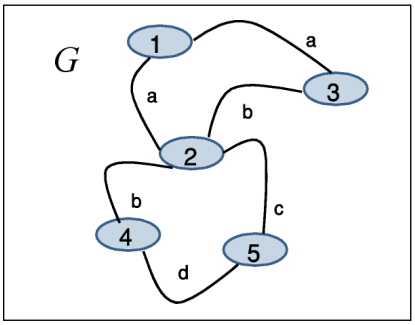


Figure II.4 - Exemple de graphe étiqueté

- *Graphe pondéré*

Un graphe *pondéré* est un graphe étiqueté où chaque arête est étiquetée par un nombre positif appelé poids de l'arête.

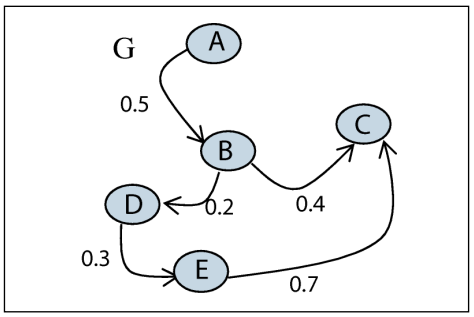


Figure II.5 - Exemple de graphe pondéré

Dans l'exemple de la figure II.5, *G* est un graphe pondéré. Le poids d'un chemin est la somme des poids des arêtes qui le composent.

Le plus court chemin (dans graphe orienté et pondéré) entre deux nœuds est le chemin de poids minimal entre ces nœuds.

- *Graphe biparti*

Un graphe $G = (V,E)$ est *biparti* si l'ensemble V de ses nœuds peut être partitionné en deux ensembles X et Y *tel que* :

- $V = X \cup Y$ avec $X \cap Y = \varnothing$,
- $\forall (a,b) \in E$; *si* $a \in X$ *alors* $b \in Y$ *et si* $a \in Y$ *alors* $b \in X$.

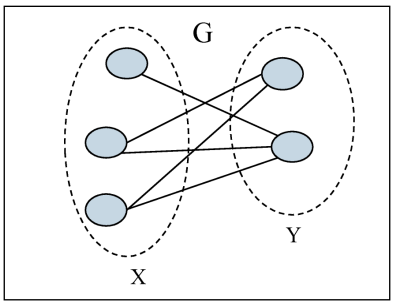


Figure II.6 - Exemple de graphe biparti

Le graphe *G* de la figure II.6 est un graphe biparti.

- *Sous-graphe*

Soient $G = (V,E)$ et $G' = (V',E')$ deux graphes orientés ou non. G est un *sous-graphe* de G' si et seulement si :
G est engendré par $V \subseteq V'$ tel que les nœuds sont les éléments de V' et les arcs (ou arêtes) sont les arcs (ou arêtes) de G' dont les extrémités appartiennent à V'.

Formellement :

G est un *sous-graphe* de $G' \Leftrightarrow V \subseteq V'$ et $E \subseteq E'$.

- *Sous-graphe partiel*

Soient $G = (V,E)$ et $G' = (V',E')$ deux graphes orientés ou non. G est un *sous-graphe partiel* de G' si et seulement si :

$V \subseteq V'$ et $E \subseteq E' \cap V \times V$

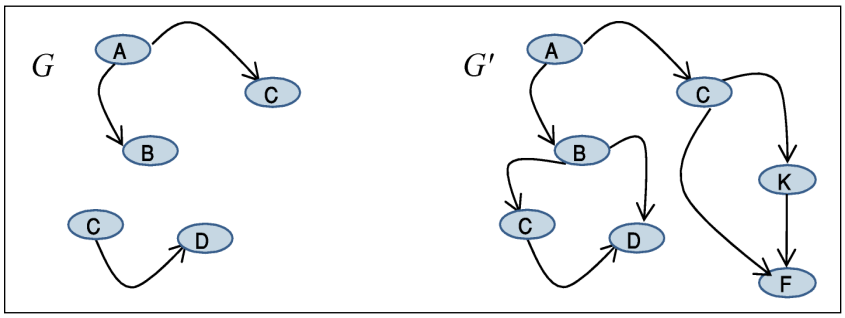


Figure II.7 - Sous-graphe partiel d'un graphe

Dans l'exemple de la figure II.7, le graphe G est un sous-graphe partiel de G'.

- *Sous-graphe induit*

Soient $G = (V,E)$ et $G' = (V',E')$ deux graphes orientés ou non. G est un *sous-graphe induit* de G' si et seulement si G contient tous les arcs ou arêtes de G' ayant leurs extrémités dans V.

Formellement :

G est un *sous-graphe induit* de $G' \Leftrightarrow V \subseteq V'$ et $E = E' \cap V \times V$.

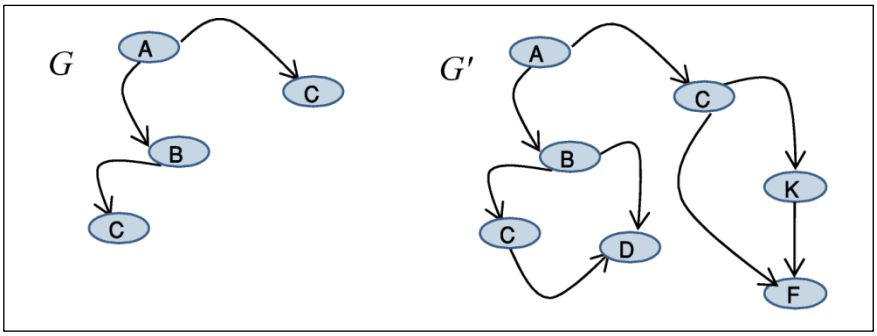


Figure II.8 - Sous-graphe induit d'un graphe

Dans l'exemple de la figure II.8, G est un sous-graphe induit de G'.

- *Sous-graphe induit par un sous ensemble de nœuds d'un graphe*

Soit $G = (V,E)$ un graphe orienté ou non et V_1 une partie de V. Un sous-graphe de G induit par V_1 noté $G' = (V_1,E(V_1))$ est obtenu en supprimant un ou plusieurs nœuds de G, ainsi que tous les arcs ou arêtes incidents à ces nœuds.

Remarque : un sous-graphe induit est aussi un sous-graphe partiel (l'inverse n'est pas vrai).

I.3. Appariement de graphes

Un appariement entre les graphes $G = (V,E)$ et $G' = (V',E')$ est une relation qui permet la mise en correspondance entre les nœuds et les arêtes de ces graphes. Une mise en correspondance des sommets et des arêtes entre deux graphes permet de retrouver facilement un maximum de caractéristiques communes de ces graphes.

Formellement, un appariement de nœuds entre G et G' est une relation m de V *vers* V' telle que $\forall (u,v) \in V \times V' ; m(u)=v$ signifie que u et v sont appariés.

Les graphes ont été utilisés pour la modélisation des données structurées et complexes dans différents domaines d'applications : en reconnaissance de formes [Conte D. et al., 2004], en chimie [Klinger S. et Austin J., 2005], [Suard F. et al., 2007], [Wieczorek S., 2009], en bio-informatique [Hu et al., 2005], pour représenter les documents XML [Zhang et al., 2006], etc. Dans toutes ces applications, il s'agit de rechercher un lien entre les graphes : l'inclusion, l'intersection, etc. Ce sont sur ces différents travaux de ces différents domaines que nous allons appuyer notre état de l'art.

I.3.1. Appariement univoque, multivoque

L'*appariement* univoque permet de mettre en correspondance un nœud et/ou une arête d'un graphe avec au plus un nœud et/ou une arête de l'autre graphe.

L'*appariement univoque* peut ne pas répondre à certains besoins, par exemple lorsqu'il s'agit d'apparier des objets où une composante de l'un des objets doit être mise en correspondance avec plusieurs composantes de l'autre objet. Dans ce cas, on parle d'*appariement multivoque* (un à plusieurs). Dans [Sorlin S. et al, 2006], les auteurs ont utilisé l'appariement multivoque pour comparer des graphes représentant des images, ce qui a permis de prendre en compte le problème de sur et de sous-segmentation des images.

I.3.2. Sous-graphe induit par un appariement

Soit m un appariement entre les graphes $G= (V, E)$ et $G'= (V',E')$. Un sous-graphe de G *induit par l'appariement* m est un sous-graphe de G noté $G_m=(V_m,E_m)$ induit par l'ensemble des nœuds de G appariés.

Formellement :

$G_m = (V_m,E_m)$ est un sous-graphe de G induit par m si et seulement si :

$V_m = \{u \in V / \exists\ u' \in V'; m(u)=u'\}$ et $E_m = \{(u,v) \in E / \exists\ (u',v') \in E'; m(u,v) = (u',v')\}$.

Dans l'exemple de la figure II.9, $G_m = (V_m,E_m)$ où $V_m = \{1,2,3,4\}$ et $E_m = \{(1,2),(2,3),(1,4)\}$, est un sous graphe induit par l'appariement m.

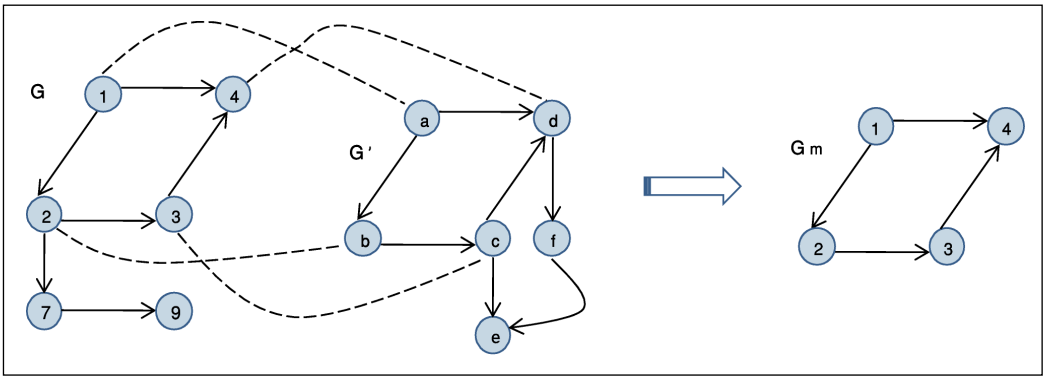


Figure II.9 - Sous-graphe induit par un appariement

I.3.3. Isomorphisme de graphes

Les graphes $G = (V, E)$ et $G' = (V', E')$ sont *isomorphes* si et seulement s'il existe une fonction bijective f de V sur V' qui préserve les arcs :

$$\forall (u,v) \in V \times V ; (u,v) \in E \Leftrightarrow (f(u), f(v)) \in E'.$$

L'isomorphisme de graphes permet de montrer que deux graphes sont structurellement identiques. En effet, si deux graphes sont isomorphes alors ils contiennent les mêmes nœuds et qui sont liés par les mêmes arêtes.

L'isomorphisme entre deux objets permet de démontrer leur similitude, ce qui permet de transposer des résultats et des propriétés, déjà démontrés, d'un objet à l'autre. Dans certains cas, le problème consiste à vérifier si deux graphes ont des structures totalement identiques. Ce genre de problème repose sur la recherche d'un appariement univoque des nœuds et ou des arêtes des deux graphes respectant un ensemble de contraintes.

I.3.4. Isomorphisme de sous-graphes

Soient G et G' deux graphes orientés. On dit que G est isomorphe à un sous-graphe de G' si et seulement s'il existe une fonction injective f de V vers V' telle que :

$$\forall (u,v) \in V \times V ; (u,v) \in E \Rightarrow (f(u), f(v)) \in E'.$$

Dans l'exemple de la figure II.10, le graphe G est isomorphe à un sous-graphe de G'.

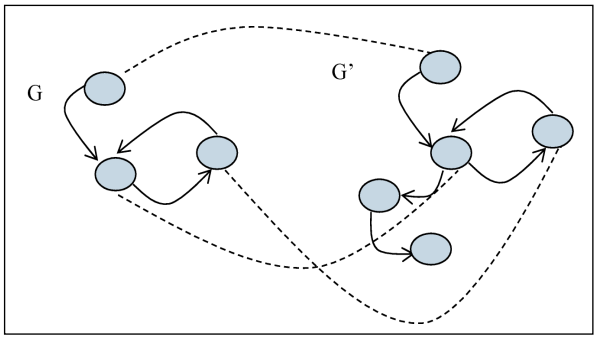


Figure II.10 - Isomorphisme de sous-graphe

Dans certaines applications, il est intéressant de savoir si un objet est une partie d'un autre objet. Par exemple, en chimie organique, il peut être utile de vérifier si une molécule entre dans la composition d'une autre molécule. En recherche documentaire, il peut parfois être

intéressant de vérifier qu'un document est inclus dans un autre ou si une image fait partie d'une autre image. Dans [Sorlin S. et Solnon C., 2005], pour montrer l'équivalence ou l'inclusion de deux graphes, il suffit de montrer que les deux graphes sont isomorphes ou que l'un des graphes est un sous-graphe isomorphe à l'autre.

I.3.5. Plus grand sous-graphe commun

Un *sous-graphe commun* de deux graphes *G* et *G'* est un sous-graphe *isomorphe* à des sous-graphes de *G* et *G'*. Un *plus grand sous-graphe commun* à deux graphes *G* et *G'* est un sous-graphe commun à *G* et *G'* ayant le maximum de nœuds et d'arêtes (isomorphisme de sous-graphe maximum).

Dans l'exemple de la figure II.11, le graphe *G"* est le plus grand sous-graphe commun de *G* et *G'*.

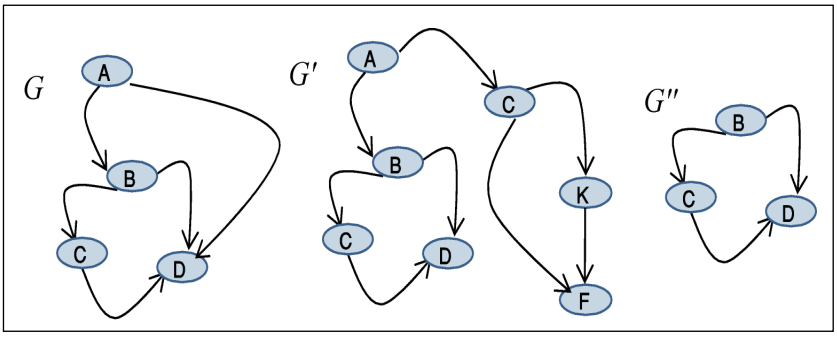


Figure II.11 - Plus grand sous-graphe commun de deux graphes

G " est isomorphe à la fois à un sous-graphe de *G* et à un sous-graphe de *G'*. Pour comparer deux objets, il est intuitif de calculer le rapport entre la quantité des caractéristiques communes (*taille de leur plus grand sous-graphe commun*) à ces deux objets et l'ensemble de leurs caractéristiques [Lin D., 1998].

I.4. Conclusion

Dans cette section, nous avons abordé les appariements couramment utilisés dans le processus de comparaison de graphes. L'isomorphisme de graphes permet de vérifier que deux graphes sont identiques (ou semblables), l'isomorphisme de sous graphes permet de vérifier qu'un graphe est inclus dans un autre, et la recherche du plus grand sous-graphe commun permet d'évaluer l'intersection de deux graphes.

En général, le problème de comparaison de deux graphes se traduit par la recherche d'un meilleur appariement entre ces graphes. Les approches générales proposées par la théorie des graphes concernant l'appariement sont de deux types : *l'appariement exact* et *l'appariement approximatif*. Dans de nombreux domaines d'application, le but n'est pas de montrer que deux graphes sont structurellement identiques (ou qu'un graphe est inclus dans un autre) mais il est plus intéressent de savoir à quel point ces graphes sont similaires. Dans de telles applications, la similarité des graphes basée sur *l'appariement exact* n'est pas appropriée. Par exemple, en recherche d'information l'appariement exact n'est pas suffisant [Sanz I. et al., 2005] puisqu'on ne cherche pas à répondre d'une façon exacte au besoin, en information, de l'utilisateur. Pour cela, des *appariements approximatifs* de graphes, à tolérance d'erreurs, basés sur la recherche du plus grand sous-graphe commun ou sur le

calcul de la distance d'édition de graphes, ont été proposés dans les travaux de [Bunke H. et Shearer K., 1998], [Wallis W.D. et al., 2001], [Conte D. et al., 2004] et [Hidovi D. et al., 2004]. En effet, les graphes comparés peuvent ne pas avoir nécessairement la même structure dans leur globalité et donc l'appariement recherché doit mettre en correspondance les parties structurellement identiques (ou similaires) des graphes comparés.

L'appariement entre deux graphes, se heurte au problème lié à une explosion de combinaisons à explorer : connu en théorie des graphes par le problème combinatoire. Dans [Garey M. et al., 1979], l'isomorphisme de sous-graphe est un problème NP-complet. La complexité augmente en fonction de la taille des graphes manipulés. Quelques approches ont proposé de réduire cette complexité en fixant des contraintes sur le type des graphes, par exemple l'utilisation des graphes planaires [Hopcroft J.E. et al., 1974]. L'arbre de décision [Messmer B.T. et al., 1999] permet de résoudre le problème du temps mais au détriment d'un prétraitement très couteux en temps et en mémoire. L'approche proposée par [Cordella L.P. et al., 2001] traite les graphes orientés et ne s'intéresse qu'à la structure des graphes et ne tient pas compte des propriétés associées aux sommets et aux arcs, pourtant ces informations sont indispensables dans la plupart des applications.

Ce problème rend la plupart des approches limitées à des graphes de petite taille [Sorlin S. et al 2006]. Un problème qui reste ouvert et plusieurs approches on été proposées pour réduire le coût combinatoire.

Dans la section suivante, nous présentons quelques mesures de similarité (ou de distance) standards couramment utilisées dans la littérature.

II. Mesure de similarité

II.1. Introduction

La similarité est un type de comparaison qui permet de juger d'une relation de proximité entre deux objets [Medin D. et al., 1993].

La similarité entre deux objets permet de mesurer ce que ces objets ont en commun. Par conséquent, plus ils ont des caractéristiques en commun, plus leur similarité sera importante. Inversement, plus la différence (dissemblance), entre deux objets, est grande plus leur similarité est faible. La similarité maximale, entre deux objets, est atteinte lorsque ces objets sont identiques.

La mesure de similarité est une étape cruciale dans un processus de comparaison, car les résultats de ce processus dépendent fortement de la mesure utilisée. Généralement, pour comparer deux objets, on a recours de manière implicite ou explicite, à un opérateur permettant d'évaluer la ressemblance ou la différence entre ces deux objets. Dans la littérature, cet opérateur est souvent désigné par une fonction de similarité ou mesure de similarité ou de dissimilarité ou tout simplement similarité.

Intuitivement, similarité et dissimilarité sont étroitement liées. Dans un certain nombre d'applications, il s'agit de trouver, parmi un ensemble d'objets, le ou les objets les plus proches ou semblables à un objet donné. Mais, il peut s'agir aussi de vérifier si un objet est un sous-objet d'un autre ou encore de déterminer tous les points communs et la différence entre deux objets donnés.

Généralement, les mesures de similarité s'appuient sur la notion mathématique de distance. Ainsi, deux objets sont d'autant plus similaires, au sens de cette distance, que leur distance est plus faible. Pour une dissimilarité, le lien sera d'autant plus fort que sa valeur de dissimilarité est petite [Celeux G. et al., 1989]. Il est cependant difficile de définir « a priori » une distance universelle [Bisson G., 2000].

II.2. État de l'art sur les mesures de similarité

L'évaluation de la proximité de deux objets est un problème qui a motivé de nombreux travaux et dans des domaines variés que ce soit en reconnaissance des formes, en vision assistée par ordinateur, en analyse des données, en recherche d'information, en classification des objets, etc. De nombreuses mesures ont été proposées pour des objectifs applicatifs et des contextes variés. En comparaison d'images [Sorlin S. et al., 2006], en recherche d'images [Smeulders A.W. et al., 2000], en biologie [Steichen O. et al., 2006] et [Lord P et al., 2003], pour comparer des objets de conception assistée par ordinateur [Champin P.A. et Solnon C., 2003], etc. Dans [Champclaux Y., 2009], il existe quatre modèles de similarité : (1) les modèles basés sur *les caractéristiques*, la comparaison de deux objets représentés par deux ensembles (leurs caractéristiques) revient à comparer ces deux ensembles, (2) les modèles *géométriques*, (3) les modèles à *alignement structurel*, (4) les modèles basés sur la notion de *distance transformationnelle* : la similarité entre deux objets est fonction du nombre d'opérations nécessaires pour transformer la structure de l'un des objets en la structure de l'autre.

Les modèles à alignement structurel ont été abordés dans les travaux de [Gentner D., 1983] et [Holyok K.J. et Tagard, 1989]. Plusieurs travaux ont utilisé l'alignement structurel pour comparer des objets. Dans ce contexte, nous citons entre autres [Bassok M. et al., 1997] pour la comparaison des phrases, [Mbarki M., 2008], [Aitelhadj A. et al., 2009] pour comparer les structures documentaires représentées à l'aide des arbres, [Djemal K., 2010] pour comparer les graphes représentant les structures multiples des documents multimédia, [Sorlin S. et al., 2006], [Demirci M.F. et al. 2006] pour comparer les images représentés à l'aide des graphes, etc.

Avant de citer quelques mesures de distance ou de similarité issues de l'état de l'art, nous énonçons d'abord ce que c'est qu'une distance au sens mathématique.

II.2.1. La notion mathématique de distance

Une distance métrique d sur un ensemble E est une application de E x E vers $R+$ telle que :

(1) $\forall x,y \in E$; $d(x,y) = 0 \Leftrightarrow x = y$	séparation
(2) $\forall x,y \in E$; $d(x,y) = d(y,x)$	symétrie
(3) $\forall x,y,z \in E$; $d(x,z) <= d(x,y)+d(y,z)$	inégalité triangulaire

La distance dans un espace vectoriel

Soit E un espace vectoriel muni d'une base orthonormée, une distance d sur E est définie à partir de la norme vectorielle comme suit :

$$\forall (x,y) \in E \text{ x } E,\ d(x,y)=||x-y||$$

En particulier si $E = R^n$ alors $\forall (x,y) \in E$ x E, x et y peuvent être écrits respectivement par : $x = (x_1,x_2,..x_n)$ et $y = (y_1,y_2,...y_n)$.

Dans les sections suivantes, nous présentons les mesures de distance ou de similarité standards les plus connues dans la littérature.

II.2.1.1. La distance de Minkowski

La distance de Minkowski entre x et y est définie par :

$$d(x,y) = \sqrt[p]{\sum_{i=1}^{n}(x_i - y_i)^p} \qquad [1]$$

p est un paramètre réel (positif) qui dépend de l'application de cette distance. La distance de Minkowski est générique dans le sens où d'autres distances sont des cas particuliers de celle-ci.

II.2.1.2. La distance de Manhattan

La distance de Manhattan entre x et y est définie par :

$$d(x,y) = \sum_{i=1}^{n} | x_i - y_i | \qquad [2]$$

Cette distance peut être vue comme un cas particulier de la distance de Minkowski (cas où p=1).

II.2.1.3. La distance euclidienne

La distance euclidienne entre x et y est définie par :

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i, y_i)^2} \qquad [3]$$

La distance euclidienne peut être vue comme un cas particulier de la distance de Minkowski (cas où p=2).

Exemple

Soient les documents d_1=document numérique, d_2=document multimédia numérique, d_3= document.

Le lexique de ces documents est L= {document, multimédia, numérique}. Le vecteur v_1, v_2 *et* v_3 représentant respectivement d_1, d_2 et d_3 sont v_1 (1, 0,1), v_2 (1, 1,1) et v_3 (1, 0,0).

- En utilisant la distance euclidienne on obtient :

  d (v_1, v_2)=1 et d (v_2, v_3)=1,4142.

- En utilisant la distance Manhattan on obtient :

  d (v_1, v_2)=1 et d (v_2, v_3)=2.

II.2.2. Mesure de similarité ou de dissimilarité

Parmi les mesures standards, les plus connues, nous citons le coefficient de *Jaccard*, celui de *Dice* et la mesure *Cosinus*.

- *Le coefficient de Jaccard*

L'indice ou coefficient de Jaccard (1901) est défini par :

$$Jaccard(X,Y) = \frac{|X \cap Y|}{|X \cup Y| - |X \cap Y|} \quad [4]$$

Dans l'exemple de la section II.2.1.3 page 61, *Jaccard* (d_1. d_2)=0.67, *Jaccard* (d_1. d_3)=0.5

Dans la littérature, la mesure de Jaccard a été utilisée sous plusieurs variantes [5].

$$Jaccard(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \quad [5]$$

- *Le coefficient de Dice*

Le coefficient de *Dice* (1945) est défini par :

$$Dice(X,Y) = \frac{2*|X \cap Y|}{|X \cup Y|} \quad [6]$$

Le coefficient de *Dice* est dérivé du coefficient de *Jaccard* en donnant plus d'importance aux éléments partagés (deux fois plus). Il est lié au coefficient de Jaccard par la relation suivante :

$$Jaccard(X,Y) = \frac{Dice(X,Y)}{2 - Dice(X,Y)} \quad [7]$$

Dans l'exemple de la section II.2.1.3 page 61, *Dice*(d_1. d_2)=0.75, *Dice* (d_1. d_3)=0.66

Les mesures de *Jaccard* et de *Dice* ont été initialement construites pour des analyses écologiques.

- *La mesure Cosinus*

Pour évaluer la similarité entre deux entités représentées par *X* et *Y*, on pourra utiliser la mesure Cosinus :

$$Cosinus(X,Y) = \frac{|X \cap Y|}{\sqrt{|X| . |Y|}} \quad [8]$$

La mesure de Cosinus a été utilisée sous plusieurs formats, par exemple :

$$Cosinus(X,Y) = \frac{u.v}{\|u\| . \|v\|} \quad [9]$$

où u (resp. v) le vecteur représentant la première entité (resp. la deuxième entité) et *u.v* est le produit scalaire de *u* et *v*.

Dans l'exemple de la section II.2.1.3 page 61, *Cosinus*(v_1, v_2)=0.81 et *Cosinus*(v_1, v_3)=0.70.

Issue de l'algèbre linéaire, la mesure de Cosinus a été utilisée dans plusieurs travaux notamment en recherche d'information.

Dans la section suivante, nous évoquons quelques mesures de distance ou de similarité qui ont été utilisées en comparaison des graphes.

- *Distance d'édition de graphes*

La distance d'édition sur les graphes est une extension de la distance d'édition sur les chaînes de caractères de [Levenshtein V., 1966].
La distance d'édition entre deux graphes permet d'évaluer le degré d'*isomorphisme*, entre ces deux graphes. Elle repose sur le *coût minimal* pour transformer un graphe en un autre. Pour cette transformation, on dispose de quelques opérations élémentaires : l'insertion, la suppression et le ré-étiquetage de nœuds et d'arcs, etc. Un coût est associé à chacune de ces opérations et le coût de la transformation est la somme des coûts des opérations élémentaires. La distance entre ces graphes est déterminée par la séquence qui nécessite le moindre coût. Il est donc évident que plus cette distance est grande et plus les graphes sont distants. Cependant, trouver cette séquence est un problème combinatoire et donc la recherche des valeurs pour le moindre coût est non triviale. Dans [Bunke H., 1999], la fonction de coût est très importante pour calculer la distance d'édition entre deux graphes mais le choix d'une telle fonction est parfois très difficile.

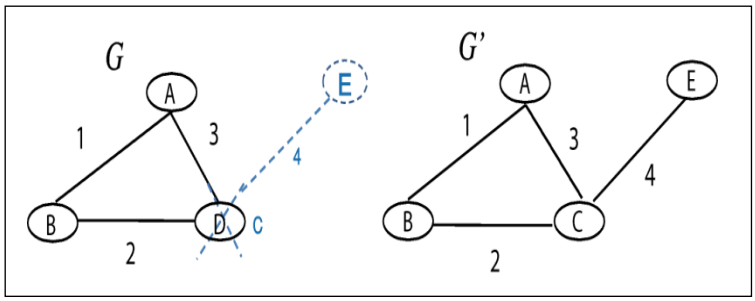


Figure II.12 - Transformation d'un graphe en un autre : distance d'édition

Dans cet exemple (figure II.12), pour transformer *G* en *G'*, on doit faire les transformations suivantes :

- renommer un nœud de *G* (*D* devient *C*),
- insérer un nœud (le nœud *E*) dans *G*,
- insérer une arête et son étiquette dans *G*.

Pour évaluer la proximité de deux graphes, [Kriegel H.P. et al., 2003] proposent une fonction de coût de mise en correspondance des arcs de ces graphes qui permet de calculer la distance entre deux graphes par la recherche du coût minimal pour maximiser le nombre d'arcs mis en correspondance. Dans leurs travaux de classification de documents, les auteurs de [Dalamagas T. et al., 2004] et de [Dalamagas T. et al., 2006] ont utilisé la distance d'édition sur des résumés d'arbres.

- *Similarité basée sur le plus grand sous-graphe commun*

Le plus grand sous-graphe commun de deux graphes permet de déterminer les points communs entre ces deux graphes. Par conséquent, plus le nombre de points communs des deux graphes est grand plus les deux graphes se ressemblent. Deux graphes sont alors jugés d'autant plus similaires que leur intersection est grande [Lin D., 1998] et [Tversky A., 1977]. Dans [Bunke H., 1997], il a été démontré que le plus grand sous-graphe commun et la distance d'édition entre graphes donnent des résultats équivalents. Bunke a proposé la relation suivante, entre la distance d'édition et le plus grand sous-graphe commun (*mcs* : maximal common subgraph) :

$$d(G,G') = |G| + |G'| - 2|mcs(G,G')| \qquad [10]$$

La mesure de similarité basée sur le plus grand sous-graphe commun (*mcs*) a été proposée dans les travaux de [Bunke H. et Shearer, 1998] :

$$Sim_{mcs}(G,G') = \frac{|mcs(G,G')|}{Max(|G|,|G'|)} \qquad [11]$$

où |mcs(*G*,*G*')|, | *G* | et | *G'*| sont respectivement la taille de *mcs*, *G* et *G*'.

Dans les travaux de [Yan X. et al., 2005] et [Shang H. et al., 2010], le sous-graphe commun maximal a été utilisé pour calculer la similarité entre graphes. La similarité basée sur la recherche d'un plus grand sous-graphe commun est plus flexible. L'appariement recherché détermine les parties communes (les parties appariées : ensemble de nœuds et d'arêtes) des graphes comparés.

II.3. Conclusion

Dans la littérature, plusieurs travaux ont introduit (ou utilisé) une mesure de distance ou de similarité pour évaluer la proximité entre objets représentés à l'aide des graphes. Cependant, dans un processus de comparaison de deux graphes, les mesures de similarité ou de distance, ne sont pas définies dans les mêmes contextes ce qui impose, à chaque fois, des contraintes sur l'appariement recherché entre les graphes. Ces contraintes dépendent du problème à résoudre, du domaine d'application, des modèles utilisés pour représenter les objets etc. Il faut donc trouver à chaque fois un compromis entre l'objectif et le comportement effectif de la mesure définie. Par conséquent, il est difficile de comparer et d'appréhender les résultats de nombreux travaux qui ont abordé ce sujet.

Dans [Atteneave F., 1950] et [Thibaut J.P., 1997], la distance de Manhattan est utilisée pour comparer des attributs psychologiquement et/ou physiologiquement séparables. Selon [Gardenfors P., 2000], la distance euclidienne est le meilleur choix pour comparer deux attributs liés. Les mesures de similarité de *Jaccard*, de *Dice* et de Cosinus sont largement utilisées dans le domaine de la recherche d'information. En revanche, les auteurs de [Witten I.H. et al., 1999] préconisent l'utilisation de la mesure de Cosinus, en recherche d'information, au lieu de la distance de Minkowski car celle-ci est plus appropriée pour la manipulation des vecteurs pondérés.

Les auteurs de [Hummel J.E., 2000] et [Hummel J.E., 2001] ont montré que les modèles géométriques et les modèles basés sur les attributs ne permettent pas la comparaison des objets structurés. Les modèles à alignement structurel permettent de comparer des

représentations complexes et des structures hiérarchiques. Ils permettent de prendre en compte, non seulement les correspondances entre composants de ces structures mais aussi les ressemblances dans les relations entre ces composants [champclaux Y., 2009].

Le calcul d'une distance, en utilisant le *modèle vectoriel*, ne pose aucun problème à part le choix de la mesure [Bärecke T., 2009]. En revanche, pour les modèles à *alignement structurel*, on doit souvent définir une métrique entre les structures dont le temps de calcul est généralement élevé et qui croit avec la taille des structures manipulées.

Avant d'aborder la classification documentaire, nous faisons un bref tour d'horizon sur la théorie de la classification en général.

III. Classification : notions générales

III.1. Introduction

Le terme *classification* apparaît pour la première fois dans la cinquième édition du dictionnaire de l'Académie Française en 1798 sous la définition : « *distribution en classes et suivant un certain ordre* ».

Dans [Kalakech M., 2011], l'auteur définit la classification comme étant un processus qui vise à découvrir la structure intrinsèque d'un ensemble de données en construisant des groupes de données qui partagent des caractéristiques similaires.

La *classification* peut désigner plusieurs sens, elle peut avoir le sens de *rangement* des objets dans des classes comme elle peut aussi avoir le sens de répartition de ces objets, etc. Des fois, on utilise des termes différents (classer, organiser, ranger, segmenter, distribuer, découper, etc) pour le même objectif. Ainsi, pour classer les objets, on parle de *typologie* en sciences humaines, *segmentation* en marketing, *taxonomie* en biologie et zoologie, *nosologie* en médecine, etc.

La classification a été abordée par le philosophe français Auguste Comte au XIXe siècle. Depuis, elle est utilisée dans des contextes variés en tant que processus d'analyse exploratoire des objets. Elle a été utilisée au dix-septième siècle par [Lavoisier, 1789] pour classer les éléments chimiques. Les biologistes l'utilisent pour regrouper les êtres vivants dans des sous-groupes homogènes. Les premiers travaux sur la classification remontent aux années 1960 [Sokal R.R. et Sneath, 1963]. De nos jours, la classification est utilisée en marketing pour étudier les comportements des clients et par conséquent faciliter la prise de décision. On pourra par exemple classer les clients selon leur attitude par rapport à une marque de produits, ou par leur chiffre d'affaire, etc. En sociologie, on pourra classer les personnes selon leur niveau intellectuel, leur racine, leur niveau de vie, etc.

Dans un certain nombre de domaines, aussi bien académiques qu'industriels, manipulant une grande quantité de données, la classification devient de plus en plus une nécessité. Classer des objets c'est regrouper entre eux des objets semblables. La classification est un processus qui permet d'organiser et de structurer un ensemble d'objets sous forme de classes, aussi homogènes que possible. Les nombreux travaux sur la classification nous ont amenés à poser un certain nombre de questions : pourquoi classer les objets ? Comment les classer ? Y a-t-il une seule démarche pour classer ces objets ?

Les démarches utilisées pour classer des objets diffèrent d'un contexte à l'autre. Cela dépend du domaine d'application, des objectifs visés, etc. Ainsi, classer les mêmes objets en utilisant deux méthodes différentes peut aboutir à des résultats différents.

En général, les objets doivent être représentés d'une façon formelle ensuite utiliser un opérateur de comparaison pour définir la ressemblance et la différence entre ces objets. Sur cette base, on pourra classer ces objets sous forme de classes d'individus similaires.

Dans la section suivante, nous abordons les méthodes de classification les plus connues dans la littérature. Nous débutons la section par une définition formelle de la classification.

III.2. Les méthodes de classification

Étant donné un ensemble $U = \{x_1,x_2,x_3,..,x_n\}$ à n éléments ($n > 0$), la classification des éléments de cette ensemble vise à regrouper ces n éléments en un ensemble de classes $C = \{c_1,c_2,....,c_m\}$ ($m > 0$) de façon à ce que les classes obtenues soient constituées d'éléments les plus semblables possible (*compacité*) et que les classes soient les plus différentes possibles entre elles (*séparation inter-classe*). D'un point de vue mathématique, une classe est un sous-ensemble de l'ensemble des objets à classer [Bouveyron C., 2006].

Formellement, une classe notée C_i peut être définie comme suit :

$$C_i=\{x\in U / p(r_i,x) \text{ est vrai}\}$$

où r_i est le représentant de la classe C_i et p un prédicat traduisant à quel point les éléments de cette classe soient similaires.

Dans [Bouveyron C., 2006], il existe deux approches différentes pour décrire une classe : l'approche *générative* et l'approche *discriminative*. La première décrit une classe par les propriétés caractéristiques des objets qui la composent alors que la seconde décrit une classe par sa frontière avec ses voisines.

Plus généralement, la classification est un processus permettant de recevoir un ensemble U d'objets en entrée et de fournir un ensemble de classes homogènes, selon un critère de classification. Par simplicité, les objets (éléments de U) sont généralement représentés dans un ensemble Ω.

Formellement, un classifieur f peut être défini par :

$$f : \Omega \rightarrow C$$
$$X_i \rightarrow f(X_i)$$

Par exemple, s'il s'agit de classer des documents XML représentés à l'aide des vecteurs, l'ensemble U est l'ensemble des documents à classer, Ω est l'ensemble des vecteurs représentant ces documents.

Exemple

Reprenons l'exemple de la page 61, les documents d_1= {*document numérique*}, d_2= {document multimédia numérique}, d_3={document}. Le lexique de ces documents est L= {*document, multimédia, numérique*}
Le vecteur v_1, v_2 *et* v_3 représentant respectivement d_1, d_2 et d_3 sont v_1(1,0,1), v_2(1,1,1) et v_3(1,0,0).

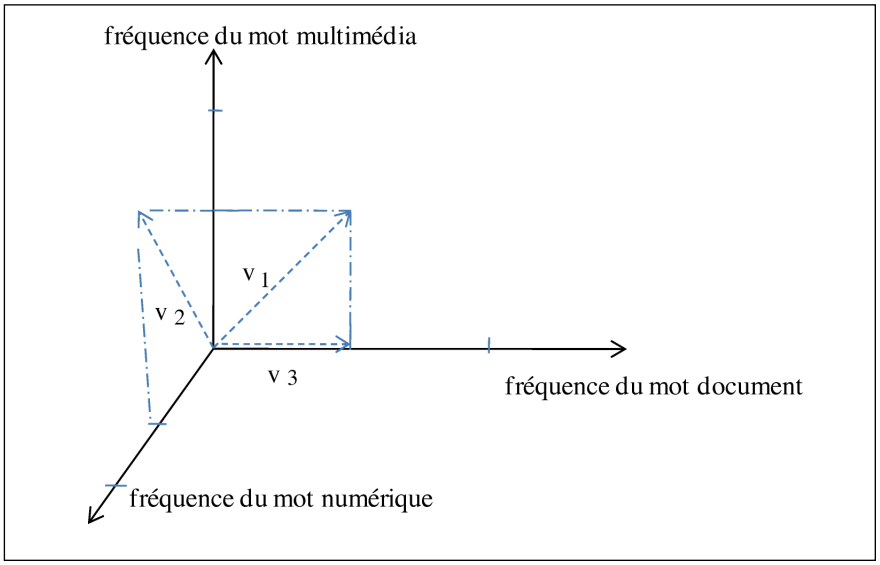


Figure II.13 - Exemple de représentation vectorielle d'un document

Dans cet exemple (figure II.13), un document est une suite de mots, il est représenté par un vecteur élément de R^3.

La classification se décline en deux types : classification supervisée et classification non *supervisée*.

III.2.1. La classification supervisée

La classification supervisée (*classification en anglais*) consiste à associer un objet à une classe parmi un ensemble de classes préalablement définies. Ce type de classification peut être utilisé lorsque toutes les classes sont connues. Dans ce cas, les ensembles Ω et *C* sont connus a priori et il suffit donc de définir la fonction *f* (*classifieur*). Le rôle du classifieur, dans ce cas, est de déterminer, selon un critère de classification, quelle est la classe qui correspond au mieux à chaque élément de Ω. Par exemple, la classification d'un ensemble de personnes en fonction de leur groupe sanguin.

En principe, la classification supervisée est facilement lisible et interprétable par l'homme, toutefois une ressource humaine (experts dans le domaine) est nécessaire.

III.2.2. La classification non-supervisée

La classification non-supervisée (*clustering en anglais*) d'un ensemble d'objets consiste à découvrir, d'une façon automatique, les groupes d'objets homogènes (partageant un certain nombre de caractéristiques communes) et de les regrouper sous forme de classes (*clusters*). Dans [Elghazel H., 2007], la classification automatique est la tâche qui segmente une population hétérogène en un certain nombre de groupes, plus homogènes, appelés classes.

Dans ce type de classification, les classes ne sont pas connues a priori et le système doit être capable de les définir et d'affecter un objet à la classe la plus similaire. Plus précisément, dans le cas de la classification non-supervisée, seul l'ensemble Ω est connu et il faut donc déterminer le classifieur *f* qui permettra de générer automatiquement l'ensemble *C* des classes. Les classes ainsi obtenues doivent répondre aux critères de

classification. Cette technique semble plus pratique, surtout quand le nombre d'objets à classer est très important, mais généralement c'est une tâche non triviale.

La classification automatique est largement utilisée dans de nombreux domaines d'application. Par exemple, en classification documentaire un grand nombre de travaux ont abordé la classification non-supervisée et plusieurs approches ont été proposées dans ce contexte.

Dans [Berkhin P., 2002] et [Jain A.k. et al., 1999], les méthodes de classification automatiques peuvent être divisées en deux grandes familles : les approches hiérarchiques et les approches par partitionnement.

III.2.3. Les méthodes hiérarchiques

Parmi les méthodes de classification hiérarchique, on distingue : la classification ascendante hiérarchique (CAH) et la classification descendante hiérarchique (CDH). La classification ascendante hiérarchique est basée sur le principe suivant : à chaque itération on crée une nouvelle partition en regroupant les deux éléments les plus proches selon le critère de la classification. Les objets les plus proches sont regroupés dans des groupes aux plus bas niveaux et les objets moins proches sont regroupés aux plus hauts niveaux. Les résultats de cette méthode peuvent être représentés au moyen d'une structure arborescente : une structure connue sous le nom de dendrogramme. Le principe est de regrouper successivement les classes à chaque niveau (figure II.14).

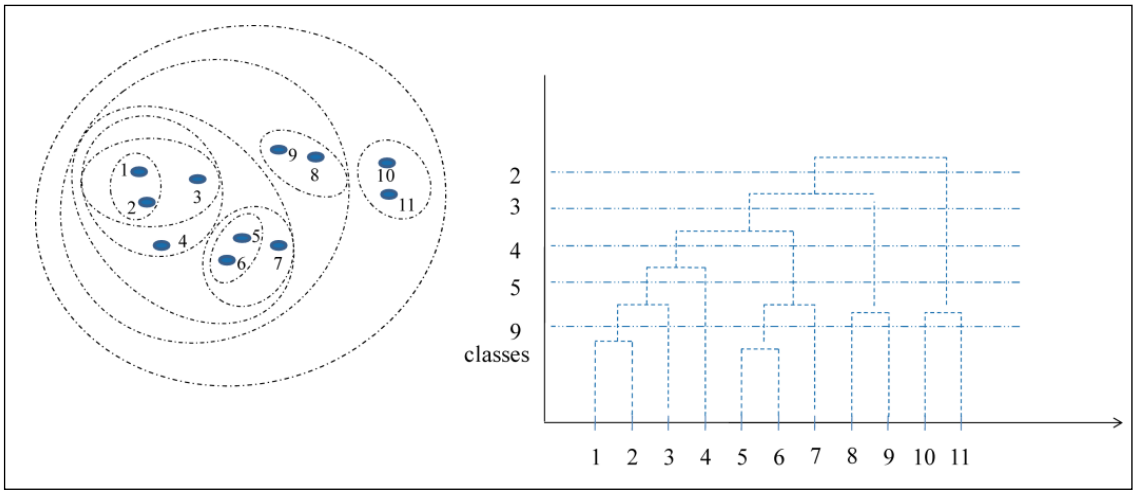


Figure II.14 - Exemple de dendrogramme

A l'inverse, la classification descendante hiérarchique consiste à considérer l'ensemble initial à classer comme étant une classe, ensuite on fait des subdivisions successives de cet ensemble. A chaque itération, une classe est scindée en deux sous-classes de façon à ce que la distance entre les deux classes soit la plus grande possible. Le traitement est arrêté quand les classes obtenues sont à un seul élément.

III.2.4. Les méthodes non hiérarchiques

La classification non-hiérarchique (*ou classification à un niveau*) est une méthode de classification qui aboutit à un regroupement d'objets à un seul niveau. Autrement dit, les

classes ne forment pas de hiérarchies. Les méthodes non hiérarchiques sont généralement itératives, elles consistent à agréger à chaque itération les classes d'objets les plus proches. Ces méthodes peuvent être utilisées efficacement pour traiter une large collection d'objets (par exemple : la méthode classique k-means). On parle alors de *partitionnement* d'une collection d'objets.

Dans la section suivante, nous abordons quelques travaux sur la classification documentaire.

III.3. Classification documentaire : état de l'art

III.3.1. Introduction

Le volume de documents disponibles, de structures et de sources différentes, rend leur classification manuelle de plus en plus difficile, voire impossible. Plusieurs solutions ont été proposées dans la littérature pour appréhender ces documents, réduire l'espace de recherche et optimiser le temps d'accès, à l'information pertinente, qui est un paramètre non négligeable pour évaluer les performances d'un système. Le regroupement de documents sous forme de classes de documents décrivant des informations similaires permet de mieux gérer l'hétérogénéité (taille, type, forme, etc) documentaire. Les classes ont un sens pour celui qui les a crées.

Dans les travaux de [Sylvain L., 2009], les techniques de clustering ont été largement utilisées afin d'optimiser l'accès à l'information en regroupant les résultats aux thématiques similaires. Cela permet de retrouver les documents pertinents à une requête utilisateur dans une même classe [Manning C.D. et al., 2008]. Dans [Djemal K., 2010], la classification est une solution permettant de focaliser les processus d'interrogation et de recherche sur un sous-ensemble de documents similaires. Dans les travaux de [Anderberg M.R., 1973], [Everitt B., 1980], [Roux M., 1986] et [Van C., 1994], la partition d'un ensemble de données sous formes de classes permet de réduire la taille de l'ensemble initial, tout en faisant apparaître une *organisation significative*.

Dans la littérature concernant la classification documentaire, nous pouvons distinguer trois catégories de travaux selon les aspects pris en compte au sein des documents : (1) la structure seule, (2) le contenu seul et (3) le contenu et la structure à la fois.

Généralement, le problème de la classification des objets repose sur deux sous-problèmes : (1) le choix du modèle de représentation le plus approprié avec les objets à classer, (2) le choix de l'opérateur permettant d'évaluer efficacement la proximité de deux objets. Ce choix doit tenir compte de la complexité et de la variabilité des corpus documentaires. Dans [Wisniewski G. et al., 2005], le processus de classification automatique doit être capable de traiter de grandes masses de données.

Dans la littérature, concernant la classification documentaire, les documents peuvent être représentés sous des formes variées induisant ainsi un degré de complexité différent suivant que l'on s'appuie sur des vecteurs, des arbres ou des graphes.

III.3.2. Les travaux qui ont utilisé les vecteurs

Le modèle vectoriel a été utilisé pour représenter les documents aussi bien en recherche d'information qu'en classification des documents, pour ne citer que ces exemples. L'idée de

base consiste à représenter un document par un vecteur dans un espace vectoriel. Le nombre de mots du lexique, présents dans les documents du corpus, constitue la dimension de l'espace vectoriel. Ainsi, un ensemble de caractéristiques (mots clés, termes significatifs, etc d'un document) jugées pertinentes par rapport au contexte, est retenu. Les poids, qui reflètent l'importance de chacune de ces caractéristiques, constituent les composantes du vecteur représentant le document. Par exemple, dans un document multimédia, un vecteur peut représenter l'histogramme des couleurs d'une image composant le document.

En classification des documents, lorsque les documents sont représentés par des vecteurs, on regroupe les vecteurs similaires (en s'appuyant sur une mesure qui permet d'évaluer la ressemblance ou la différence entre deux vecteurs) sous formes de classes de façon à classer les documents proches au sens de la mesure utilisée.

Dans [Doucet A. et al., 2002] un document est représenté sous la forme d'un vecteur dans lequel les composants sont soit des balises, soit des mots du texte, soit une combinaison des deux. L'approche utilisée dans ces travaux permet une classification documentaire tout en tenant compte à la fois du contenu et de la structure des documents classés. Les auteurs de [Yi J. et Sundaresan, 2000], ont utilisé le modèle vectoriel pour représenter les structures documentaires. Les composants d'un vecteur peuvent être soit des mots, soit d'autres vecteurs (une représentation récursive de la structure). Dans [Sigogne A., 2008], les vecteurs ont été utilisés pour représenter les documents Web afin de les regrouper sous forme de classes de documents proches. Les auteurs de [Razo F.D. et al., 2005] utilisent deux vecteurs pour modéliser un arbre représentant un schéma XML ; le vecteur « *st* » pour conserver la position du père de chaque nœud et le vecteur « *lb* » pour enregistrer les étiquettes de l'arbre (figure II.15).

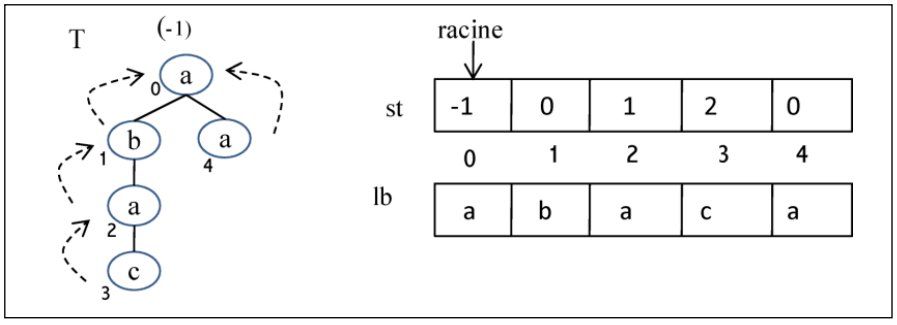


Figure II.15 - Représentation d'un arbre [Razo F.D. et al., 2006]

Les vecteurs sont des structures de données faciles à manipuler. Toutefois, ils ne permettent pas de représenter les objets complexes et structurés, car ils ne considèrent pas les éventuels liens qui peuvent exister entre les composantes structurelles des objets. La pauvreté de leur pouvoir d'expression a été mise en évidence par les travaux utilisant les arbres.

III.3.3. Les travaux qui ont utilisé les arbres

Les arbres sont des structures de données qui permettent de représenter les objets à structure hiérarchique. Ils sont communément utilisés pour représenter l'information d'une façon organisée. Les arbres ont été utilisés pour représenter les schémas XML, les documents ou des parties de documents, etc.

Les travaux de [Termier A. et al., 2002], [Costa G. et al., 2004], [Razo L.F. et al., 2006], [Saleem k., 2008] et [Kutty S. et al., 2008] ont utilisé les *sous-arbres fréquents* (sous-arbres qui apparaissent fréquemment dans les collections d'arbres considérés) pour classer les documents. Dans le processus de classification structurelle de [Termier et al., 2002], Il s'agit d'associer un document à une classe en cherchant dans un corpus d'arbres, un ensemble de sous-arbres fréquents pour mieux caractériser le document. L'algorithme *TreeFinder,* utilisé dans cette approche, permet une classification structurelle des documents en associant à chaque classe une structure représentative. Dans les travaux de [Razo L.F. et al., 2006], il s'agit d'intégrer des schémas XML en vue de construire un schéma médiateur pour l'interrogation de documents XML. Les travaux de [Nierman A. et Jagadish, 2002] et [Francesca F.D. et al. 2003], [Dalamagas T. et al. 2004], sont fondés sur les *résumés d'arbres* pour classer structurellement les documents. Dans ces approches, la similarité peut être basée sur la distance d'édition entre arbres. Le coût de calcul, nécessaire à ces algorithmes, augmente en fonction de la taille des sous-arborescences et du nombre de documents manipulés. Il a été montré dans [Wisniewski et al., 2005] que les algorithmes utilisés dans [Nierman A. et Jagadish, 2002] sont complexes et ne permettent pas de traiter des données de grande taille.

Dans leur approche de classification structurelle des documents, les auteurs de [Dalamagas T. et al., 2006] utilisent les *résumés d'arbres* obtenus par transformations (réduction de la profondeur, élimination des nœuds répétés, etc). Cependant, ces transformations peuvent être à l'origine d'une perte d'information sémantique et contextuelle. Par exemple, la réduction de la profondeur (figure II.16) implique la suppression des composants et des relations entre ces composants. En effet la relation (*A,P*) de T_1 peut ne pas jouer le même rôle que la relation (*A,P*) de T_2.

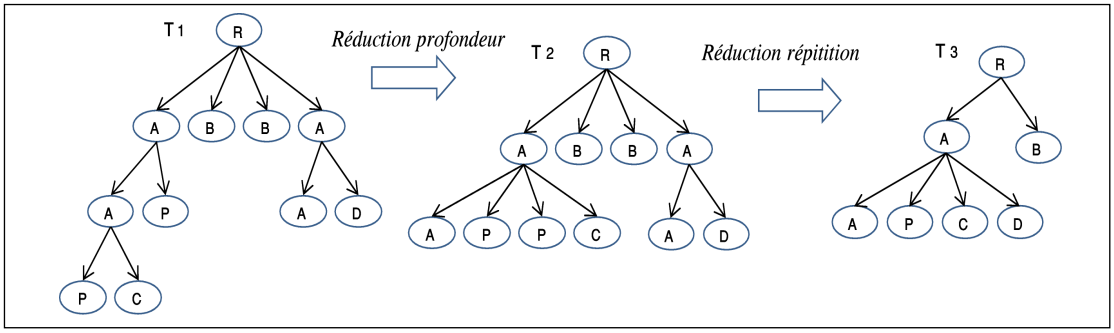


Figure II.16 - Extraction du résumé structurel [Dalamagas T. et al., 2006]

Dans leur approche de classification sémantique des documents XML, [Tagarelli A. et al., 2010] ont proposé un modèle de représentation de données qui exploite la notion de *tuple-arbre* pour identifier les sous-structures sémantiquement cohérentes dans les documents XML.

Dans [Vercoustre A.M. et al., 2006], les documents XML sont représentés sous forme d'une *arborescence*, laquelle est considérée comme un ensemble de chemins. Ainsi, la classification est basée sur le calcul de la fréquence de ces chemins. L'idée, de linéarisation des arbres proposée dans ces travaux est très intéressante. En revanche, les étapes de prétraitement, qui consistent à réduire le nombre de chemins, et celle de filtrage des balises qui peut être à l'origine de perte d'informations ; ce qui peut avoir un impact négatif sur la qualité du classifieur. [Sanz I. et al., 2005] ont défini formellement les notions de *fragment*

et de r*égion (ensemble de fragments)* dans les documents XML représentés à l'aide d'arbres. Pour identifier les fragments et les régions, les auteurs ont proposé l'inclusion approximative entre les *sous-arbres* d'arbre de motif et l'arbre de la collection hétérogène de documents.

Dans l'approche de [Mbarki M., 2008], les structures logiques et sémantiques des documents XML à classer sont représentées sous forme d'arborescences. Dans cette arborescence, chaque nœud représente un *élément* de la structure logique, un *composant* de cet élément, un *attribut* rattaché à un élément générique ou une *métadonnée* qui décrit un composant générique. L'auteur a proposé un modèle qui permet une description logique et une description sémantique des documents multimédia. La description logique traduit la structure logique du document en décrivant les éléments structurels et leurs attributs. La description sémantique permet plusieurs descriptions des métadonnées des composantes multimédia. Ainsi, un même élément de la structure logique peut être décrit de plusieurs manières. Le modèle proposé est composé deux niveaux : un niveau générique contenant des structures génériques et un niveau spécifique représentant la structure d'un document particulier. Chaque structure générique représente un ensemble de structures spécifiques du niveau spécifique. Ce modèle a servi de base pour définir un entrepôt de documents multimédia. L'un des objectifs de cet entrepôt est de permettre un accès facile un l'information multimédia, dans une masse importante de données, afin d'exploiter cette information d'une manière aisée. Plus précisément, le modèle dans ces travaux permet une description riche des documents multimédia numériques. Cette richesse permet l'accès à l'information fine (granules documentaires) et l'analyse des informations de l'entrepôt selon plusieurs dimensions. Il a proposé une mesure qui permet d'évaluer le degré d'inclusion entre deux structures arborescentes :

$$Sim(T,T') = 1 - \frac{\sum_{v_j} Danc(v_j)}{\sum_{v_j} Panc(v_j)} \qquad [12]$$

$Danc(v_j)$: représente la distance, d'alignement, des ancêtres du nœud v_j.
$Panc(v_j)$: représente le poids des ancêtres du nœud v_j.

Dans [Aitelhadj A. et al., 2009], les auteurs ont utilisé les arbres pour représenter les documents en format XML afin de les classer et ont proposé une méthode de classification de structures arborescentes des documents. Pour cela, ils ont introduit une mesure permettant d'évaluer la similarité entre deux arborescences T_1 et T_2 :

$$Similarité(T_1,T_2) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} Sim(e_{1i},e_{2j})}{Max(|T_1|,|T_2|)} \qquad [13]$$

$|T_1|$: la taille de T_1, $Sim(e_{1i},e_{2j})$ représente la similarité des nœuds e_{1i} et e_{2j} appartenant respectivement à T_1 et T_2.

Le calcul de la mesure proposée engendre une explosion combinatoire car il s'agit d'explorer toutes les combinaisons possibles permettant la mise en correspondance des nœuds des deux arbres comparés.

Les approches utilisant les arbres pour représenter les documents multimédia sont confrontées au problème de la limite de représentation de relations multiples entre deux mêmes nœuds d'un document. Pour pallier ce problème, il est donc nécessaire de trouver un modèle de représentation en adéquation avec les documents multimédias.

III.3.4. Les travaux qui ont utilisé les graphes

En tant que structures de données flexibles, les graphes sont utilisés pour représenter les documents multimédia d'une façon plus riche. Dans les travaux de [Harchaoui Z. et Bach, 2007], les graphes ont été utilisés pour représenter les images segmentées afin de les classer. Les auteurs de [Ambauen R. et al. 2003] ont utilisés les graphes étiquetés pour représenter les images, les nœuds représentent les régions (étiquetées par leurs propriétés : couleur, taille,…) de la scène et les arcs représentent les relations binaires entre les régions. L'objectif, de cette représentation est de comparer les images afin de les classer. Pour comparer des images cérébrales à un modèle du cerveau, [Boeres M. et al., 2004] ont utilisé les graphes. Étant donné que l'image médicale est souvent bruitée et sur-segmentée, la comparaison des graphes dans ce contexte est basée sur un appariement un-à-plusieurs (une région de l'image modèle peut correspondre à plusieurs région de l'image médicale).

Les graphes multi-étiquetés ont été utilisés dans [Champin P.A. et Solnon, 2003] pour représenter des objets de conception assistée par ordinateur. Les nœuds du graphe représentent les composants de l'objet et les arêtes du graphe représentent les relations binaires entre ces composants. Des étiquettes sont attribuées à chaque nœud et à chaque arête pour exprimer les différents types de composants et de relations. Il s'agit, dans ce contexte, de chercher à identifier de façon automatique tous les objets déjà conçus similaires à un nouvel objet à concevoir. L'appariement utilisé dans cette application est un appariement multivoque : un composant d'un graphe est mis en correspondance avec plus d'un composant de l'autre graphe (deux composants d'un graphe, par exemple, peuvent jouer le même rôle qu'un autre composant de l'autre graphe). Plus formellement, un appariement multivoque de deux graphes étiquetés $G = (V, r_V, r_E)$ et $G' = (V', r_{V'}, r_{E'})$ et une relation $M \subseteq V \times V'$ composée de tous les couples $(u,v) \in V \times V'$ tels que u est apparié à v.
Où $r_V \subseteq V \times L_V$: est une relation associant une ou plusieurs étiquettes aux nœuds.
Et $r_E \subseteq V \times V \times L_E$: est une relation associant une ou plusieurs étiquettes aux arcs.
r_V et r_E sont les caractéristiques des sommets et des arcs.

Pour chaque appariement M, les auteurs ont défini l'ensemble des étiquettes communes à G et G' comme suit :

$$
\begin{aligned}
G \cap_M G' = & \{ (v,l) \in r_V / \exists (v,v') \in M ; (v',l) \in r_{V'} \} \\
& \cup \{ (v',l) \in r_{V'} / \exists (v,v') \in M ; (v,l) \in r_V \} \\
& \cup \{ (v_i,v_j,l) \in r_E / \exists (v_i,v_i') \in M ; \exists (v_j,v_j') \in M , (v_i',v_j',l) \in r_{E'} \} \\
& \cup \{ (v_i',v_j',l) \in r_{E'} / \exists (v_i,v_i') \in M ; \exists (v_j,v_j') \in M , (v_i,v_j,l) \in r_E \}
\end{aligned}
$$

Les auteurs de ces travaux ont proposé une mesure de similarité qui permet d'évaluer le score de similarité entre deux graphes G et G' par rapport à un appariement M est :

$$Sim_M(G,G') = \frac{f(descr(G) \cap_M descr(G')) - g(splits(M))}{f(descr(G) \cup descr(G'))} \qquad [14]$$

où *f* et *g* sont deux fonctions dépendantes de l'application considérée.
descr(*G*) décrit complètement le graphe *G*.
splits(*M*) : désigne l'ensemble des nœuds, d'un graphe, appariés à plus d'un nœud de l'autre graphe. $descr(G) \cap_M descr(G')$ désigne l'ensemble des caractéristiques communes à *G* et *G'* par rapport à *M*.

Finalement, pour les auteurs, la similarité *Sim*(*G,G'*) de deux graphes *G* et *G'* est définie comme la plus grande similarité possible (celle obtenue par le meilleur appariement) :

$$SimG,G') = \underset{M \subseteq VxV'}{Max}\left[\frac{f(descr(G) \cap_M descr(G')) - g(splits(M))}{f(descr(G) \cup descr(G'))} \right] \quad [15]$$

[Sorlin S. et al., 2006] ont proposé une nouvelle mesure de similarité de graphes représentant des images. Il a été montré que cette mesure est générique du fait qu'elle est paramétrée par deux fonctions, Sim_v et Sim_e, qui dépendent de l'application considérée :

$$Sim(G,G') = \underset{M \subseteq VxV'}{Max}\left[\frac{\sum_{v \in V \cup V'} Sim_v(v, M(v)) + \sum_{(u,v) \in E \cup E'} Sim_e((u,v), M(u,v))}{|V \cup V'| + |E \cup E'|} \right] \quad [16]$$

Les auteurs de [Sorlin S. et al., 2006] ont montré que la mesure proposée par [Champin P.A. et Solnon, 2003] peut être vue comme un cas particulier de leur mesure. En revanche, ils ont noté que le calcul de leur mesure engendre une explosion combinatoire. En reconnaissance de formes, les auteurs de [Suard F. et Rakotomamonjy, 2007] ont utilisé les graphes pour représenter les objets afin de les comparer. Ils ont introduit la notion de saces de chemins pour mesurer la similarité entre deux graphes.

[Djemal K., 2010] a utilisé les graphes pour représenter les documents à structures multiples dans le cadre d'un entrepôt de documents. Pour calculer le score de similarité entre deux graphes *G* et *G'*, représentant des documents XML, il a proposé la mesure suivante :

$$Sim(G,G') = 1 - \frac{\sum_{\varepsilon,\varepsilon'} D_n(\varepsilon,\varepsilon')}{\sum_{\varepsilon} P_f(\varepsilon) + \sum_{\varepsilon'} P_f(\varepsilon')} \quad [17]$$

D_n : fonction d'alignement des relations ε et ε'

P_f : fonction permettant de calculer le poids final d'une relation (arc d'un graphe) et qui est égale au produit de trois fonctions P_{str} : pondération structurelle, P_{Adap} : pondération d'adaptation et P_{Rep} : pondération reflétant la représentativité des relations. Le calcul de la pondération finale, qui dépend du produit de ces trois fonctions, nécessite un temps de réponse qui croit avec la taille des graphes.

III.3.5. Conclusion

Les vecteurs sont fréquemment utilisés pour représenter des objets simples, ils sont utilisés en recherche d'information pour représenter par exemple des documents textuels et sont souples en mémoire et facile à manipuler. Dans [Denoyer L., 2004], la représentation des documents à l'aide des vecteurs présente un bon compromis entre complexité et

performance des systèmes. En revanche, ils ne permettent pas de modéliser des objets complexes. Par exemple, les vecteurs ne permettent pas de représenter les composantes structurelles, et leurs relations, d'un document. En effet un document multimédia peut être décomposé en sous-documents et chaque sous-document peut aussi être décomposable. On doit donc conserver aussi bien les composants du document que les relations entre ces composants. Pour franchir cette limite, la plupart des travaux ont utilisé les arbres pour représenter les documents.

Les arbres sont utilisés, par un grand nombre de travaux, pour représenter les documents sous forme de composants hiérarchiques. Un nœud de l'arbre peut être considéré comme un sous document lui même composé ou non. Ils sont utilisés, par exemple, en recherche d'information pour représenter les documents et les requêtes, en classification de documents. Les travaux manipulant les documents complexes ont montré les limites des arbres. En effet, les arbres ne permettent pas la représentation des relations multiples entre deux composantes d'un même objet.

La représentation des objets complexes nécessite l'utilisation des modèles suffisamment expressifs afin d'intégrer leur complexité. Les graphes sont des outils riches qui permettent la représentation des objets structurés. Par exemple, un document est représenté sous forme d'un ensemble d'entités (les nœuds du graphe) connectées les unes aux autres par des relations (les arêtes) où plusieurs relations peuvent exister entre deux mêmes entités.

Dans la section suivante, nous évoquons le problème de validation des classes générées par une méthode de classification automatique.

III.4. Validation des résultats d'une classification

A l'issue d'un processus de classification automatique, il est judicieux de s'assurer de la validité des classes générées. La validation des résultats d'une classification automatique est un problème qui a attiré l'attention de plusieurs travaux. Nous citons par exemple les travaux de [Genane Y., 2004], [Beney J., 2006] et [Yosr N. et Sinaoui, 2009]. Toutefois, l'évaluation d'un processus de classification n'est pas une tâche triviale. Plusieurs approches ont été proposées dans la littérature pour évaluer l'efficience et la qualité d'un classifieur selon des critères de validation. En revanche, il est difficile d'appréhender toutes les méthodes de classification et de confirmer directement qu'une méthode est plus efficace qu'une autre. Selon [Kanal L., 1993] et [Minsky M., 1991], il n'existe aucune méthode pouvant manifester une supériorité sur les autres pour tous les problèmes et dans toutes les situations. Dans les travaux de [Genane Y., 2004] qui consistent à comparer des partitions, il existe trois types de critères de validation : (1) critère interne qui permet de *mesurer l'écart* entre la structure engendrée et les données. (2) critère externe qui permet de comparer les résultats d'une classification automatique à une information sur la structure des données connue a priori. (3) critère relatif qui permet de comparer deux structures de classification afin de déterminer la meilleure.

III.4.1. Séparation des classes

L'un des problèmes liés à la qualité des classifieurs est la séparation des classes : les classes doivent être distantes le plus possible. La qualité d'un classifieur automatique dépend de sa capacité de générer des classes les plus compactes possibles et les plus distants possibles. Selon [Bisson G., 2000], deux objets lointains représentent des données

qui appartiennent à des groupes différents. En effet, lorsque les classes sont très proches, cela permet d'engendrer d'autres problèmes notamment l'appartenance d'un même objet à deux classes différentes. Dans une telle situation les classes ne sont plus homogènes et donc la classification perd sont intérêt.

III.4.2. Taux de bonne classification

Dans [Beney J., 2006], en classification documentaire, une manière d'évaluer un classifieur est de comptabiliser les documents qui manquent dans chaque classe (l'incomplétude) et ceux qui y sont de trop (le bruit). Dans le cas où le nombre d'objets à classer n'est pas important, on pourra valider les résultats d'une structure de classification en calculant le taux de classification :

$$Taux\ de\ classification = \frac{nombre\ d'objets\ bien\ classés}{nombre\ total\ d'objets} \qquad [18]$$

Et par conséquent, le taux erreurs du classifieur peut être évalué comme suit :

Taux erreur = 1 – *taux de classification*

Le critère du taux de bonne classification est parmi les méthodes d'évaluation les plus connues. Dans les travaux de [El moubarki L., 2009], la plupart des procédures de validation sont exposées comme étant des méthodes de détermination du bon nombre de classes.

III.4.3. Utilisation d'une classification de référence

Une façon de valider les résultats d'une classification est de les comparer avec ceux d'une classification préexistante considérée comme une classification de référence. Selon [Yosr N. et Sinaoui, 2009], pour évaluer les performances d'un algorithme de classification, on compare le résultat obtenu à une référence en utilisant une classification préexistante ou des corpus de référence. Bien que cette méthode semble pratique, il n'est pas toujours facile de trouver une classification de référence ou un corpus de référence des objets à classer.

III.4.4. Rappel et précision

Les notions précision et rappel ont été aussi utilisées pour évaluer un processus de classification. La précision évalue l'homogénéité des classes tandis que le rappel mesure l'exhaustivité des contenus des classes. Un classifieur est plus efficace quand la précision et le rappel sont proches de 1.

Plus explicitement, pour une partition d'un ensemble *E* en *n* classes ; $C=\{c_1,c_2,\ldots c_n\}$, la précision P_i de chaque classe c_i est définie par :

$$P_i = \frac{Nombre\ de\ documents\ correctement\ attribués\ à\ la\ classe\ c_i}{Nombre\ de\ documents\ attribués\ à\ c_i} \qquad [19]$$

La précision de la classification est définie comme suit :

$$P = \frac{\sum_{i=1}^{n} P_i}{n} \quad [20]$$

Le rappel R_i de la classe c_i est défini par :

$$R_i = \frac{Nombre\ de\ documents\ correctement\ attribués\ à\ la\ classe\ c_i}{Nombre\ de\ documents\ \in\ c_i} \quad [21]$$

Enfin le rappel de la classification est :

$$R = \frac{\sum_{i=1}^{n} R_i}{n} \quad [22]$$

Ces méthodes d'évaluation reposent sur des connaissances préalables sur les objets à classer. Il est donc difficile de les appliquer quand il s'agit de classer, d'une manière automatique, un grand nombre d'objets.

III.4.5. Indice de qualité

D'autres travaux ont proposé des indices de qualité permettant d'évaluer l'efficience d'un classifieur. Nous citons par exemple :

- les indices inertiels [Lebart L. et al., 1982] sont utilisés pour évaluer la qualité d'une classification. L'inertie intra-classe, qui permet de mesurer l'homogénéité d'une classe, est la distance entre les individus de celle-ci et son représentant. L'inertie inter-classe est la distance ente les représentants des différentes classes.
- l'indice de Dunn [Dunn J., 1974] qui permet de mesurer la compacité et la séparation des classes. Il est défini par le rapport entre la plus petite distance intra-classe $d_{\min}$ et la plus grande distance inter-classe $d_{\max}$.

$$Dunn = \frac{d_{\min}}{d_{\max}} \quad [23]$$

  L'objectif ici est de maximiser l'indice de Dunn c'est-à-dire de minimiser $d_{\min}$ (augmenter la compacité des classes) et de maximiser $d_{\max}$ (augmenter la séparation des classes).

- et (3) l'indice de Davies-Bouldin [Davies D.L. et Bouldin, 2000] vise à minimiser le rapport des dispersions intra-classe et de la séparation inter-classe :

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{j\in[1,k]\ et\ j\neq i}\left[\frac{\sigma_i + \sigma_j}{d(C_i, C_j)}\right] \quad [24]$$

  Avec k est le nombre de classes, σ_i la moyenne des distances entre les individus de la classe i et le centroïde C_i de cette classe. $d(C_i,C_j)$ est la distance entre les deux centroïdes C_i et C_j. Ainsi, l'indice de Davies-Bouldin est d'autant plus faible que les classes seront plus compactes et distantes les unes des autres.

Globalement, le point commun de ces travaux est d'évaluer le degré d'homogénéité entre les éléments d'une classe et le degré d'hétérogénéité (séparation) entre les classes.
La qualité des classes dépend principalement de la cohérence des objets qui les composent. Il est évident que plus les éléments d'une classe présente des caractéristiques propres communes, plus cette classe est homogène. Par conséquent, plus les distances intra-classe sont faibles, plus l'homogénéité de celle-ci est forte (compacité).

IV. Conclusion

Dans ce chapitre, nous avons présenté l'ensemble des notions de base requises pour la suite de ce mémoire, et notamment les notions utilisées pour les graphes et sur les mesures de similarité ou de distance. L'évaluation de la similarité entre deux graphes est une tâche complexe car elle repose soit directement soit indirectement sur la recherche d'isomorphe de sous-graphe : un problème NP-complet. Malgré la proposition de plusieurs algorithmes, la question de complexité de l'isomorphisme de graphe reste un problème ouvert.

Nous avons également présenté un état de l'art sur la classification automatique des documents numériques. Nous avons constaté que les travaux, qui ont abordé ce thème, peuvent être groupés en trois catégories : des travaux qui ont tenu compte du contenu seul, ceux qui ont tenu compte de la structure seule et ceux qui ont tenu compte de la structure et du contenu des documents à la fois. Ces travaux diffèrent les uns des autres par le modèle utilisé pour représenter les documents (vecteurs, arbres ou graphes), le type de documents (textuels, images, multimédia) utilisés, la démarche utilisée pour les classer et l'objectif de chaque approche. Par conséquent, la comparaison de ces méthodes et la validation de leurs résultats n'est pas une tâche triviale. Il est donc difficile de dire qu'une méthode est meilleure que l'autre ; ainsi malgré le nombre important de méthodes de classification documentaire existantes, le problème reste ouvert.

Nous avons terminé ce chapitre par une présentation (section III.4) de quelques méthodes d'évaluation de la qualité d'une classification automatique.

V. Bibliographie

[Aïtelhadj A. et al. 2009] Aïtelhadj A., Mezghiche M., Souam F., « Classification de structures arborescentes : Cas de documents XML », CORIA, 6th French Information Retrieval Conference, Presqu'île de Giens, France, May 5-7, 2009. Proceeding. LSIS-USTV, ISBN 2-9524747-1-0 page 301-317, 2009.

[Ambauen R. et al., 2003] Ambauen R., Fischer S. and BUNKE H. "Graph Edit Distance with Node Splitting and Merging, and Its Application to Diatom Identification". In *IAPR-TC15,* 2003.

[Anderberg M.R., 1973] Anderberg M.R. "Cluster Analysis for Applications", Academic Press,1973.

[Atteneave F., 1950] Atteneave F., "Dime,sons of Similarity", American Journal of Psychology, Vol. 65 pp 516-556, 1950.

[BÄRECKE T., 2009] BÄRECKE Thomas « Isomorphisme inexact de graphes par optimisation évolutionnaire », Thèse de Doctorat de l'université de Paris VI, 2009.

[Bassok M. et al., 1997] Bassok, M. and Medin, D.L. "Birds of a feather flock together: Similarity judgments with semantically rich stimuli". In journal of Memory & Language, vol. 36, p. 311-336, 1997.

[Beney J., 2006] Beney Jean « *Classification supervisée de documents : théorie et pratique* », Hermes Science, février 2008, 184p.

[Berkhin P., 2002] Berkhin P. "*Survey of clustering data mining techniques"*, Accrue Software, 2002.

[Bisson G., 2000] Bisson G., « La similarité : une notion symbolique/ numérique. Apprentissage symbolique-numérique ». Eds Moulet, Brito, Cepadues Edition, 2000.

[Bunke H. et Shearer k., 1998] Bunke, H. et Shearer K. "A graph distance metric based on the maximal common subgraph". Pattern Recogn. Letters 19 (3-4), 255–259, 1998.

[Bunke H., 1997] Bunke, H. (1997). "On a relation between graph edit distance and maximum common subgraph". Pattern Recogn. Letters 18 (9), 689–697.

[Bunke H., 1999] Bunke, H. 1999, "Error Correcting Graph Matching: On the Influence of the Underlying Cost Function". *IEEE Transactions on Pattern Analysis and Mach Intelligence*, 21(9) : 917–922,1999.ISSN 0162-8828. doi:http: // dx.doi.org /10.1109 / 34.790431.

[Boeres M. et al., 2004] Boeres M., Ribeiro C. & Bloch I. (2004). "A randomized heuristic for scene recognition by graph matching". In *WEA 2004*, p. 100–113.

[Bouveyron C., 2006] Bouveyron C. « Modélisation et classification des données de grande dimension application à l'analyse d'mages » Thèse de Doctorat de Joseph Fourier, 2006

[Champclaux Y., 2009] Champclaux Y., « Un modèle de recherche d'information basé sur les graphes et les similarités structurelles pour l'amélioration du processus de recherche d'information », Thèse de Doctorat de l'Université de Toulouse, 2009.

[Champin P-A., Solnon C., 2003] Champin P-A., Solnon C., "Measuring the similarity of labeled graphs". Dans 5th Int. Conf. On Case-Based Reasoning (ICCBR 2003), Kevin D.

Ashley and Derek G. Bridge ed. Trondheim (NO). pp. 80-95. LNAI 2689. Springer Berlin. 2003.

[Celeux G. et al., 1989] Celeux, G., E. Diday, G. Govaert, Y. Lechevallier, et H. Ralambondrainy (1989). « *Classification Automatique des Données, Environnement statistique et informatique* ». Dunod informatique, Bordas, Paris.

[Conte D. et al., 2004] Conte D., Foggia P., Sansone C. et Vento M., "Thirty Years of Graph Matching in Pattern Recognition". Int. Journal of Pattern Recognition and Artificial Intelligence, 18(3):265–298, 2004.

[Cordella L.P. et al., 2001] Cordella, L.P., Foggia, P., Sansone, C. et Vento, M. 2001. "An Improved Algorithm for Matching Large Graphs", *Proc. 3rd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, pp. 149 -159.

[Costa G. et al., 2004] Costa, G., G. Manco, R. Ortale, et A. Tagarelli. A Tree-Based Approach to Clustering XML Documents by Structure. In *PKDD*, pp. 137–148, 2004.

[Dalamagas T. et al., 2004] Dalamagas T., Cheng T., Winkel K-J. and Sellis T., « Clustering XML Documents Using Structural Summaries », In *EDBT Workshops* p 547-556, 2004.

[Dalamagas T. et al., 2006] Dalamagas T., Cheng T., Winkel K-J, Sellis T.K., "A methodology for clustering XML documents by structure". Information Systems 31(3): 187-228 (2006).

[Davies D.L., 2000] Davies D.L., Bouldin D.W.(2000): "A cluster separation measure". IEEE Trans. Pattern Anal. Machine Intell, 1(4), 224-22.

[Demirci M. F. et al., 2006] Demirci M. F., A. Shokoufandeh, Y. Keselman, L. Bretzner, et S. Dickinson (2006). "Object recognition as many-to-many feature matching", International Journal of Computer Vision 69(2), 203–222.

[Denoyer L., 2004] Denoyer L., « Apprentissage et Inférence statistique dans les bases de documents structurés: Application aux corpus de documents textuels », Thèse de Doctorat Paris 6, 2004.

[Doucet A. et al., 2002] Doucet A. et Ahonen-Myka H., "Naïve Clustering of a large XML Document Collection ", In INEX Workshop, pp 81-87, 2002.

[Dunn J., 1974] Dunn J. (1974): "Well Separated clusters and optimal fuzzy partitions ", Journal of Cybernetics,4, 95-104.

[Elghazel H., 2007] Elghazel Haytham, « Classification et Prévision des Données Hétérogènes : Application aux Trajectoires et Séjours Hospitaliers », Thèse de Doctorat de l'Université Claude Bernard Lyon 1 France, 2007.

[El moubarki L., 2009] El moubarki L. « Décomposition et évaluation des mesures de stabilité d'un partitionnement » Thèse de Doctorat de l'université de Paris-Dauphine, 2009.

[Everitt B., 1980] Everitt B. "Cluster Analysis, Halsted", 1980.

[Francesca F.D., et al., 2003] Francesca F. D., Gordano G., Ortale R., Tagarelli A "Distance-based Clustering of XML Documents", *In Proceedings of the First International Workshop on Mining Graphs,Trees and Sequences*, 2003, pp 75-78.

[Garey M., 1979] Garey M. et Johnson D., "Computers and Intractability: A Guide to the Theory of NP-Completeness", W. H. Freeman, New-York, 1979.

[Gardenfors P., 2000] Gardenfors P. Conceptual spaces : "the geometry of though", Cambridge, MIT Press.

[Genane Y., 2004] Genane Youness « Contributions à une méthodologie de comparaison de partitions » Thèse de Doctorat de l'université de Paris 6 France, 2004.

[Gentner D., 1983] Gentner D., "Structure-mapping: A theoretical framework for analogy", *Cognitive Science*, 7, 155-170. (Reprinted in A. Collins & E. E. Smith (Eds.), *Readings in cognitive science: A perspective from psychology and artificial intelligence*. Palo Alto, CA: Kaufmann), 1983.

[Harchaoui Z. et Bach, 2007] Harchaoui Z. et Bach F., "Image Classification with Segmentation Graph Kernels", Dans CVPR. IEEE, 2007.

[Holyoak K.J. et al., 1989] Holyoak, K. J. et Thagard, P. "Analogical mapping by constraint satisfaction", Cognitive Science, 13, p. 295-355, 1989.

[Hidovi D. et al., 2004] Hidovi D., Pelillo M. (2004), "*Metrics for Attributed Graphs Based on the Maximal SimilarityCommon Subgraph*". International Journal of Pattern Recognition and Artificial, 7/2004.

[Hopcroft J.E. et al., 1974] Hopcroft J. E. et Wong, J. K, "Linear Time Algorithm for Isomorphism of Planar Graphs", *In Proceedings 6th ACM Symposium on Theory of Computing*, pp. 172 -184, 1974.

[Hu H. et al., 2005] Hu H., Hang Y., Han J., et X. Zhou (2005). "Mining coherent dense subgraphs across massive biological network for fonctional discovery", Bioinformatics 1(1), 1–9.

[Hummel J.E., 2000] Hummel, J.E. "*Where view-based theories break down: The role of structure in shape perception and object recognition"*, In E. Dietrich and A. Markman (Eds.). Cognitive Dynamics: Conceptual Change in Humans and Machines. Hillsdale, NJ: Erlbaum, 2000.

[Hummel J.E., 2001] Hummel, J.E. "*Complementary solutions to the binding problem in vision: Implications for shape perception and object recognition" Visual Cognition,* 8, p. 489-517, 2001.

[Jain A.k., 1999] Jain A. k., Murty M. N. and Flynn: P. J., "*Data Clustering: A Review"*, ACM Computing Surveys,31, pp. 264-323, 1999.

[Kalakech M., 2011] Kalakech Mariam, « Sélection semi-supervisée d'attributs : Application à la classification de textures couleur », Thèse de Doctorat, Université de Lille 1, 2011.

[Kanal L., 1993] Kanal L. N., "On pattern, categories, and alternate realities", *Pattern Recognition Letters"*, Vol. 14, pp. 241-255, 1993.

[Klinger S. et Austin J., 2005] Klinger Stefan, Austin Jim : "Chemical similarity searching using a neural graph matcher", ESANN 2005: 479-484.

[Kriegel H.P. et al., 2003] KRIEGEL H.P. ET SCHÖNAUER S., "Similarity Search in Structured Data, Lecture Notes in Computer Science", N° 2737, 2003, pp. 309-319.

[Kutty S., 2008] Kutty S., Tran, T., Nayak, R., et Li, Y. (2008). "Clustering XML Documents Using Closed Frequent Subtrees" : A Structural Similarity Approach‖. Lecture Notes In Computer Science, 183-194.

[Lavoisier, 1789] Lavoisier, dans son ouvrage : "Traité élémentaire de chimie présenté dans un ordre nouveau et d'après les découvertes modernes" ,1789.

[Lebart L. et al., 1982] Lebart L., Maurineau A., Piron M. (1982): « Traitement des données statistiques », Dunod, Paris.

[Levenshtein V., 1966] Levenshtein V., "Binary Codes Capable of Correcting Deletions, Insertions and Reversals", Soviet Physics Doklady, vol. 10, p. 707, 1966.

[Lin D., 1998] Lin D., "An information-theoretic definition of similarity", In Proc. 15th International Conf. on Machine Learning, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.

[Lord P et al., 2003] Lord P, Stevens R, Brass A, Goble C. "Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation". Bioinformatics;19:1275–1283, 2003.

[Manning C.D., 2008] Manning C.D., Raghavan P., Schütze H., "Introduction to Information Retrieval", Cambridge University Press, 2008.

[Mbarki M., 2008] Mbarki Mohamed, « De la modélisation à l'exploitation des documents à structures multiples », Thèse de Doctorat de l'Université Paul Sabatier. - Toulouse, France , 2008.

[Medin D. et al., 1993] Medin, D., Goldstone, R. L. et Gentner, D., "Respect for similarity. *Psychological Review", vol. 2*, p. 254-278, 1993.

[Messmer B.T. et al., 1999] Messmer, B.T., et Bunke, H. 1999. "A Decision Tree Approach to Graph and Subgraph Isomorphism Detection", *Pattern Recognition*, 32, pp. 1979 – 1998.

[Minsky M., 1991] Minsky M., "Logical versus analogical, or symbolic versus connectionist, or neat versus scruffy", *Artificial Intelligence Magazine*, Vol. 12 (2), pp. 34-51, 1991.

[Nierman A. et Jagadish, 2002] Nierman A., Jagadish H. V., "Evaluating Structural Similarity in XML Documents" , *In Proceedings of the Fifth International Workshop on the Web and Databases*, WebDB, 2002, Madison,Wisconsin, USA.

[Razo L.F. et al., 2006] Del Razo Lopez F., Laurent A., Poncelet P., Teisseire M., « Recherche de sous-structures fréquentes pour l'intégration de schémas XML », In Conférence Extraction et Gestion des Connaissances (EGC 2006), Lille, Janvier 2006, volume II, p 487-498.

[Razo F. D. et al., 2005] Razo F. D., A. Laurent, et Teisseire M. « Représentation efficace des arborescences pour la recherche des sous-structures fréquentes », In *Actes de l'atelier Fouille de données complexes, Conférence Extraction et Gestion des Connaissances (EGC 2005)*, pp. 113.120.

[Roux M., 1986] Roux M. « Algorithmes de Classification, Masson »,1986.

[Saleem K., 2008] Saleem Khalid. (2008). "schema matching and integration in large scale snario". Thèse de Doctorat de L'Université Montpellier II, France, 2008.

[Sanz I. et al., 2005] Sanz I., Mesiti M., Guerrini G., and Berlanga Llavori R. (2005). "Approximate Subtree Identification in Heterogeneous XML Documents Collections". Lecture notes in computer science, 2005.

[Shang H., 2010] Shang H., K. Zhu, X. Lin, Y. Zhang, et R. Ichise (2010). "Similarity search on supergraph containment". In Proc. of ICDE, pp. 637–648.

[Sigogne A., 2008] Sigogne Anthony, « Classification incrémentale et regroupement de documents Web », Stage de Master de recherche, Université Paris-Est Marne-la-Vallée, 2008.

[Smeulders A.W. et al., 2000] Smeulders, A. W., Worring, M., Santini, S., Gupta, A., and Jain, R. 2000. "Content-based image retrieval at the end of the early years". IEEE Trans. Pattern Analysis and Machine Intelligence 22, 12, 1349{1380.

[Sokal R.R. et Sneath, 1963] Sokal R.R. and Sneath P.H., "Principles of Numerical Taxonomy". W. H. Fressman and Compagny, 1963.

[Sorlin S. et al., 2006] Sorlin S., Sammoud O., Solnon C., Jolion JM., « Mesurer la similarité de graphes », Dans Extraction de COnnaissance à partir d'Images (ECOI 2006), Atelier de Extraction et Gestion de Connaissances (EGC 2006), Nicole VINCENT et Nicolas LOMENIE ed. ed. Lille. pp. 21-30, 2006.

[Sorlin S. et Solnon C., 2005] Sorlin S., C. Solnon. "Reactive tabu search for measuring graph similarity". In 5th IAPR-TC-15 workshop on Graph-based Representations in Pattern Recognition, Luc Brun, Mario Vento ed. Poitiers. pp. 172-182. Springer-Verlag. 2005.

[STEICHEN O., et al., 2006] STEICHEN O., DANIEL-LE BOZEC C., THIEU M., ZAPLETAL E. et JAULENT M.-C. (2006). « Computation of semantic similarity within an ontology of breast pathology to assist inter-observer consensus», Comput Biol Med, 36(7-8), 768–788.

[Sylvain L., 2009] Sylvain Lamprier, Tassadit Amghar, Bernard Levrat, and Frédéric Saubion. « Clustering en recherche d'information : Concentration vs. Distribution de l'information pertinente ». In CORIA'09 : 6ième Conférence en Recherche d'Information et Applications, pages 115-130, 2009.

[Suard F. et al., 2007] Suard F. A. Rakotomamonjy, **«** Mesure de similarité de graphes par noyau de sacs de chemins », 21ième colloque GRETSI sur le traitement du signal et des images, Troyes, septembre 2007.

[Tagarelli A., 2010] Andrea Tagarelli, Sergio Greco: "Semantic clustering of XML documents". ACM Trans. Inf. Syst. 28(1), 2010.

[Thibaut J.P, 1997] Thibaut, J.-P. « Similarité et catégorisation ». L'année psychologique, 97, p. 701-736, 1997.

[Termier A. et al., 2002] Termier A., Rousset, M. C., et Sebag, M. (2002). "TreeFinder: a First Step towards XML Data Mining", Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02), IEEE Computer Society Washington, DC, USA, 450.

[Tversky A., 1977] Tversky, A. (1977). "Features of Similarity". In *Psychological Review*, Volume 84, pp. 327–352.American Psychological Association Inc.

[Van C., 1994] Van Cutsem, "Classification and Dissimilarity Analysis", Springer Verlag, 1994

[Vercoustre A.M. et al., 2006] Vercoustre, A. M., Fegas, M., Lechevallier, Y., Despeyroux, T., et Rocquencourt, I. (2006). « Classification de documents XML à partir d'une représentation linéaire des arbres de ces documents ». Paris France, 433-444.

[Yi J. et Sundaresan, 2000] Yi J. et N. Sundaresan "A classifier for semi-structured documents", In *KDD 2000 : Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, pp. 340–344, ACM Press 2000.

[Yosr N. et al., 2009] Yosr Naïja, Sinaoui Kaouthar Blibech: "A novel measure for validating clustering results applied to road traffic". KDD Workshop on Knowledge Discovery from Sensor Data 2009: 105-113, (Paris, France, June 28 - 28, 2009).

[Wallis W.D. et al., 2001] Wallis W.D., Shoubridge P., Kraetz M., Ray D.. "Graph distances using graph union". Pattern Recognition Letters 22 (2001).

[Wieczorek S., 2009], Wieczorek S., Gilles Bisson, Sylvaine Rou « Atelier AGS : Apprentissage et Graphes pour les Systèmes complexes », Hammamet, 25 Mai 2009 Dans le cadre de la plate-forme AFIA,CAp'09.

[Wisniewski G., 2005] Wisniewski G., Denoyer L., Gallinari P, « Classification automatique de documents structurés. Application au corpus d'arbres étiquetés de type XML » CORIA 2005: 167-184.

[Witten I.H., 1999] WITTEN I. H., MOFFAT A. et BELL T. C. (1999). "Managing gigabytes : compressing and indexing documents and images". Morgan Kaufmann.

[Zhang N. et al., 2006] Zhang, N., T. Özsu, I. Ilyas, et A. Aboulnaga (2006). "Fix: Feature-based indexing technique for xml documents". In Proc. of VLDB, pp. 259–270.

Deuxième partie : Nos propositions et résultats expérimentaux

Chapitre III - Proposition d'une méthode de comparaison de structures documentaires basée sur les graphes

Résumé du chapitre :

Ce chapitre est consacré à la première partie de notre contribution à savoir la proposition d'une méthode de comparaison de structures. Il est composé de deux parties :
Dans la première partie, nous présentons notre mesure de similarité basée sur l'isomorphisme de sous-graphe. Les définitions et notations nécessaires sont d'abord introduites, ensuite nous avons défini une nouvelle mesure de similarité.
La deuxième partie de ce chapitre est consacrée à la classification des documents multimédias à structures multiples. Nous présentons notre processus de classification en deux étapes : (1) l'extraction de l'organisation et du contenu du document. (2) l'association de la structure de ce document à sa classe structurelle. La recherche de la classe structurelle d'un document passe par un processus de comparaison de graphes représentant les structures documentaires.

Sommaire Chapitre III

I. Introduction

Dans le monde numérique, l'information multimédia est disponible en grande quantité et sous différents formats (texte, image, son, etc). Cependant, toute cette information serait sans intérêt si notre capacité à y accéder efficacement n'augmentait pas elle aussi [Laroum S. et al., 2009]. Il est donc nécessaire de disposer d'outils automatiques permettant d'accéder rapidement à l'information désirée, réduisant ainsi l'effort de l'utilisateur. La classification automatique est une solution qui permet d'organiser et de structurer une large collection de documents afin de réduire l'espace de recherche et par conséquent d'améliorer les performances du processus d'accès à l'information.

La méthode de classification non-supervisée (Cf. chapitre II, section III.2.2 page 67) à laquelle nous nous sommes intéressé dans ce travail s'avèrent utiles pour traiter un nombre important de documents. Le problème, étant donné un corpus de documents, est comment regrouper ces documents sous forme de classes de documents proches ? Ce problème engendre plusieurs questions notamment : Comment représenter ces documents ? Comment évaluer la similarité entre deux documents ?

Pour classer structurellement les documents multimédias à structures multiples, nous restons dans le cadre de *MVDM* (Cf. chapitre I, section VII.2.3 page 40) et nous considérons que la structure documentaire est un facteur suffisamment discriminant pour une classification. En classification documentaire, la comparaison des documents est un problème fondamental. Comparer deux documents nécessite de modéliser ces documents d'une manière formelle et d'utiliser (ou définir) une mesure appropriée pour évaluer la similarité entre ces documents. Le modèle choisi doit être capable d'exprimer le maximum d'informations sur les documents afin de les comparer efficacement. Dans [Sorlin S., 2006], plus la modélisation des documents sera sophistiquée et plus la comparaison de ces documents, sera précise mais difficile. Nous nous intéressons à la représentation des documents multimédias à l'aide des graphes. Comparer deux documents revient donc à comparer les graphes qui les représentent.

Une des problématiques principales de ce travail est de savoir comparer deux documents multi-structurés, et en conséquence de pouvoir *comparer des structures* de documents afin de savoir identifier les ressemblances entre structures et les règles de transformation d'une structure en une autre (évaluation d'un coût de transformation). Plus précisément, le problème se place dans le contexte de comparaison de deux structures documentaires en cherchant leurs sous-structures communes afin de pouvoir identifier un degré d'inclusion ou d'équivalence entre ces structures. La théorie des graphes peut être un appui pour résoudre ce genre de problème. L'isomorphisme de (sous) graphes permet de montrer que l'un des graphes est inclus dans l'autre (ou que les deux graphes sont structurellement identiques) alors que l'intersection entre les graphes permet de montrer les points communs entre ces graphes.

Les modèles géométriques et les modèles basés sur les attributs ne permettent pas la comparaison des objets structurés [Hummel J.E., 2000] et [Hummel J.E., 2001]. Nous avons choisi le modèle à *alignement structurel* (Cf. chapitre II, section II.2 page 60) afin d'évaluer la similarité entre deux structures documentaires. Nous avons défini une mesure de similarité qui permet d'identifier les sous-structures communes à deux documents multimédias, tout en tenant compte des contraintes liées à ce type de documents (relations entre composants, ordre des composants, etc). Ainsi, la *nouvelle mesure* de similarité

proposée est basée sur l'isomorphisme de sous-graphe. Nous montrons que cette mesure est *structurelle* et non une mesure *de surface* comme c'est le cas, par exemple, pour la mesure de Jaccard et celle de Cosinus. Dans [Gentner D. et al., 1989] la similarité dite de *surface* est basée sur les propriétés descriptives des objets alors que la similarité structurelle entre objets est évaluée sur la base des relations entre ces objets. Nous montrons, par conséquent que les mesures standards existantes ne peuvent pas répondre efficacement à notre problématique.

Comme nous l'avons évoqué dans la première section du chapitre II, la recherche d'isomorphisme de sous-graphes est un problème combinatoire. Nous tentons de réduire cette complexité combinatoire. Pour cela, nous considérons un graphe comme un ensemble de chemins. Comparer deux graphes revient donc à comparer les chemins qui les composent.

Dans un premier temps, nous présentons la classification structurelle de documents, au sens où nous l'entendons, dans le cadre du modèle *MVDM*. Ensuite, nous définissons notre mesure de similarité. Dans un deuxième temps, nous présentons une méthodologie de classification structurelle basée sur la mesure proposée.

II. Entrepôt de documents multimédias

II.1. Entrepôt de documents

L'entrepôt de documents permet une organisation structurée, d'une grande masse de données afin de faciliter leur gestion et leur exploitation. Dans [W3 IOS] l'entrepôt de documents est défini comme une architecture intégrée qui offre un accès quasi instantané à une quantité considérable de documents. L'entrepôt de documents permet l'intégration des documents de sources différentes, il constitue une source d'informations pouvant être d'une grande importance aussi bien pour les utilisateurs que pour les décideurs.

Avant de donner une définition de la classification documentaire, nous présentons tout d'abord le modèle *MVDM*.

II.2. Présentation du modèle MVDM

MVDM « *Multi Views Document Model* » est un modèle proposé par [Djemal K., 2010] pour représenter les différentes structures d'un document multimédias à structures multiples. En effet, un document multimédia est multi-structuré par essence. Il peut être considéré comme composé de plusieurs sous-documents (exemple figure III.1), eux mêmes plus ou moins complexes ; car chaque sous-document a une ou plusieurs structures. Ces structures peuvent être de même nature (par exemple figure III.2, deux structures logiques pour le même document) ou de natures différentes (structure physique, logique, temporelle, etc).

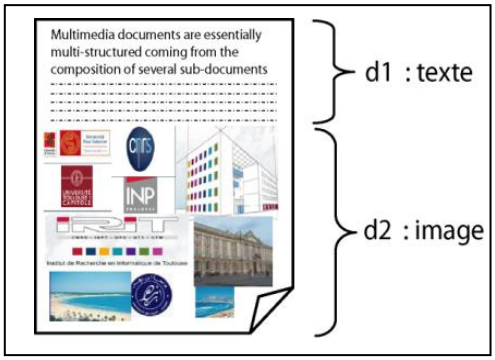


Figure III.1 - Document composé des sous-documents d_1 et d_2

Le modèle *MVDM* a introduit la *notion de vue* : ensemble de nœuds structurants et de relations entre ces nœuds. Un nœud peut être simple ou complexe (par exemple un fragment multimédia image, figure III.1). Dans ce dernier cas, le nœud peut être considéré comme un sous-document, lui-même peut être fragmenté en un ensemble de nœuds et de relations entre ces nœuds. Il peut y avoir plus d'une relation possible entre deux composantes d'un document. Cela permet de matérialiser plusieurs organisations pour ce document. Selon ce modèle, la *notion de structure* documentaire peut être englobée dans une notion plus large qui est celle de vue. Une *vue spécifique* correspond à une organisation particulière ou à un point de vue sur un document. Elle traduit l'une des structures d'un document à structures multiples [Djemal K., 2010]. Par exemple dans la figure III.12, la vue spécifique Vsp_1 est une description par *locuteur* d'un document audio alors que la vue spécifique Vsp_2 est une description par *thème* du même document. Ces deux vues sont agrégées dans *une même structure logique* du document «journal_audio».

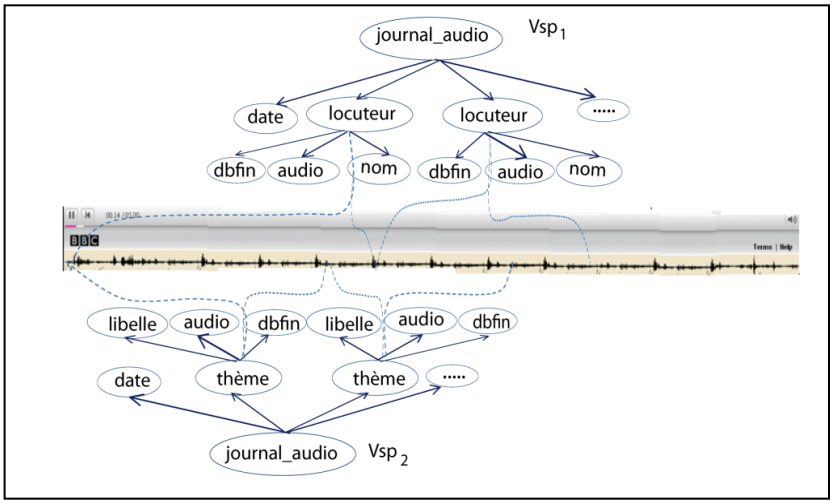


Figure III.2 - Deux descriptions (deux vues) d'un même contenu audio

Nous rappelons que le modèle *MVDM* est composé de deux niveaux de description (figures III.3 et III.4) : un *niveau générique* et un *niveau spécif*ique. Le niveau spécifique où les documents sont décrits au travers de leurs différentes structures traduites par des vues spécifiques (vues et contenus). Le niveau générique décrit les vues génériques représentants des classes. Une *vue générique* représente une collection de vues spécifiques proches d'un point de vue structurel.

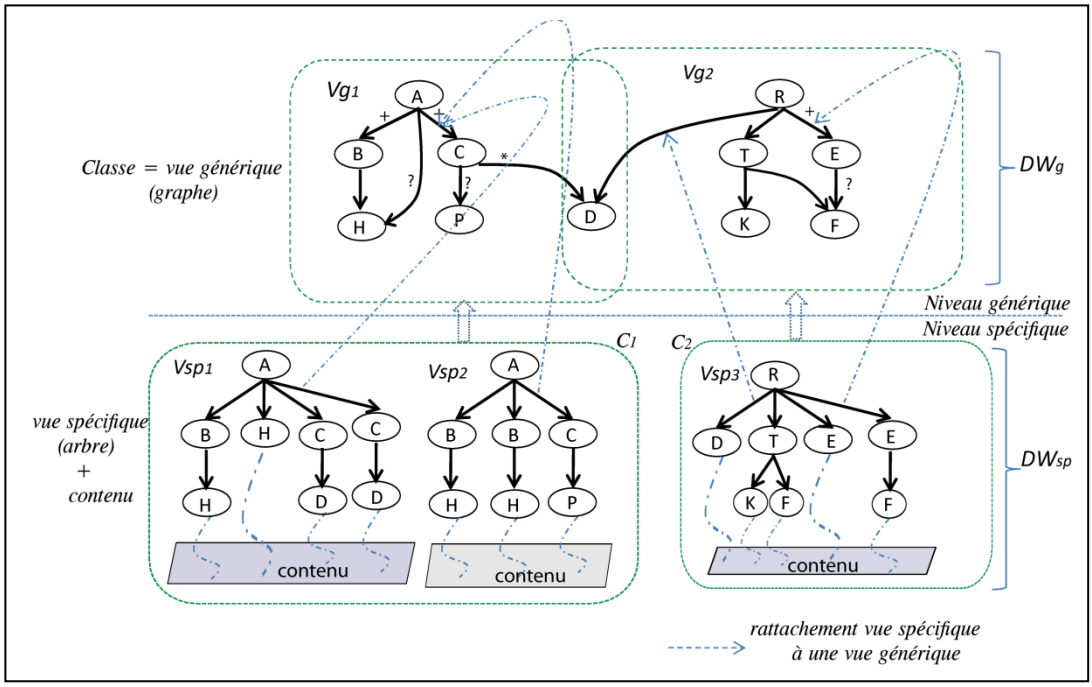


Figure III.3 - Architecture de l'entrepôt documentaire *DW*

Dans l'exemple de la figure III.3, les vues spécifiques Vsp_1 et Vsp_2 sont rattachées à la *vue générique* Vg_1 (représentant de la classe). Chaque fragment de Vsp_1 (resp. de Vsp_2) est rattaché à un fragment similaire de Vg_1. Plus généralement le rattachement spécifique-générique entre les vues ainsi que les agrégations des vues spécifiques-structures spécifiques et des vues génériques-structures génériques sont illustrés par la figure III.4. Dans cet exemple, la vue spécifique Vsp_1 est rattachée à la vue générique Vg_2.

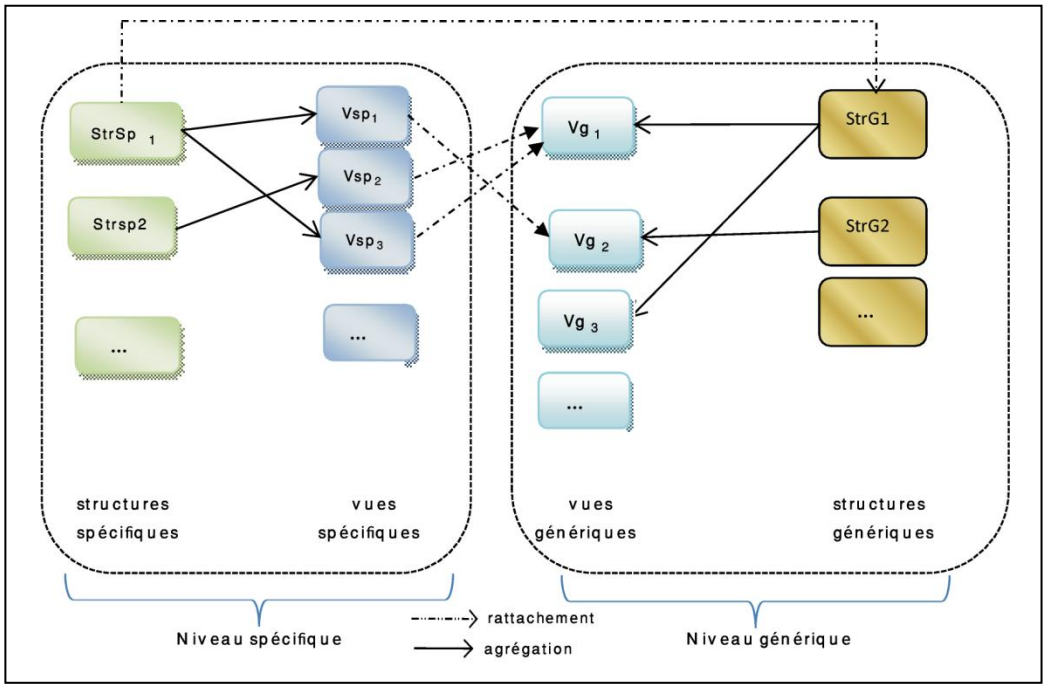


Figure III.4 - Illustration du rattachement (agrégation) vue-structure

Pour décrire les structures documentaires, nous restons dans le cade du modèle *MVDM*. Comme nous l'avons déjà abordé dans le chapitre I, section VII.2.3 page 39, ce modèle permet une description riche des documents à structure multiples et que cette richesse peut être exploitée pour classer ces documents [Portier P.E., 2010].

Nous pouvons écrire : $DW = DW_g \cup DW_{sp}$ (figure III.3) où DW_g représente le niveau générique (les classes) de DW et DW_{sp} représente le niveau spécifique : les caractères spécifiques de chaque document. Les représentants des classes sont des index qui permettent d'interagir avec une large collection de documents de différentes sources qui sont généralement d'une grande hétérogénéité. En effet, l'accès au représentant d'une classe permet l'accès de façon ciblée à la sous-collection de documents de DW_{sp}, représentée par celui-ci. Formellement, nous pouvons définir ainsi :

- $DW_g = \bigcup_{i=1} \{Vg_i\}$ l'ensemble des vues génériques de l'entrepôt, où chaque vue générique Vg_i est composée d'un ensemble de chemins génériques. On peut écrire : $Vg_i = \bigcup_{j=1} \{chm_j\}$ où chaque chemin chm_j générique est un ensemble de relations génériques : $chm_j = \bigcup_{r=1} \{e_r\}$,
- $DW_{sp} = \bigcup_{k=1} \{Vsp_k\}$ l'ensemble des vues spécifiques de l'entrepôt, où chaque vue spécifique Vsp_k est composée d'un ensemble de chemins spécifiques. On peut donc écrire : $Vsp_k = \bigcup_{c=1} \{chm_c\}$ où chaque chemin spécifique chm_c est un ensemble de relations spécifiques : $chm_c = \bigcup_{s=1} \{e_s\}$.

II.3. Définition d'une classification documentaire

Dans le cadre de *MVDM*, le problème de classification documentaire revient au problème de rattachement d'une vue spécifique d'un document donné à la vue générique (du niveau générique : DW_g) la plus proche. Le choix de la vue générique de l'entrepôt la plus proche à la quelle la vue spécifique doit être rattachée repose sur la comparaison de celle-ci à l'ensemble des vues génériques de l'entrepôt.

Les systèmes classiques de comparaison de documents sont basés sur les similarités dites de « *surface* » c'est à dire sur un modèle de similarité basé sur les caractéristiques descriptives des objets sans tenir compte des relations entre ces caractéristiques. Pourtant, les mêmes éléments structurels peuvent ne pas exprimer le même contexte dans deux documents différents. Par exemple, une même image peut ne pas avoir le même rôle dans deux documents différents. Ces systèmes ne tiennent pas compte de l'information implicite véhiculée par la structure documentaire et qu'on ne peut pas ignorer dans un processus de comparaison. Ignorer la structure du document revient à ignorer sa sémantique [Schlieder T. et al., 2002].

Dans nos travaux, nous nous intéressons à la classification *structurelle* (de structures de documents), considérant que la structure est un facteur discriminant intéressant pour la classification. Ainsi, la classification structurelle au sens où nous l'entendons [Idarrou A. et

al., 2010a] permet de créer, dans un entrepôt de documents, des classes appelées *vues génériques* (Cf. Annexe, page 189). Une *vue générique* est une **superposition d'arbres** représentant les structures de documents, elle est enrichie au fur et à mesure de la classification (Cf. section IV.2.3.1 page 123). Cette superposition d'arbres engendre une structure de **graphe enraciné** (exemple figure III.3). Il ne s'agit pas d'un simple résumé, comme c'est le cas des travaux utilisant les résumés d'arbres pour représenter les documents, mais plutôt d'une description riche (sans perte d'informations) représentant un ensemble de structures spécifiques structurellement proches.

Pour classer les documents, il est nécessaire de disposer d'un opérateur approprié qui permet d'évaluer la proximité entre deux documents et de pouvoir par conséquent juger d'une relation de proximité entre ces documents. Généralement, la mesure de similarité est une étape cruciale dans un processus de classification, car les résultats de ce processus dépendent fortement de la mesure utilisée. Lorsque les documents sont représentés en graphes, comparer structurellement deux documents revient donc à comparer les graphes qui les représentent. En théorie des graphes, le problème de comparaison des graphes se ramène au problème de recherche d'un isomorphisme de (sous) graphes. L'isomorphisme de (sous) graphes permet de montrer que deux graphes sont structurellement identiques ou l'un est inclus dans l'autre.

Nous situons nos travaux dans ce cadre de recherche d'isomorphisme de sous-graphes et nous proposons une *nouvelle mesure de similarité structurelle.*

Dans la section suivante, nous introduisons une nouvelle mesure de similarité structurelle basée sur l'isomorphisme de sous-graphes ainsi que la fonction de pondération des graphes sous jacente.

III. Définition d'une mesure de similarité structurelle

III.1. Concepts de base

En préambule de cette section, nous présentons un ensemble de concepts de base et de définitions que nous allons utiliser le long de notre processus de comparaison de structures de documents représentées à l'aide des graphes.

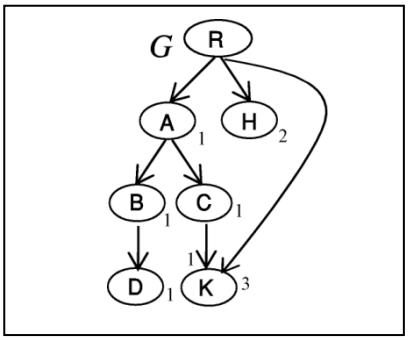


Figure III.5 - Exemple de structure de documents en graphe

La figure III.5 servira de base des exemples pour les définitions que nous introduisons dans le tableau III.1.

Soient $G=(V,E,L_V,f,L_E,g)$ et $G'=(V',E',L_{V'},f',L_{E'},g')$ deux graphes étiquetés, orientés et ordonnés. V (resp. V') ensemble de nœuds de G (resp. de G') et $E \subseteq V \text{x} V$ (resp. $E' \subseteq V' \text{x} V'$) l'ensemble des arcs de G (resp. de G').

où f (resp. f') est la fonction qui permet d'associer à chaque nœud de V (resp. de V') son étiquette dans L_V (resp. $L_{V'}$) et g (resp. g') la fonction qui permet d'associer à chaque arc de E (resp. E') son image dans L_E (resp. $L_{E'}$).

Le graphe $G = (V,E)$ de l'exemple de la figure III.5 est ordonné ; les nœuds fils d'un même nœud sont liés par une relation d'ordre totale de gauche vers la droite.

Notion	définition	exemple
père(*u*)	l'ensemble des nœuds représentant le père (ascendant direct) du nœud *u* : *père*(*u*) = {$x \in V / (x,u) \in E$}	*Dans cet exemple :* *père*(*K*) = {*C*,*R*}
fils(u)	l'ensemble des nœuds dont le père est *u : fils*(*u*) = {$x \in V / (u,x) \in E$ }	*fils*(*A*) = {*B*,*C*}
anc(u)	l'ensemble des ancêtres du nœud *u* (*u* non inclus)	*anc(D)* = {*B*,*A*,*R*}
nœuds adjacents	deux nœuds sont *adjacents* s'ils sont reliés par au moins un arc	*B* et *D* sont adjacents
frère(u)	l'ensemble des nœuds ayant le même nœud père que *u :* *frère*(*u*) = {$x \in V / père(u) = père(x)$}	*frère*(*A*) = {*H*,*K*} (*K* du chemin *R/K*)
ordre(u)	la position de *u* par rapport à ses frères	*ordre(H)*=2, sur le chemin *R/K*, *ordre*(*K*)=3
nœud racine	un nœud *R* d'un graphe *G* est une racine de *G* s'il existe un chemin joignant *R* à chacun des nœuds de *G*	*R* est le nœud racine de *G*
chemin	une suite de nœuds adjacents de *V*, notée $u_1/u_2/u_3/.../u_p$ tel que *:* $\forall k \in [1,p-1]$; $(u_k,u_{k+1}) \in E$	*R/A/B*
chm(u)	Une suite de nœuds adjacents de la racine au nœud *u*	*Chm*(*C*)=*R/A/C*
prof(*u*)	la profondeur d'un nœud *u* dans un chemin est la position de *u* par rapport à la racine *R* (avec *prof*(*R*)=*0*)	*prof*(*K*) = 3 dans le chemin *R/A/C/K*
nœud feuille	un nœud n'ayant pas de fils	*D*, *K* et *H* sont des nœuds feuilles
chemin terminal	un chemin terminal est un chemin (à partir de laracine) qui se termine par un nœud feuille	*R*/*K* est un chemin terminal
un graphe	un graphe *G* peut être considéré comme un ensemble de chemins terminaux	$G = \{chm_1,chm_2,chm_3,chm_4\}$ avec chm_1 = *R/A/B/D*, chm_2 = *R/A/C/K*, chm_3=*R/H et* chm_4 = *R/K*

Tableau III.1 - Définitions de quelques concepts de base

Le graphe *G* (figure III.5) est composé de quatre chemins terminaux : *R/A/B/D*, *R/A/C/K*, *R/H* et *R/K*.

***Théorème* 1**

Un chemin d'un graphe G est un sous-graphe de G.

Dans la figure III.5, le chemin chm_1=*R/A/B/D* peut être écrit : $chm_1 = (V_1,E_1)$ avec V_1 = {*R,A, B, D*} et E_1 = {*(R,A),(A,B),(B,D)*}.

Dans cet exemple, nous avons $V_1 \subset V$ et $E_1 \subset E$. Ce qui implique, d'après la définition de sous-graphe (Cf. chapitre II, section I.2.2 page 56) que chm_1 est un sous graphe de *G*.

Plus généralement, soit *G*=(*V,E*) un graphe orienté composé de *n* chemins terminaux.
Soit chm_i un chemin tels que $chm_i = u_1/u_2/u_3/.../u_p$ où $(u_k,u_{k+1}) \in E$ avec $k \in 1,2, ..., p\text{-}1$. On peut écrire : $chm_i = (V_i,E_i)$ avec $V_i = \{u_1,u_2,\ldots,u_p\}$ *et* $E_i = \{(u_1, u_2), (u_2, u_3),\ldots, (u_{p-1}, u_p)\}$.

Nous avons $V_i \subseteq V$:

Soit $k \in [1,p]$; $u_k \in V_i$ donc u_k est nœud de chm_i d'où $u_k \in V$ (car chm_i est une suite de nœuds adjacents de *V*).

De même $E_i \subseteq E$. En effet, $\forall\ k \in [1,p\text{-}1]$ $(u_k,u_{k+1}) \in E_i$; (u_k,u_{k+1}) est un arc du chemin chm_i donc $(u_k,u_{k+1}) \in E$ (car chm_i est un chemin de *G*).

***Définition* 1 :**

Soient *G et G'* deux graphes orientés, ordonnés et étiquetés et *m* un appariement bidirectionnel entre *V et V'*.

(1) L'appariement *m* préserve les arcs si et seulement si :

$\forall\ (u,v) \in V\text{x}V$; $(u,v) \in E \Rightarrow (m(u), m(v)) \in E'$,

(2) L'appariement *m* préserve les étiquettes des nœuds si et seulement si :

$\forall\ u \in V$; $f(u) = f'(m(u))$,

(3) L'appariement *m* préserve les étiquettes des arcs si et seulement si :

$\forall\ (u,v) \in E$; $g(u,v) = g'(m(u),m(v))$

(4) L'appariement *m* préserve l'ordre des frères *u* et *v* si et seulement si :

(ordre(*u*) <= ordre(*v*) et *père(m(u)) = père*(*m*(*v*)) alors *ordre*(*m*(*u*)) <= *ordre*(*m*(*v*)).

Après avoir présenté les concepts de base, nous présentons une fonction de pondération des graphes.

III.2. Pondération d'un graphe

Dans un processus de comparaison de structures documentaires, nous pensons que les informations apportées par les *relations structurelles* présentent un intérêt incontournable et que deux structures documentaires composées de mêmes éléments, ne signifie pas obligatoirement qu'elles sont similaires. Selon la théorie de la mise en correspondance

développée par [Gentner D., 1983], les bonnes analogies sont celles basées sur les relations entre entités plutôt que sur leurs propriétés descriptives.

Dans cet ordre d'idée, nous avons défini un modèle de pondération de graphe sur lequel sera basée la mesure de similarité. Selon cette mesure, le poids d'un arc doit refléter l'importance, d'un point de vue structurel, de celui-ci dans le graphe. Il doit donc tenir compte des *différentes relations* entre les composantes d'un graphe et de la *position de chacune de ces composantes* : dans un chemin et par rapport aux composantes frères.

La pondération d'un graphe repose alors sur la fonction P_e qui permet d'attribuer des poids aux arcs de ce graphe.

Soit $G = (V,E)$ un graphe orienté, étiqueté et ordonné et (u,v) un arc de G. Le poids $P_e(u,v)$ (Cf. formule [25]) de l'arc (u,v) traduit les aspects hiérarchique et contextuel dans le sens où ce poids dépend de la position de l'arc dans son chemin et de la position de ses extrémités dans un niveau (ordre par rapport à leurs frères). Plus précisément, le poids d'un arc dépend à la fois de la profondeur et de l'ordre de ses extrémités. Plus le nœud v est proche de la racine plus le poids $P_e(u,v)$ de l'arc (u,v) est important.

La fonction P_e de pondération d'un arc proposée est définie par :

$$P_e : E \rightarrow]0,1[$$

$$(u,v) \mapsto P_e(u,v)$$

$$avec\ \ P_e(u,v) = \begin{cases} 1 - \dfrac{\alpha}{k} & \text{si } prof(v)=1 \\ P_e(x,u) - \dfrac{\alpha}{k^{prof(v)}} & \text{sinon ; } x \in père(u) \end{cases} \qquad [25]$$

- $x \in père(u)$: u peut avoir plusieurs nœuds pères (figure III.6, les nœuds R et H).
- $prof(v)$: profondeur de v : position dans un chemin,
- k (*une puissance de* 10) est un paramètre indiquant le nombre maximum de nœuds fils pour chaque nœud (*nombre de fils maximum*<k) dépendant de la nature de la collection de documents traitée (Cf. remarque ci-après),
- α est un paramètre qui dépend du type du nœud v :

$$\alpha = \begin{cases} 1 \text{ si } v \text{ est un attribut ou métadonnée} \\ ordre(v) \text{ sinon} \end{cases}$$

Remarque :

Dans la formule [25], le nombre de chiffres de la partie décimale de $P_e(u,v)$, qui dépend de k, indique le niveau de profondeur de l'extrémité de l'arc (u,v). Par exemple :

Pour $k = 10$:

- $P_e(u,v) = 0.a_1$: le niveau de profondeur de v (dans son chemin) est 1
- $P_e(u,v) = 0.b_1b_2$: le niveau de profondeur de v (dans son chemin) est 2

- $P_e(u,v) = 0.c_1c_2c_3$: le niveau de profondeur de *v* (dans son chemin) est 3
- *etc*

Pour k = 100 :

- $P_e(u,v) = 0.a_1a_2$: le niveau de profondeur de *v* (dans son chemin) est 1
- $P_e(u,v) = 0.b_1b_2\ b_3b_4$: le niveau de profondeur de *v* (dans son chemin) est 2
- $P_e(u,v) = 0.c_1c_2c_3c_4c_5c_6$: le niveau de profondeur de *v* (dans son chemin) est 3
- *etc.*

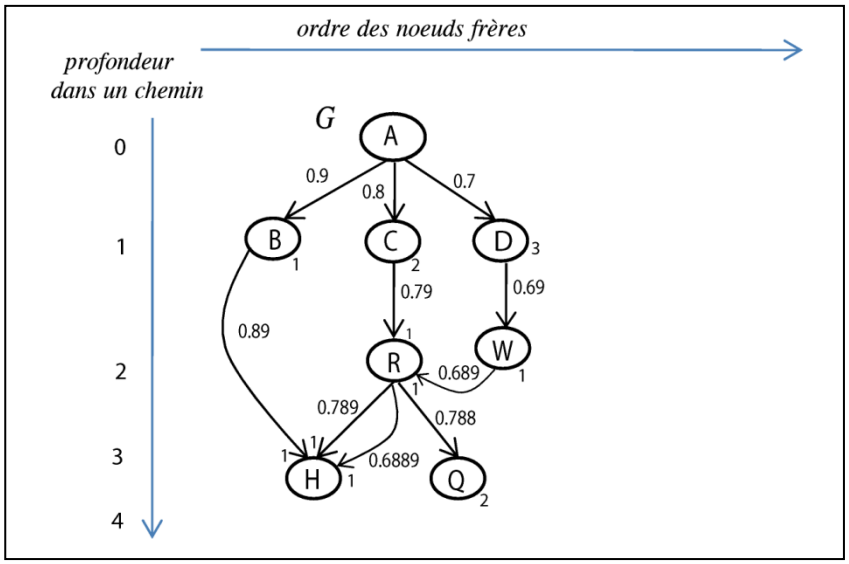


Figure III.6 - Exemple de pondération d'un graphe

Dans l'exemple de la figure III.6 :

a) les chemins terminaux : *A/B/H*, *A/C/R/H*, *A/C/R/Q* et *A/D/W/R/H*, *frère*(*B*) = {*C*,*D*} avec *ordre*(*B*) = 1, *ordre*(*C*) = 2 et *ordre*(*D*) = 3. *frère*(*H*) = {*Q*} avec *ordre*(*H*) = 1 et *ordre*(*Q*) = 2 (sur les chemins *A/C/R/H et A/C/R/Q*, père(*H*) = père(*Q*) = *R*). Les nœuds *R* et *W* n'ont pas de frères : *ordre*(*R*) = *ordre*(*W*) = 1. Sur le chemin *A/B/H* (resp. *A/D/W/R/H*), le nœud *H* n'a pas de frères *ordre*(*H*) = 1,

b) le poids de l'arc (*A*,*C*) est $P_e(A,C) = 1 - ordre(C)/10 = 1 - 2/10 = 0.8$. L'arc (*R*,*H*) possède deux poids différents car il appartient à deux chemins :

- sur le chemin *A/C/R/H* : $P_e(R,H) = P_e(C,R) - 1/1000 = 0.789$ (3 chiffres décimaux : *prof(H) = 3*),
- sur le chemin *A/D/W/R/H :* $P_e(R,H) = P_e(W,R) - 1/10000 = 0.6889$ (4 chiffres décimaux : *prof(H) = 4*),
- L'arc (*R*,*Q*) appartient au chemin *A/C/R/Q*, son poids $P_e(R,Q) = P_e(C,R) - 2/1000 = 0.788$ sachant que *prof(Q) = 3*.

III.3. Isomorphisme de sous-graphes

Nous avons évoqué dans la section I.4 du chapitre II le problème du coût combinatoire engendré par la recherche d'isomorphisme de sous-graphes. Afin de réduire ce coût, nous avons choisi de considérer un graphe comme un ensemble de chemins.

Nous montrons à partir de l'exemple de la figure III.7 que le fait de considérer un graphe comme un ensemble de chemins est une solution qui permet de réduire cette combinatoire. Dans cet exemple, le graphe *G* est composé de 6 nœuds répartis en 3 chemins :

- *book/author/name*,
- *book/author/address/street*
- et *book/author/address/city*.

Le graphe *G'* est composé de 12 nœuds répartis en 7 chemins :

- *book/writer/name*,
- *book/writer/address/number*,
- *book/writer/address/street*,
- *book/writer/address/city*,
- *book/editor/address/number*,
- *book/editor/address/street*,
- *book/editor/address/city*.

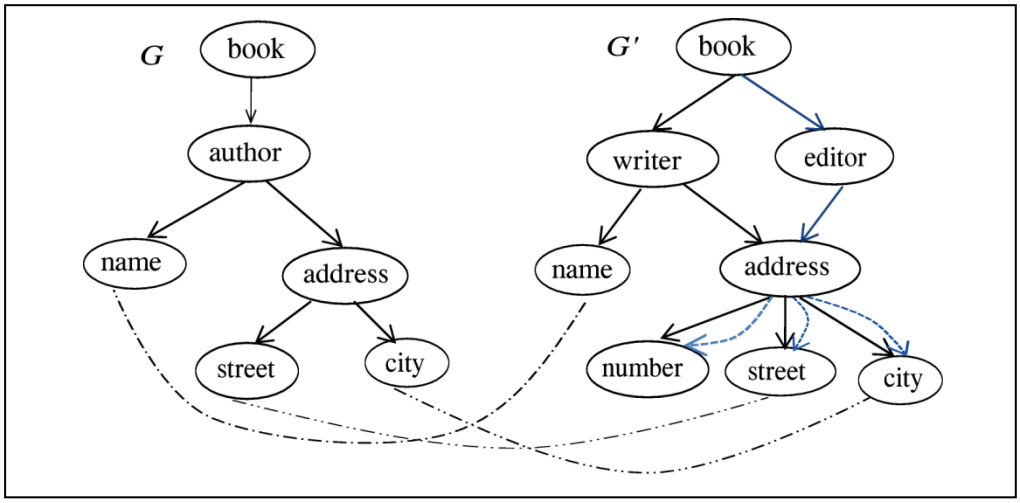


Figure III.7 - Exemple d'appariement de chemins : étiquettes synonymes

Le nombre d'appariements (*injectifs*) **des nœuds** entre *G* et *G'* revient alors à un arrangement du nombre de nœuds de *G* parmi le nombre de nœuds de *G'* soit A^6_{12}=665280. En revanche, le nombre d'appariements (*injectifs*) **des chemins** entre *G* et *G'* est le nombre d'arrangements de 3 parmi 7 chemins soit A^3_7 = 210. La différence entre ces deux valeurs montre qu'il est plus intéressant, d'un point de vue coût de comparaison, de s'appuyer sur les chemins pour la recherche d'isomorphisme de sous-graphes.

La sélection des chemins se fait sur la base des étiquettes des nœuds. Deux nœuds peuvent être considérés comme similaires s'ils ont des étiquettes sémantiquement identiques au sens d'une ressource terminologique standard ou spécifique. Dans l'exemple ci-dessus (figure III.7), les nœuds étiquetés : « *author* » et « *writer*» sont considérés comme étant synonymes et donc pouvant être appariés. Il en sera de même, par exemple, pour « *address* » et « *adresse* », «*city* » et « *ville* », «*editor* » et « *éditeur* », « *nom* » et « *name* » etc. On considérera dans nos travaux que nous disposons de telles ressources terminologiques. Nous n'abordons pas les aspects de traitements linguistiques.

Soient $\{chm_1, chm_2, \ldots, chm_n\}$ l'ensemble des chemins de G et $\{chm'_1, chm'_2, \ldots, chm'_{n'}\}$ l'ensemble des chemins de G', on pose $G = \bigcup_{i\in[1,n]} \{chm_i\}$ (chm_i est un ensemble de relations $chm_i = \bigcup_j \{e_j\}$) et $G' = \bigcup_{i\in[1,n']} \{chm'_i\}$ (chm'_i est un ensemble de relations $chm'_i = \bigcup_h \{e'_h\}$).

Nous avons montré précédemment qu'un chemin *chm* d'un graphe G est un sous-graphe de G. Avant de définir *notre mesure de similarité structurelle*, nous définissons tout d'abord la mesure d_{Inc} qui permet d'évaluer le *degré d'inclusion* (au sens de notre mesure) d'un chemin dans un graphe :

$$d_{Inc}(chm, G') = \min_{k\in[1,n']} \left[\frac{\sum_{e_j \in chm} |P_e(e_j) - w_{j,k}|}{\sum_{e_j \in chm} P_e(e_j)} \right] \qquad [26]$$

$$\text{où } w_{j,k} = \begin{cases} P_e(e'_h) \text{ si } \exists e'_h \in chm'_k \ / \varphi_e(e_j) = e'_h \text{ où } chm'_k \text{ est un chemin de } G' \\ 0 \text{ sinon} \end{cases} \qquad [27]$$

- n' : le nombre de chemins de G',
- φ_e : une fonction d'alignement bidirectionnelle de E (resp. de E') vers E' (resp. E) qui permet d'aligner deux arcs similaires :

$$\varphi_e : E \rightarrow E'$$
$$a \mapsto \varphi_e(a) = a' \text{ ; où l'arc } a' \text{ est similaire à l'arc } a.$$

Nous montrons que $d_{Inc}(chm, G') \in [0,1]$.

En effet $\forall k \in [1,n'] \ \forall e_j \in chm$; $0 <= |P_e(e_j) - w_{j,k}| <= P_e(e_j)$ ($car \ \forall j,k; w_{j,k} >= 0$)

$$\Rightarrow \forall k \in [1,n'] \ \forall e_j \in chm \ ; \ 0 <= \sum_{e_j \in chm} |P_e(e_j) - w_{j,k}| <= \sum_{e_j \in chm} P_e(e_j)$$

$$\Rightarrow \forall k \in [1,n'] \ \forall e_j \in chm \ ; \ 0 <= \frac{\sum_{e_j \in chm} |P_e(e_j) - w_{j,k}|}{\sum_{e_j \in chm} P_e(e_j)} <= 1$$

$\Rightarrow 0 <= d_{Inc}(chm,G') <= 1$

Dans l'exemple de la figure III.7, le graphe *G'* est composé de 7 chemins {*chm'*$_1$,*chm'*$_2$,…,*chm'*$_7$} (figure III.8).

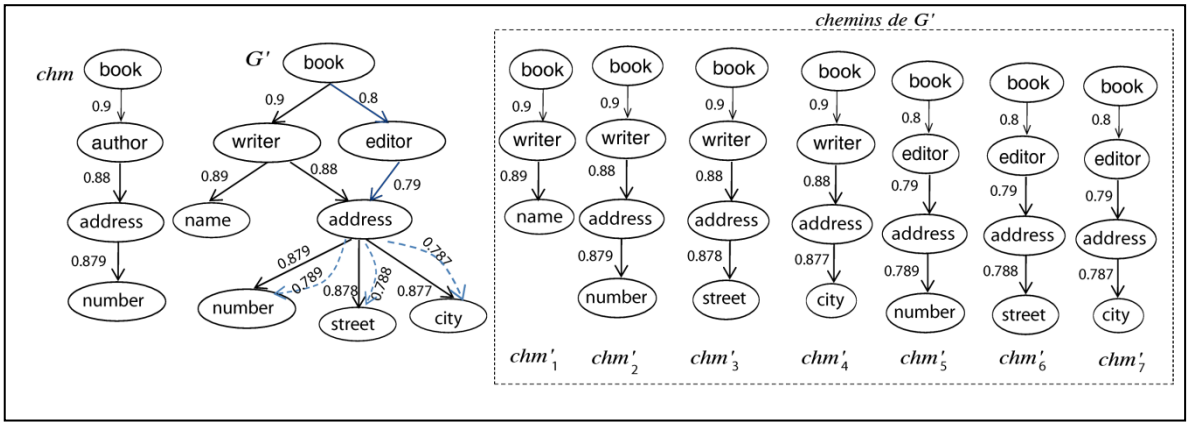


Figure III.8 - Exemple d'inclusion d'un chemin dans un graphe

Nous allons calculer le degré d'inclusion (au sens de notre mesure) du chemin *chm* (*book/author/address/number*) dans *G'* noté $d_{Inc}(chm,G')$:

Valeur de *k*	**Valeur de** $\frac{\sum_{e_j \in chm} \lvert P_e(e_j) - w_{j,k} \rvert}{\sum_{e_j \in chm} P_e(e_j)}$
1	0.66
2	0.00
3	0.33
4	0.33
5	0.70
6	1.00
7	1.00

Tableau III.2 - Calcul de degré d'inclusion d'un chemin dans un graphe

d'où $d_{Inc}(chm,G') = \min_{k\in[1,7]} \left[\frac{\sum_{e_j \in chm} \lvert P_e(e_j) - w_{j,k} \rvert}{\sum_{e_j \in chm} P_e(e_j)} \right] = 0$

***Théorème* 2 :**

$d_{Inc}(chm,G')$ = 0 implique que *chm* est isomorphe à un sous-graphe de *G'* (*chm* est similaire à un chemin de *G' : chm* inclus dans *G'*).

En effet :

$$d_{Inc}(chm,G') = 0 \Leftrightarrow \sum_{e_j \in chm} | P_e(e_j) - w_{j,k} | = 0$$

$$\Rightarrow \forall\ e_j \in chm\ ;\ \exists\ k \in [1,n']\ ;\ P_e(e_j) = w_{j,k}$$

$$\Rightarrow \forall\ e_j \in chm\ ;\ \exists\ e'_h \in chm'_k / \varphi_e(e_j) = e'_h \text{ (formule [27])}$$

$$\Rightarrow chm \text{ est inclus dans } G' \text{ (} chm \text{ inclus dans } chm'_k \text{ un chemin de } G').$$

Dans l'exemple de la figure III.8, *chm* est isomorphe à chm_2 un sous-graphe de *G'*.

***Théorème* 3 :**

$d_{Inc}(chm,G') = 1$ si et seulement si $chm \cap G' = \phi$.

En effet :

$$d_{Inc}(chm,G') = 1 \Leftrightarrow \min_{k \in [1,n']} \left[\frac{\sum_{e_j \in chm} | P_e(e_j) - w_{j,k} |}{\sum_{e_j \in chm} P_e(e_j)} \right] = 1$$

$$\Leftrightarrow \forall\ k \in [1,n']\ ;\ w_{j,k} = 0$$

$$\Leftrightarrow chm \cap G' = \phi$$

***Théorème* 4** :

$0 <= d_{Inc}(chm,G') < 1$ implique que $chm \cap G' \neq \phi$.

$d_{Inc}(chm,G') < 1$ implique qu'il existe un chemin de *G'* dont l'intersection avec *chm* est non vide (le cas où $d_{Inc}(chm,G')=0$ est déjà traité théorème 2).

En effet, soit *chm* un chemin de *G* tel que $chm = \{e_1,e_2,...,e_c\}$ et *G'* un graphe composé de *n'* chemins.

Supposons que $d_{Inc}(chm,G') < 1$

$$d_{Inc}(chm,G') < 1 \Rightarrow \frac{\sum_{e_j \in chm} | P_e(e_j) - w_{j,k} |}{\sum_{e_j \in chm} P_e(e_j)} < 1$$

$$\Rightarrow \sum_{e_j \in chm} | P_e(e_j) - w_{j,k} | < \sum_{e_j \in chm} P_e(e_j)$$

$$\Rightarrow \sum_{e_j \in chm} | P_e(e_j) - w_{j,k} | - \sum_{e_j \in chm} P_e(e_j) < 0$$

$$\Rightarrow \sum_{e_j \in chm} (| P_e(e_j) - w_{j,k} | - P_e(e_j)) < 0$$

$\Rightarrow \exists\ j \in [1,c]$ et $\exists\ k \in [1,n']$ tel $w_{j,k} \neq 0$ (car $P_e(e_j) > 0$ et $w_{j,k} >= 0$)

$\Rightarrow \exists\ j \in [1,c]$ et $\exists\ e'_h \in chm'_k / \varphi_e(e_j) = e'_h$ (formule [27])

C'est-à-dire, qu'il existe au moins un arc de *chm* similaire à un arc de *G'*. Donc $chm \cap G' \neq \phi$.

Exemple :

Dans la figure III.9, nous avons :

- (a) $d_{Inc}(chm_1, G') = 1$; $(chm_1 \cap G' = \phi)$,
- (b) $d_{Inc}(chm_1, G') = 0.33$; $(chm_1 \cap G' \neq \phi)$,
- (c) $d_{Inc}(chm_1, G') = 0$; $(chm_1 \subset G')$.

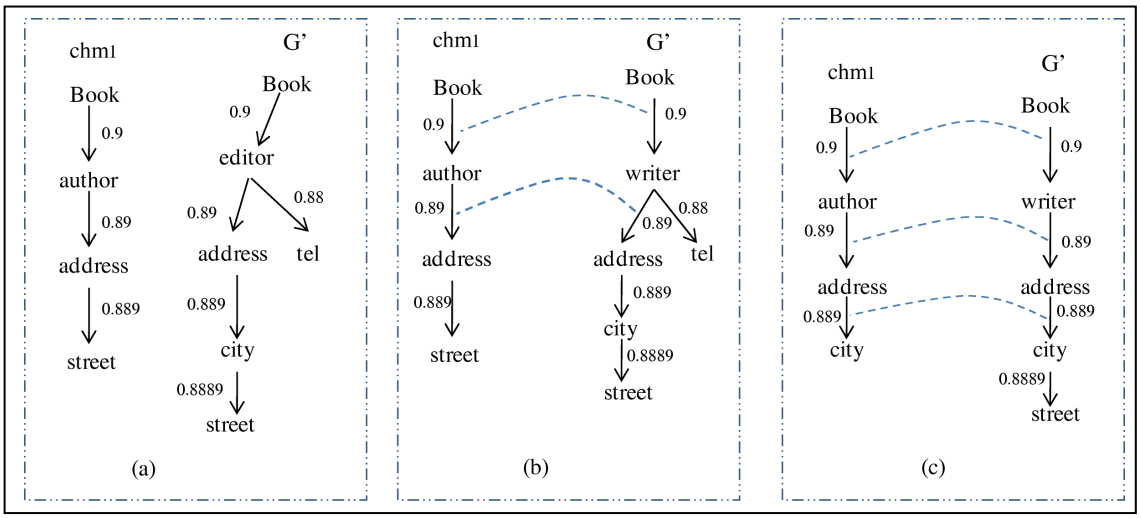


Figure III.9 - Exemple de comparaison de chemins

***Théorème* 5** :

$\sum_{i \in [1,n]} d_{Inc}(chm_i, G') = 0$ implique que G (composé de n chemins) est isomorphe à un sous-graphe de G'.

En effet : $\sum_{i \in [1,n]} d_{Inc}(chm_i, G') = 0$

$\Rightarrow \forall i \in [1,n]$; $d_{Inc}(chm_i, G') = 0$

$\Rightarrow \forall i \in [1,n]$; chm_i est isomorphe à un sous-graphe (chemin) de G' (théorème 2)

$\Rightarrow G$ est isomorphe à un sous-graphe de G'.

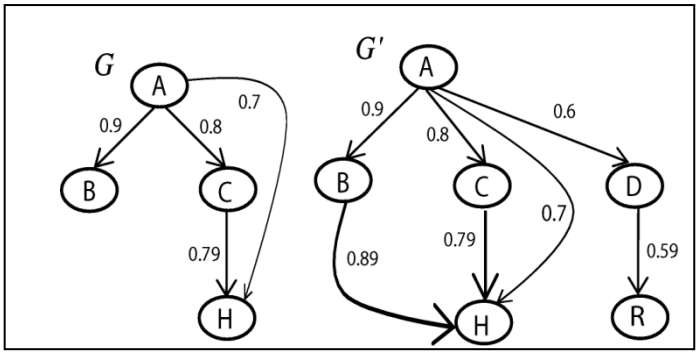


Figure III.10 - Exemple d'inclusion de graphes : isomorphisme de sous-graphe

Dans l'exemple de la figure III.10, le graphe *G* est composé de trois chemins terminaux : $chm_1 = A/B$, $chm_2 = A/C/H$, et $chm_3 = A/H$.

Le graphe *G*' est composé de quatre chemins terminaux :

$chm'_1 = A/B/H$, $chm'_2 = A/C/H$, $chm'_3 = A/H$ et $chm'_4 = A/D/R$.

$$\sum_{i\in[1,3]} d_{Inc}(chm_i, G') = d_{Inc}(chm_1, G') + d_{Inc}(chm_2, G') + d_{Inc}(chm_3, G')$$

$$= \left[\frac{(0.9-0.9)}{0.9} + \frac{(0.79-0.79)+(0.8-0.8)}{1.59} + \frac{(0.7-0.7)}{0.7} \right] = 0$$

Le graphe *G* est isomorphe à un sous graphe de *G'* ($G \subset G'$).

III.4. Une nouvelle mesure de similarité structurelle

Les systèmes classiques de comparaison renvoient une valeur indiquant que les deux objets comparés sont similaires ou non. Cependant, dans la plupart des applications, il est intéressant d'avoir plus de précision sur la proximité des objets comparés. Nous nous sommes intéressés à la catégorie des systèmes qui permettent d'évaluer la proximité entre deux objets à partir d'une valeur continue permettant de quantifier la ressemblance et la différence entre ces deux objets.

Nous proposons une *nouvelle mesure de similarité structurelle* basée sur l'isomorphisme de sous-graphes. Cette mesure reflète la structure des graphes comparés dans le sens où l'on compare les chemins des graphes en tenant compte à la fois de la position des nœuds, de l'ordre des nœuds frères et des liens entre ces nœuds. L'isomorphisme de sous-graphes induits permet de démontrer qu'un graphe est inclus dans un autre, alors que l'isomorphisme de sous-graphes partiels permet de déterminer l'intersection entre deux graphes. Dans notre contexte, nous considérons que l**a position des nœuds et les relations entre ces nœuds** sont deux paramètres incontournables dans un processus de comparaison structurelle de documents multimédias. Ainsi, la fonction de pondération que nous avons proposée (Cf. section III.2 page 100), et sur laquelle repose notre mesure de similarité, tient compte de ces deux paramètres (exemple figure III.11).

Pour évaluer la *similarité structurelle Sim(G,G')* entre deux graphes, nous avons défini la mesure suivante :

$$Sim(G,G') = 1 - Dist(G,G') \quad [28]$$

$$avec \quad Dist(G,G') = \frac{d_{GG'} + d_{G'G}}{2} \quad [29]$$

$$et\ d_{GG'} = \frac{1}{n} \sum_{i \in [1,n]} d_{Inc}(chm_i, G') \quad et\ d_{G'G} = \frac{1}{n'} \sum_{j \in [1,n']} d_{Inc}(chm'_j, G) \quad [30]$$

où $d_{GG'}$ (resp. $d_{G'G}$) : est la distance d'alignement entre *G* et *G'* (resp. entre *G'* et *G*) *et n* et *n'* (non nuls) sont respectivement le nombre de chemins de *G* et *G'*.

La division par *n* (resp. *n'*) permet de normaliser, entre 0 et 1, la valeur de $d_{GG'}$ (resp. de $d_{G'G}$).

Nous montrons que $Dist(G,G') \in [0,1]$.

Nous avons vu précédemment que, $\forall chm$; $d_{Inc}(chm,G') \in [0,1]$.

$$\Rightarrow \forall\ i \in [1,n]\ ;\ 0 <= d_{Inc}(chm_i, G') <= 1$$

$$\Rightarrow\ 0 <= \sum_{i \in [1,n]} d_{Inc}(chm_i, G') <= n$$

$$\Rightarrow 0 <= \frac{1}{n} \sum_{i \in [1,n]} d_{Inc}(chm_i, G') <= 1$$

$$\Rightarrow 0 <= d_{GG'} <= 1$$

De même on pourra démontrer que $0 <= d_{G'G} <= 1$.

Or, $0 <= d_{GG'} <= 1$ et $0 <= d_{G'G} <= 1 \Rightarrow 0 <= \frac{d_{GG'} + d_{G'G}}{2} <= 1$

c'est à dire $0 <= Dist(G,G') <= 1$.

La similarité prend sa valeur maximale lorsque les deux graphes sont structurellement identiques.

***Corollaire* 1** :

$d_{GG'} = 0$ implique *G* est isomorphe à un sous-graphe de *G'* (théorème 5).

Plus généralement, $d_{GG'}$ (resp. $d_{G'G}$) permet d'évaluer le degré d'inclusion de *G* dans *G'* (resp. de *G'* dans *G*) et de déduire que :

- un graphe est totalement inclus dans un autre (isomorphisme de sous-graphes induits),
- deux graphes partagent des caractéristiques communes ($0 <= d_{GG'} < 1$ ou $0 <= d_{G'G} < 1$),

- deux graphes sont disjoints ($d_{GG'} = 1$ ou $d_{G'G} = 1$).

Corollaire 2 :

$d_{GG'} = 0$ et $d_{G'G} = 0 \Rightarrow G$ et G' sont isomorphes.

En effet, d'après théorème 5, $d_{GG'} = 0$ implique que G est inclus dans G' et que $d_{G'G} = 0$ implique que G' est inclus dans G. Par conséquent G et G' sont isomorphes.

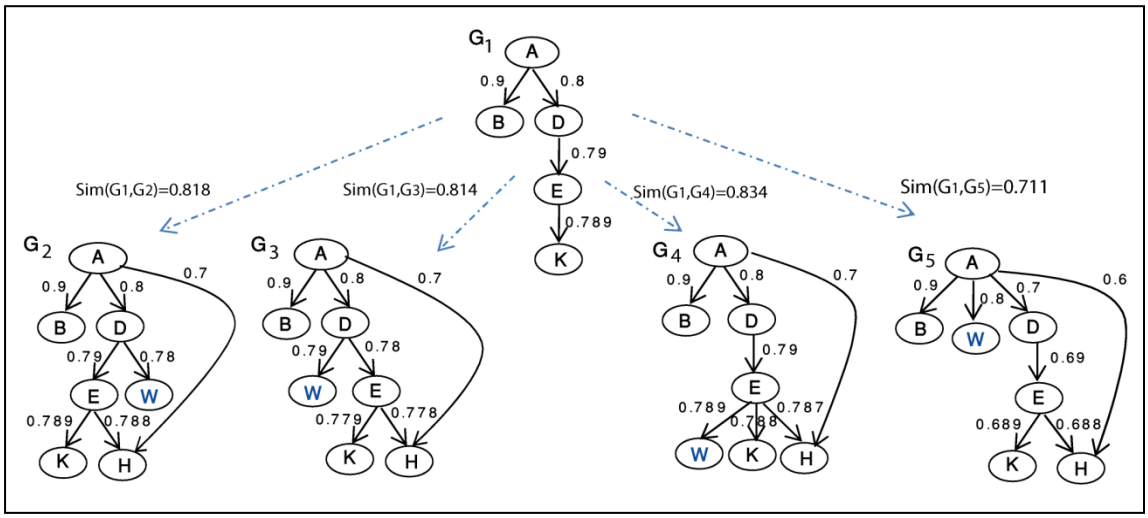


Figure III.11 - Exemple de similarité de graphes

Dans l'exemple de la figure III.11, G_1 est isomorphe à un sous-graphe de G_2 ($G_1 \subseteq G_2$).

En effet, $d_{G_1G_2} = \left[\frac{(0.9-0.9)}{0.9} + \frac{(0.789-0.789)+(0.79-0.79)+(0.8-0.8)}{2.379} \right] / 2 = 0$

alors que, $d_{G1G3} = 0.004$, $d_{G1G4} = 0.0002$, $d_{G1G5} = 0.063$.

Toujours dans le même exemple, la similarité structurelle entre chaque paire de graphes :

$Sim(G_1,G_2) = 0.818$, $Sim(G_1,G_3) = 0.814$, $Sim(G_1,G_4) = 0.834$ et $Sim(G_1,G_5) = 0.711$.

Dans cet exemple, la différence entre $Sim(G_1,G_2)$, $Sim(G_1,G_3)$, $Sim(G_1,G_4)$ et $Sim(G_1,G_5)$ s'explique par le fait que la mesure proposée tient compte de la répartition des éléments structurels dans les graphes comparés. Nous remarquons une différence, qui devient importante dans le cas de $Sim(G_1,G_5)$, entre les valeurs des similarités en raison de différences de positionnement de certains nœuds, et en particulier du nœud *W* (ordre différent ou niveau différent). Ceci montre que la mesure de similarité proposée tient bien compte des deux paramètres profondeur et ordre, en pénalisant les différences de profondeur.

III.5. Comparaison de notre mesure avec d'autres mesures

Pour comparer la mesure proposée avec d'autres mesures existantes, nous avons choisi deux types de mesures : un type basé sur les caractéristiques descriptives sans tenir compte des relations entre les composants des objets et un type basé sur l'alignement structurel.

- *mesures basées sur les caractéristiques descriptives* (*dites de « surface »*)

Pour comparer les structures représentées par les graphes de la figure III.11, nous avons utilisé les mesures de Jaccard et celle de Cosinus (Cf. chapitre II, section II.2.2 page 62) afin de comparer notre mesure avec un modèle de mesures dites de « *surface* ».

$$Jaccard(G_1,G_2) = Jaccard(G_1,G_3) = Jaccard(G_1,G_4) = Jaccard(G_1,G_5) = \frac{|G_1 \cap G_2|}{|G_1 \cup G_2|} = \frac{5}{8} = 0.625$$

$$Cos(G_1,G_2) = Cos(G_1,G_3) = Cos(G_1,G_4) = Cos(G_1,G_5) = \frac{|G_1 \cap G_2|}{\sqrt{|G_1| * |G_2|}} = \frac{5}{6.32} = 0.791$$

Jaccard	**Cosinus**	**Notre mesure**
$Jaccard(G_1,G_2)$ = 0.63	$Cos(G_1,G_2)$ = 0.79	$Sim(G_1,G_2)$ = 0.82
$Jaccard(G_1,G_3)$ = 0.63	$Cos(G_1,G_3)$ = 0.79	$Sim(G_1,G_3)$ = 0.81
$Jaccard(G_1,G_4)$ = 0.63	$Cos(G_1,G_4)$ = 0.79	$Sim(G_1,G_4)$ = 0.83
$Jaccard(G_1,G_5)$ = 0.63	$Cos(G_1,G_5)$ = 0.79	$Sim(G_1,G_5)$ = 0.71

Tableau III.3 - Comparaison de notre mesure avec celles de Jaccard et de Cosinus

Les graphes G_2, G_3, G_4, *et* G_5 se composent des mêmes nœuds. En revanche, ces nœuds n'ont pas la même répartition sur les 5 graphes. Plus précisément, ces graphes ne sont pas identiques d'un point de vu structurel. Nous constatons que les valeurs présentées par les lignes des colonnes 1 et 2 du tableau III.3 sont les mêmes. Elles ne dépendent pas de l'organisation des nœuds des graphes G_1, G_2, G_3, G_4, *et* G_5. Contrairement aux mesures de *Jaccard* et de *Cosinus*, notre mesure est structurelle et non une mesure de *surface*, elle tient compte de l'aspect structurel des objets appariés, ce qui se traduit clairement au travers des valeurs de la troisième colonne du tableau III.3. En effet, notre mesure est basée sur une fonction de pondération (Cf. section III.2 page 100) tenant compte des aspects hiérarchique et contextuel.

- *mesures basées sur le modèle à alignement structurel*

Pour comparer notre mesure avec une mesure du modèle à alignement structurel, nous avons choisi deux exemples de mesures :

A) La mesure de [Mbarki, 2008] (Cf. chapitre II, section III.3.3 page 72). Dans ses travaux, l'auteur a utilisé les arbres pour représenter les structures logiques et sémantiques des documents. La mesure proposée est basée sur une distance d'alignement des nœuds des arbres à comparer. Nous considérons un arbre comme un cas particulier de graphe. Dans ce contexte, nous avons choisi de comparer la mesure proposée par [Mbarki, 2008] avec notre mesure. Pour cela, nous avons pris cinq structures représentées à l'aide des graphes de la figure III.12. Les nœuds sont pondérés en utilisant la fonction de pondération proposée par le même auteur.

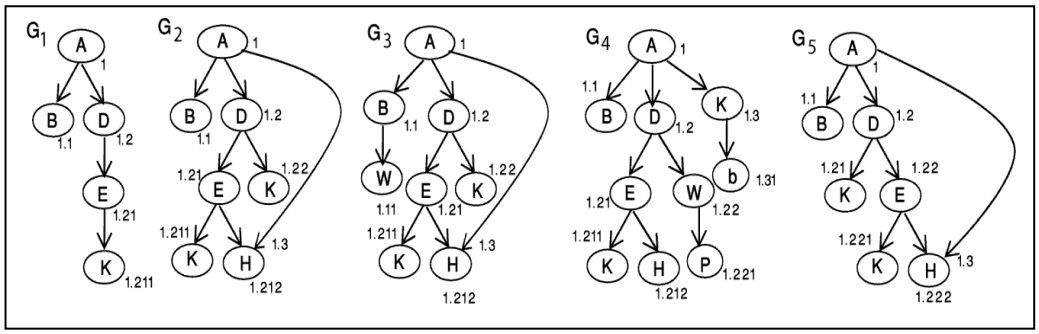


Figure III.12 - Pondération des nœuds : approche de [Mbarki M., 2008]

Mesure de Mbarki	**Notre mesure**
$Sim(G_1,G_2) = 1.00$	$Sim(G_1,G_2) = 0.82$
$Sim(G_1,G_3) = 1.00$	$Sim(G_1,G_3) = 0.77$
$Sim(G_1,G_4) = 1.00$	$Sim(G_1,G_4) = 0.80$
$Sim(G_1,G_5) = 0.99$	$Sim(G_1,G_5) = 0.81$

Tableau III.4 - Comparaison de notre mesure avec celle de [Mbarki M., 2008]

Comme nous l'avons déjà évoqué précédemment (Cf. chapitre, II section III.3.3 page 72), la mesure de [Mbarki M., 2008] permet de calculer le degré d'inclusion d'un graphe dans un autre. Elle ne permet pas d'évaluer la similarité entre deux graphes. La pondération proposée par l'auteur privilégie le nœud fils (de niveau n+1) sur le nœud père (de niveau n). Nous remarquons que selon cette mesure, la similarité entre un graphe G et un graphe G' qui le contient est égale à 1 et cela quelque soit G' (exemple, les lignes 1, 2 et 3 de la colonne 1 du tableau III.4). Par conséquent, il est difficile d'interpréter le résultat $Sim(G,G') = 1$. Contrairement à la mesure de [Mbarki M., 2008], notre mesure pénalise aussi les composantes non appariés du graphe G'. Plus précisément, notre mesure permet d'évaluer l'inclusion dans les deux sens (la recherche d'isomorphisme de sous-graphes dans les deux sens : de G vers G' et de G' vers G) entre les deux graphes comparés.

B) La mesure proposée par [Djemal K., 2010] (Cf. chapitre II, section III.3.4 page 74), est basée sur une distance d'alignement des arcs. Nous allons reprendre un exemple de structures représentées par les graphes de la figure III.13, présenté dans les travaux de [Djemal K., 2010]. Ensuite, nous allons comparer notre mesure et celle proposée dans [Djemal K., 2010] en calculant la similarité entre d'une part S_1 et S_2 et d'autre part S_1 et S_3 (tableau III.5).

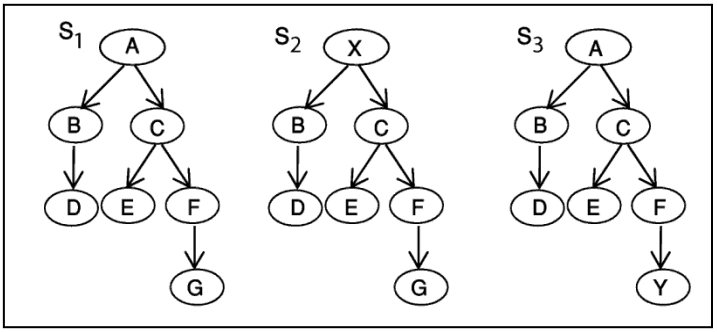


Figure III.13 - Exemple de comparaison de structures [Djemal K., 2010]

Mesure de Djemal	Notre mesure
$Sim(S_1,S_2) = 0.60$	$Sim(S_1,S_2) = 0.56$
$Sim(S_1,S_3) = 0.82$	$Sim(S_1,S_3) = 0.90$

Tableau III.5 - Comparaison de notre mesure avec celle de [Djemal K., 2010]

Dans notre approche, une composante (nœud ou arc) d'un graphe de niveau de profondeur *n* est plus importante que celle de niveau *n*+1. Ce qui explique la différence entre les valeurs de la première ligne du tableau III.5. La deuxième ligne du tableau montre que notre mesure est plus efficace, les graphes sont similaires à un arc près.

Contrairement aux mesures basées sur les caractéristiques descriptives, nous constatons qu'avec les mesures basées sur le modèle à alignement structurel intégrant les structures on obtient des résultats plus fins.

Pour calculer la similarité (ou distance) entre deux graphes, nous avons introduit :

« *Dist_Graph()* » : une fonction qui permet de calculer la distance entre deux graphes (Cf. Annexe page 184).

IV. Classification structurelle des documents multimédias

IV.1. Introduction

Dans la littérature, concernant la classification documentaire, nous pouvons distinguer trois catégories de travaux selon les aspects pris en compte au sein des documents : (1) la structure seule, (2) le contenu seul et (3) à la fois contenu et structure.

Notre processus de classification structurelle vise à organiser une large collection de documents hétérogènes sous forme de sous-collections de documents homogènes d'un point de vue structurel, tout en gardant les caractères spécifiques de chaque document (contenu et structures spécifiques). Les individus d'une classe sont représentés par leurs vues spécifiques et les classes sont matérialisées par les vues génériques. La construction de ces vues génériques repose sur un processus de classification des vues spécifiques par appariement de graphes. La question à laquelle nous tentons de répondre dans cette section

est comment construire les vues génériques (classes) dans le cadre d'un entrepôt documentaire ?

Nous proposons une classification structurelle, de documents, où les classes ne sont pas connues a priori, elles sont calculées automatiquement lors de l'intégration des documents, à partir d'une mesure de similarité que nous avons définie à la section III.4 (page 108) de ce chapitre. La similarité entre un document et une *vue générique* (représentant d'une classe) repose sur un processus de comparaison de la structure du document avec chacune des *vues génériques* de l'entrepôt. Ceci pose des problèmes de temps de calculs et d'efficience de nos algorithmes. Ainsi, nous avons proposé un *sous-processus de filtrage* (*présélection*) des *vues génériques* de l'entrepôt susceptibles d'être similaires à la structure du nouveau document. Cette restriction de l'espace de comparaison permet d'optimiser le temps de réponses de nos algorithmes tout en conservant la qualité des classes générées.

Une des questions liées à la validation des résultats d'une classification non-supervisée (Cf. chapitre II, section III.4 page 75) est comment évaluer la qualité des classes générées ?

La qualité d'un classifieur dépend de sa capacité de générer des classes les plus homogènes possibles et les plus distantes possibles. Dans ce contexte, nous avons proposé d'utiliser une *distance minimale inter-classe* comme paramètre fixé a priori par l'utilisateur. Augmenter la distance entre classes permet de diminuer le bruit et augmenter la précision de la classification. En revanche quand les classes sont très proches, un autre problème surgit : l'appartenance d'un même document à deux classes différentes. Dans une telle situation les classes ne sont plus homogènes et donc la classification perd sont intérêt.

Dans ce qui suit, nous présentons notre processus de *classification structurelle* des documents multimédias à structures multiples.

IV.2. Processus de classification

Pour pouvoir classer structurellement un document, il est nécessaire de comparer sa structure avec le représentant de chacune des classes existantes, et de déterminer la ou les classes les plus similaires. L'idée de base de notre processus d'intégration d'un nouveau document multimédia dans l'entrepôt documentaire est : (1) d'extraire la vue spécifique *Vsp* de ce document et (2) de calculer la similarité entre *Vsp* et chacune des vues génériques *Vg* de DW_g. Au final deux cas peuvent être envisagés :

- rattacher *Vsp* à la classe *Vg* la plus similaire (rattacher les composants spécifiques : nœuds et relations du document aux composants génériques similaires de *Vg*, exemple figure III.3 page 98) tout en gardant les *caractères spécifiques* (contenu et structures spécifiques) du document à intégrer,
- créer une nouvelle classe s'il n'existe pas de vue générique similaire à *Vsp*.

***Définition* 2** :

Deux vues *X* et *Y* sont similaires si et seulement si : *Sim*(*X*,*Y*) >= *Seuil_Sim*

où Sim() est la fonction de similarité définie (Cf. section III.4, page 108) et *Seuil_Sim* $\in[0,1]$ est un paramètre fixé a priori.

Prenons les exemples de graphes de la figure III.11.

Si le seuil de similarité *Seuil_Sim* est fixé à 0.80 alors les graphes G_1 et G_2 sont similaires car $Sim(G_1,G_2) = 0.818$. En revanche, les graphes G_1 et G_5 ne sont pas similaires car $Sim(G_1,G_5) = 0.711$ ($Sim(G_1,G_5) < Seuil\_Sim$).

La question primordiale est donc : comment déterminer la vue générique (la classe) de l'entrepôt la plus similaire à la vue du document à intégrer ?

Afin de déterminer la vue générique à laquelle la vue spécifique doit être rattachée, nous proposons une démarche de classification structurelle hiérarchique ascendante non-supervisée, basée sur la mesure de similarité structurelle définie dans la section III.4. La première structure documentaire insérée dans la base sert de premier représentant (première classe). Les classes sont ensuite construites par agrégation des documents proches d'un point de vue structurel (Cf. Annexe, page 189).

Le schéma suivant (figure III.14) représente la démarche de notre processus de classification structurelle des documents, de l'extraction de la vue spécifique du document en entrée à sa comparaison avec les vues génériques de l'entrepôt documentaire :

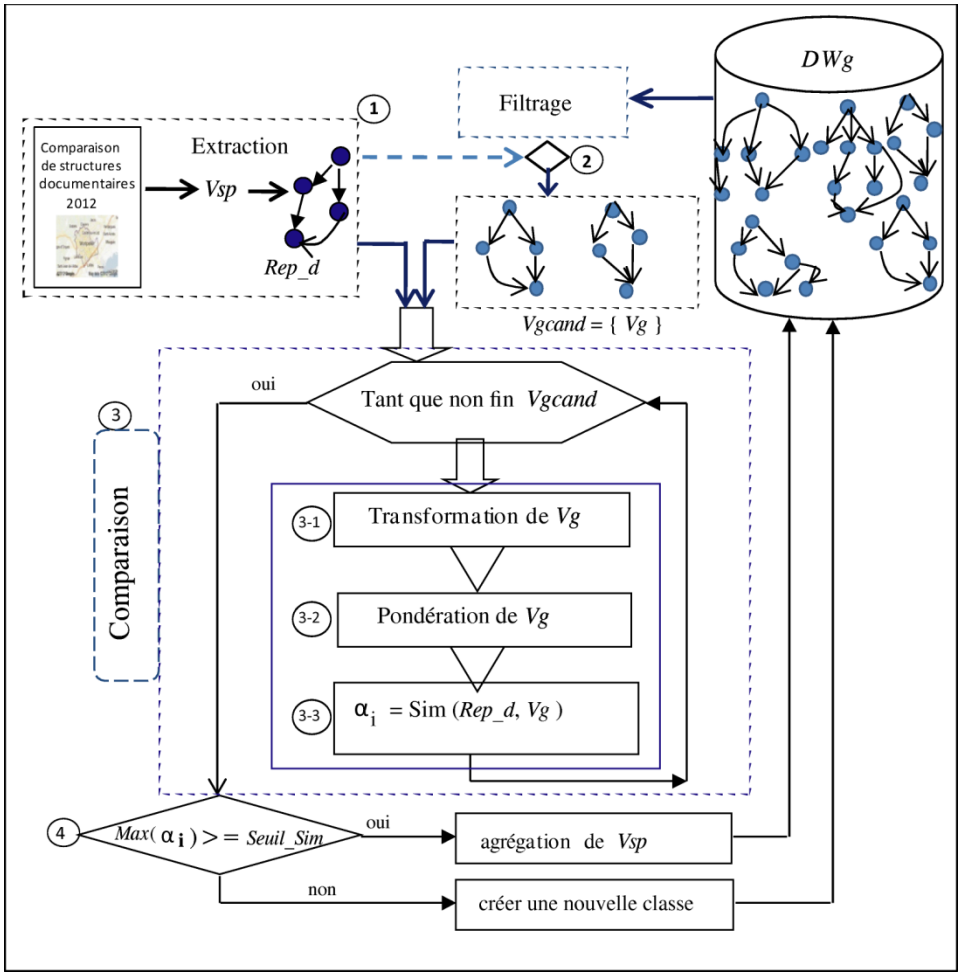


Figure III.14 - Processus de classification structurelle

Où *Vgcand* est l'ensemble des vues génériques, de l'entrepôt documentaire, susceptibles à être similaires avec la vue spécifique *Vsp* en entrée. *Rep_d* est le représentant de la vue spécifique du document à intégrer. Celui-ci sera utilisé dans le processus de classification.

IV.2.1. Extraction de la structure d'un document

Cette étape consiste à extraire la structure et le contenu du document à intégrer. Après l'extraction de l'organisation spécifique du document, un représentant du document, qui *matérialise la représentation générique* de la vue spécifique, est généré à partir de celle-ci (figure III.15 et figure III.18). Le représentant *Rep_d* du document à intégrer, ainsi obtenu, sera utilisé par la suite dans le processus de comparaison. Il est matérialisé par un ensemble d'instances des classes *« VueGen »*, *« NoeudGen »* et *« RelationGen »* du modèle *MVDM* (Cf. chapitre I, section VII.2.3 page 40).

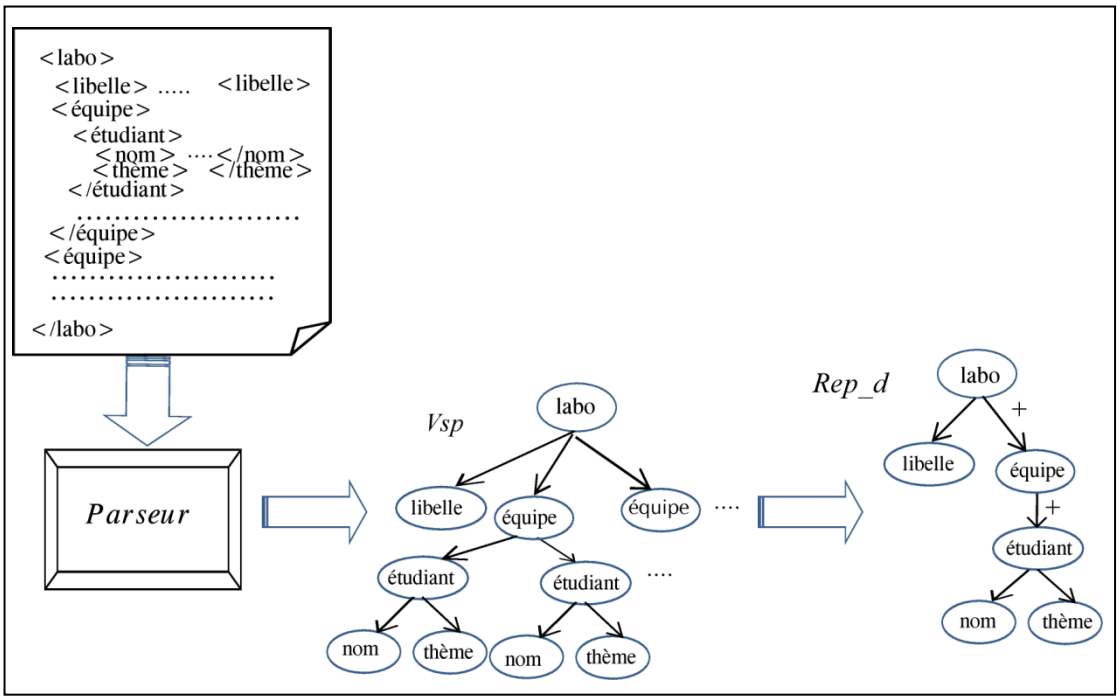


Figure III.15 - Chaîne d'extraction de la vue spécifique au représentant d'un document

Dans *Rep_d*, une information (cardinalité) est ajoutée à chaque arc. Elle indique l'occurrence de cet arc dans le graphe :

- " " : cardinalité par défaut qui signifie un et un seul,
- "+" : signifie un ou plus (1..n),
- " ?" : signifie zéro ou un (0..1),
- "*" : signifie zéro ou plusieurs (0..n).

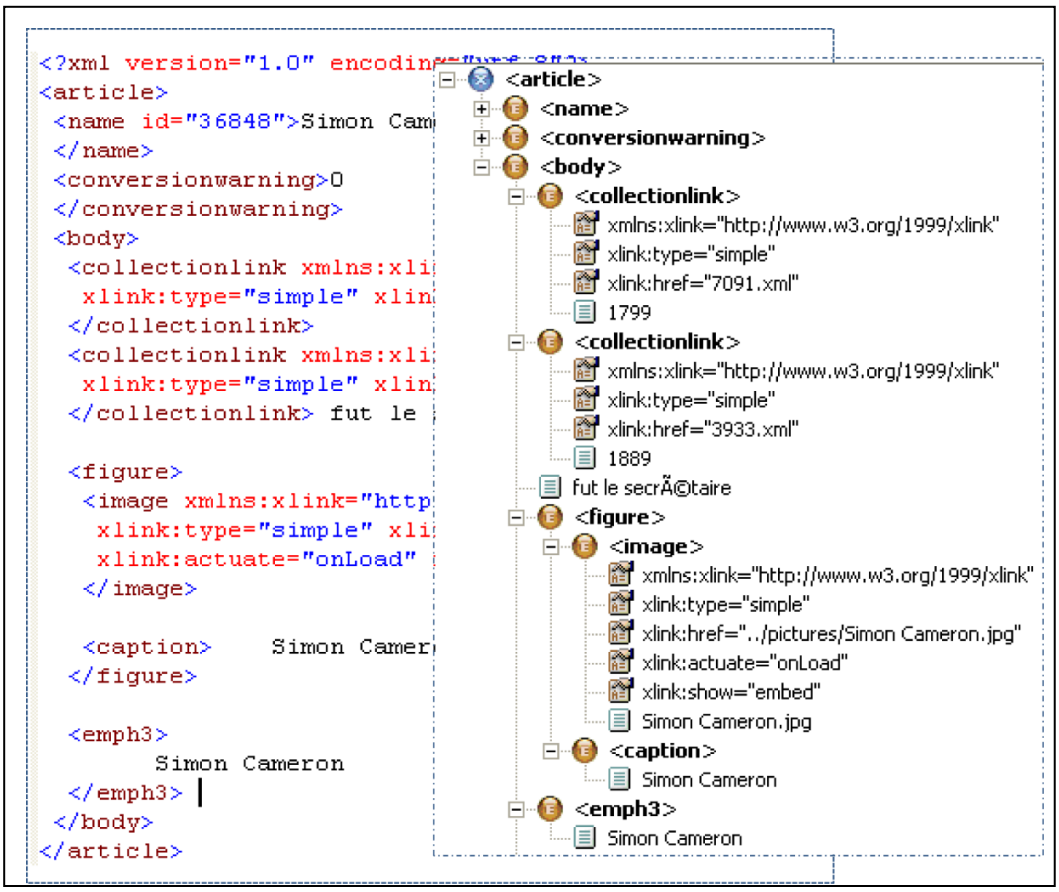


Figure III.16 - Exemple d'un document XML éditée par « Bonfire Studio »

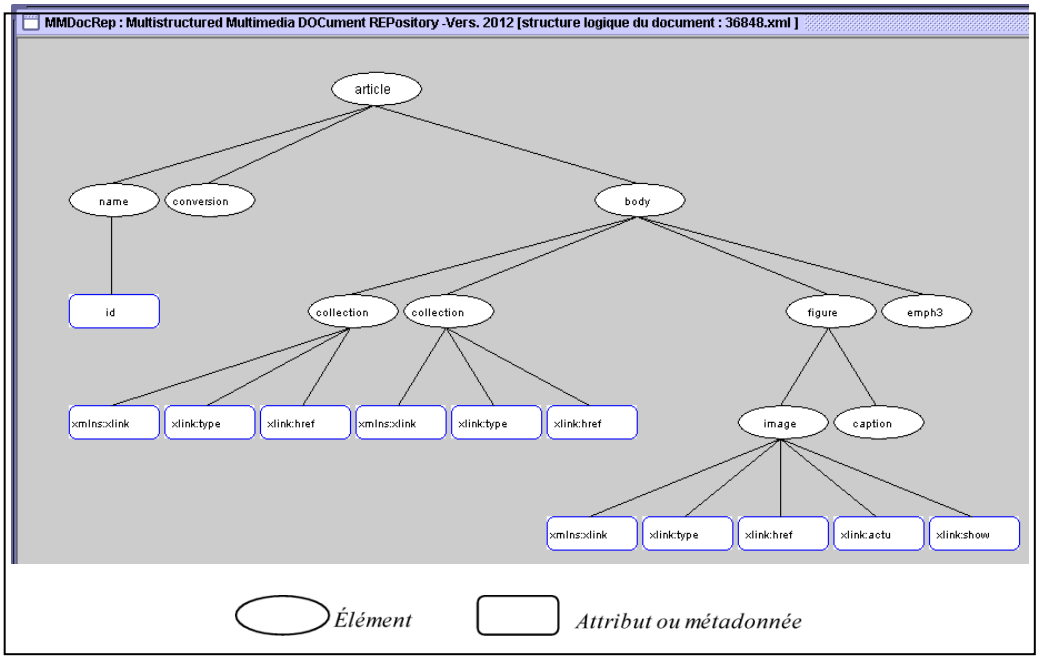


Figure III.17 - Structure logique du document de la figure III.16

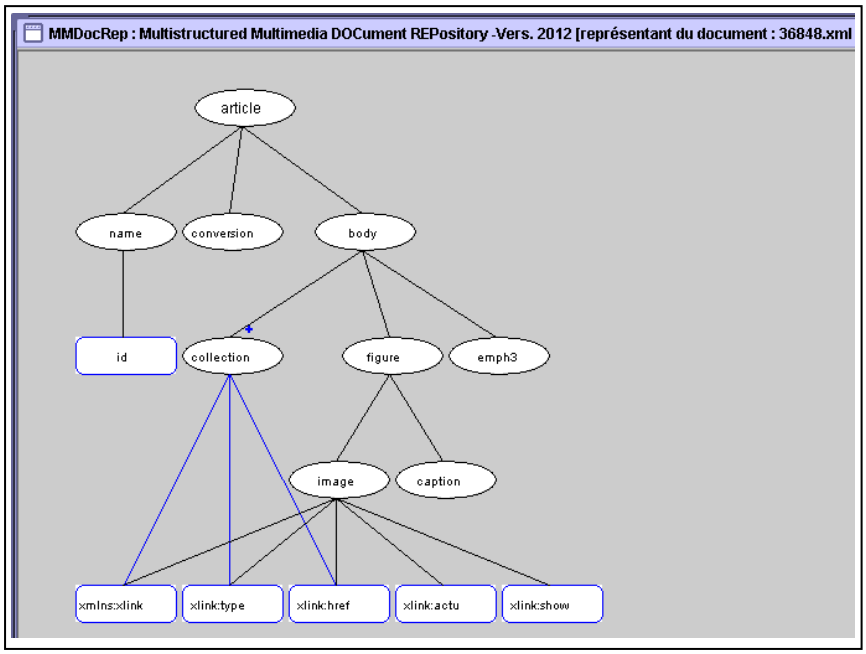


Figure III.18 - Exemple de représentant *Rep_d* du document de la figure 16

Nous considérons que l'utilisation des chemins est une solution qui permet entre autre d'enlever l'ambigüité des étiquettes multiples dans un graphe. Dans l'exemple de la figure III.18, le traitement du graphe *Rep_d* revient à traiter l'ensemble {chm_1, chm_2, chm_3, chm_4, chm_5, chm_6, chm_7, chm_8, chm_9, chm_{11}, chm_{12}} de ses chemins (figure III.19), avec :

chm_1 = *article/name/id,*

chm_2 = *article/conversionwarning,*

chm_3 = *article/body/collectionlink/xmlns:xlink,*

chm_4 = *article/body/collectionlink/xlink:type,*

chm_5 = *article/body/collectionlink/xlink:href,*

chm_6 = *article/body/figure/image/xmlns:xlink,*

chm_7= *article/body/figure/image/xlink:type,*

chm_8 = *article/body/figure/image/xlink:href,*

chm_9 = *article/body/figure/image/xlink:actuate,*

chm_{10} = *article/body/figure/image/xlink:show,*

chm_{11} = *article/body/figure/caption,*

chm_{12} = *article/body/emph3.*

Par exemple, une même étiquette « *xmlns:xlink* » pour deux nœuds : deux contextes différents. L'utilisation des chemins permet d'explorer un nœud dans son contexte (figure III.19).

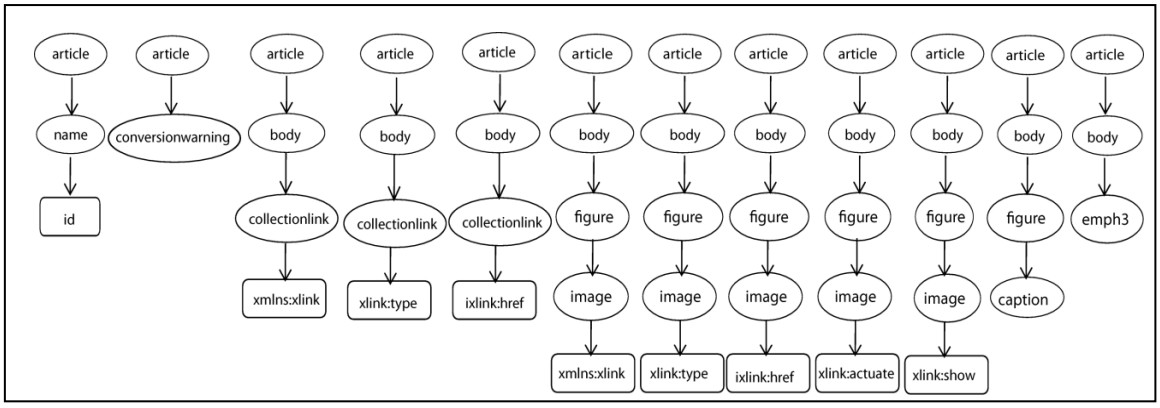


Figure III.19 - Ensemble des chemins du graphe *Rep_d* de la figure III.18

Dans notre approche, nous traitons un nœud selon son aspect hiérarchique en tenant compte des relations structurelles dans un chemin. Par exemple, dans le chemin chm_3 le nœud *« xmlns:xlink »* est traité comme étant le descendant du nœud *« collectionlink »*, lui-même descendant du nœud « *body* » et ainsi de suite. Plus précisément, le nœud seul n'est pas significatif ou du moins n'est pas suffisant pour prendre une décision par exemple.

Après l'extraction de la vue spécifique et le représentant *Rep_d* du document à intégrer. Ce dernier doit être comparé avec toutes les vues génériques (représentants des classes) de l'entrepôt afin de déterminer la vue générique la plus similaire à celui-ci. Formellement : soit $DW_g = \{Vg_1, Vg_2, \ldots, Vg_m\}$ l'ensemble des vues génériques de l'entrepôt documentaire où chaque vue *Vg* de DW_g est représentée sous forme de graphe orienté étiqueté et ordonné. Dans ce qui suit, nous posons *Rep_d* = (*V*,*E*) et *Vg* = (*V'*,*E'*) avec *V* (resp. *V'*) l'ensemble des nœuds du graphe *Rep_d* (resp. de *Vg*) et *E* (resp. *E'*) l'ensemble des arcs du graphe *Rep_d* (resp. de *Vg*).

La similarité entre les graphes *Rep_d* et *Vg* est basée sur la recherche d'un isomorphisme de sous-graphe entre les deux graphes. Cela engendre un problème combinatoire car il s'agit d'explorer tous les appariements (un-à-un) possibles entre les deux graphes. De plus, cet appariement doit être calculé avec l'ensemble des vues génériques de l'entrepôt documentaire. Ce problème suscite un autre problème lié aux temps de calculs et d'efficience de nos algorithmes. Dans la section suivante, nous présentons le *sous-processus de filtrage* permettant de réduire l'espace de comparaison lors de l'exploration des vues génériques de l'entrepôt.

IV.2.2. Filtrage des vues génériques candidates à la comparaison

Le processus de *filtrage* consiste à restreindre l'espace des vues génériques candidates à la comparaison. Dans cette étape, on sélectionne les vues génériques susceptibles d'être similaires au représentant *Rep_d* de la vue spécifique du document à intégrer (étape précédente). Il s'agit, pour chaque *Vg*, de chercher un appariement φ_n *injectif* entre *Rep_d* et *Vg* permettant la mise en correspondance des nœuds ayant la même étiquette (ou des étiquettes similaires) et qui permet :

(1) de préserver l'ordre des nœuds *fils* d'un même nœud,

(2) d'apparier un pourcentage de nœuds similaires qui dépend d'un *seuil de filtrage* (un paramètre fixé a priori).

Une vue générique est *candidate* à la comparaison si elle satisfait les conditions (1) et (2). On pourra donc définir la fonction Φ suivante :

$\Phi : \{Rep\_d\} \times DWg \rightarrow \{vrai, faux\}$

$(Rep\_d, Vg) \mapsto \Phi(Rep\_d, Vg)$

où $\Phi(Rep\_d, Vg) = vrai$ si la vue *Vg* est une vue *candidate* à la comparaison

$= faux$ sinon.

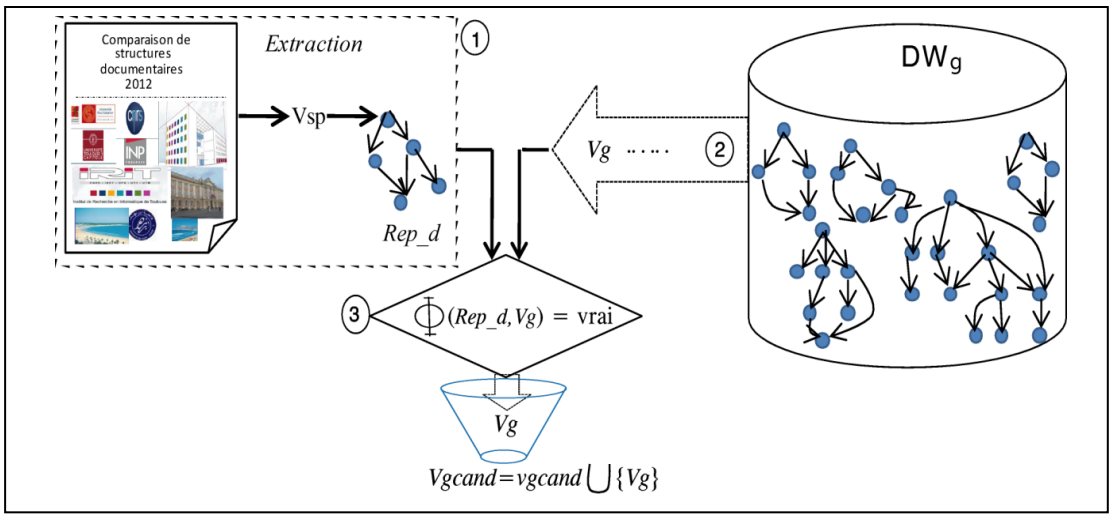


Figure III.20 - Architecture du processus de *filtrage*

IV.2.2.1. Préservation de l'ordre

Dans cette étape, nous imposons la contrainte de préservation de l'ordre des *nœuds fils* (d'un même nœud) sur l'appariement φ_n. Formellement, l'appariement φ_n doit vérifier :

$\forall (u,v) \in V_e^2$; *père*(*u*) = *père*(*v*) et o*rdre*(*u*) <= o*rdre*(*v*) et *père*($\varphi_n(u)$)= *père*($\varphi_n(v)$) $\Rightarrow$ o*rdre*($\varphi_n(u)$) <= o*rdre*($\varphi_n(v)$)

Nous pensons qu'il est important de tenir compte de l'ordre des nœuds frères dans un processus de comparaison structurelle. Par exemple, dans un journal télévisé la vidéo et l'audio doivent être synchronisés pour assurer la cohérence globale du journal. Dans l'exemple de la figure III.22, le scénario (*A*,*B*,*C*) est différent de (*C*,*B*,*A*).

Dans la figure III.21, la vue générique Vg_3 ne vérifie pas la condition (1) du filtrage. Plus précisément, dans cet exemple nous avons dans *Rep_d* : *père*(*B*) = *père*(*D*) et *ordre*(*B*) < *ordre*(*D*). En revanche dans Vg_3, *père*(*D*) = *père*(*B*) mais *ordre*(*D*) < *ordre*(*B*).

Remarque : Les nœuds de type attribut et métadonnée ne seront pas concernés par cette étape (l'étape (1)).

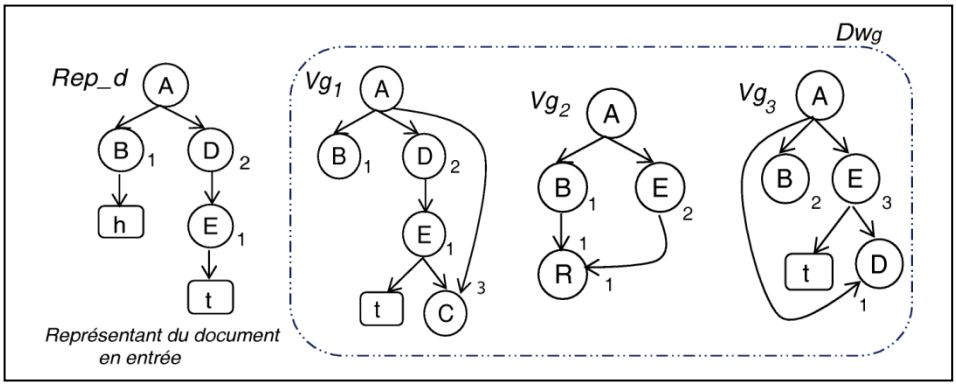


Figure III.21 - Exemple de vues à comparer

Une vue générique dont l'ordre des nœuds *fils* n'est pas respecté est considérée suffisamment différente et elle sera éliminée de la liste des vues candidates à la comparaison. En effet, dans un document multimédia, le sens ne dépend pas seulement de la signification des éléments structurels de ce document mais elle concerne également la synchronisation et le lien (avant, pendant, après,…) entre ces éléments. Dans notre contexte, nous considérons que les relations (des fois implicites), entre les éléments structurels d'un document, représentent des informations supplémentaires qui complètent la signification globale et contextuelle de ces éléments. Une même image, par exemple, dans deux documents différents peut ne pas exprimer le même contexte et donc peut ne pas avoir la même importance dans les deux documents [Idarrou A. et al., 2012a]. Dans l'exemple de la figure III.22, la cohérence globale de la présentation dépend de la synchronisation des objets qui la composent.

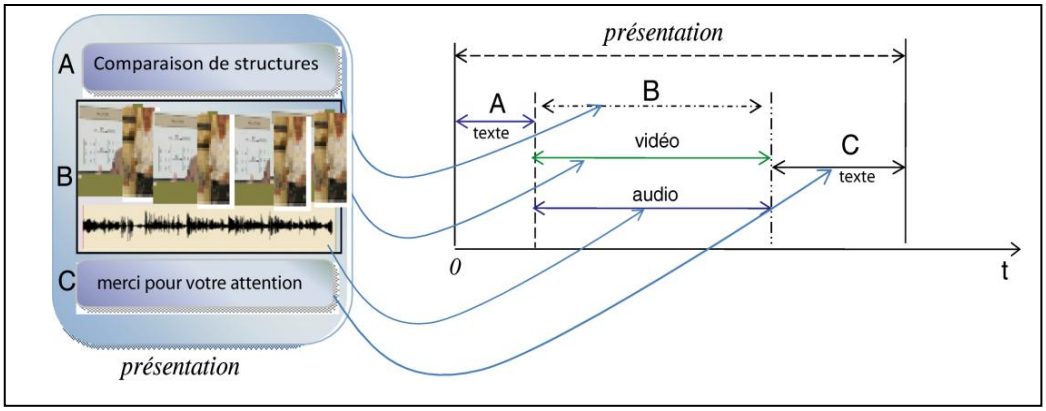


Figure III.22 - Exemple de synchronisation entre composants d'un document

A l'issue de cette étape, un ensemble $Vgcand_1$ de vues génériques, vérifiant la condition (1) du processus de filtrage, est retenu pour l'étape suivante.

Dans l'exemple de la figure III.21, $Vgcand_1 = \{Vg_1, Vg_2\}$ est l'ensemble des vues génériques sélectionnées dans cette étape.

IV.2.2.2. Présélection des vues génériques de l'entrepôt

Cette étape permet de sélectionner parmi les vues génériques vérifiant l'étape précédente celles qui ont un certain degré de similarité avec le représentant *Rep_d* du

document à intégrer. Ce degré de ressemblance est évalué par la fonction C_f sur la base d'un pourcentage de nœuds communs à *Rep_d* et *Vg*. La fonction C_f permet de mesurer le degré de ressemblance entre *Rep_d* et une vue générique *Vg* :

$$C_f : \{Rep\_d\} \text{ x } DW_g \rightarrow [0,1]$$

$$(Rep\_d, Vg) \mapsto C_f(Rep\_d, Vg)$$

Contrairement à la mesure de ressemblance proposée[1] dans [Idarrou A. et al., 2010a], dans la formule [31], qui définit la fonction C_f, nous tenons compte de la nature (élément ou attribut/métadonnée) des nœuds communs entre *Rep_d* et *Vg* :
avec :

$$C_f(Rep\_d, Vg) = \begin{cases} \left[\frac{|\varphi_n(V_e)|}{|V_e|} + \frac{|\varphi_n(V_a)|}{|V_a|} + \frac{|\varphi_n(V'_e)|}{|V'_e|} + \frac{|\varphi_n(V'_a)|}{|V'_a|} \right] / 4 \text{ si } |V_a| \neq 0 \text{ et } |V'_a| \neq 0 \\ \left[\frac{|\varphi_n(V_e)|}{|V_e|} + \frac{|\varphi_n(V'_e)|}{|V'_e|} \right] / 2 \text{ si } |V_a| = 0 \text{ et } |V'_a| = 0 \\ \left[\frac{|\varphi_n(V_e)|}{|V_e|} + \frac{|\varphi_n(V'_e)|}{|V'_e|} \right] / 3 \text{ sinon c'est à dire si } |V_a| = 0 \text{ ou bien } |V'_a| = 0 \end{cases} \quad [31]$$

- φ_n est l'appariement bidirectionnel de *V* vers *V'* (resp. de *V'* vers *V*) qui permet de mettre en correspondance un nœud de *V* (resp. de *V'*) à au plus un nœud similaire (ayant la même étiquette ou une étiquette similaire) et de *même type* de *V'* (resp. de *V*).
- $V = V_e \cup V_a$ où V_e est l'ensemble des nœuds de type élément de *Rep_d* et V_a est l'ensemble des nœuds de type attribut (ou métadonnée) de *Rep_d*,
- $V' = V'_e \cup V'_a$ où V'_e est l'ensemble des nœuds de type élément de *Vg* et V'_a est l'ensemble des nœuds de type attribut (ou métadonnée) de *Vg*. Avec $V_e \neq \varnothing$ et $V'_e \neq \varnothing$,
- $\varphi_n(V) = \{u' \in V' / \exists\ u \in V ;\ \varphi_n(u)=u'\}$ l'ensemble de nœuds de *V'* appariés à des nœuds de *V*,
- $\varphi_n(V') = \{u \in V / \exists\ u' \in V' ;\ \varphi_n(u')=u\}$ l'ensemble de nœuds de *V* appariés à des nœuds de *V'*,
- $\frac{|\varphi_n(V_e)|}{|V_e|}$ le pourcentage de nœuds, de type élément de *Rep_d* appariés à des nœuds de type élément de *Vg*,

[1] $C_f(Rep\_d, Vg) = \left[\frac{|\varphi_n(V)|}{|V|} + \frac{|\varphi_n(V')|}{|V'|} \right] / 2$

- $\frac{|\varphi_n(V_a)|}{|V_a|}$ le pourcentage de nœuds de type attribut ou métadonnée de *Rep_d* appariés à des nœuds de type attribut ou métadonnée de *Vg*,
- |X| est le cardinal de l'ensemble X.

Le troisième cas de la formule [31] permet de traiter le cas où l'un des ensembles V_a ou bien V'_a est vide mais pas les deux à la fois (un troisième terme implicite). La division par 3 permet de pénaliser les nœuds de type attribut (des deux graphes) qui ne sont pas mis en correspondance (exemple de la figure III.21, lorsqu'il s'agit de comparer *Rep_d* et Vg_2). En effet, si V_a est vide mais V'_a est non vide alors $|\varphi_n(V'_a)| = 0$ car les éléments de V'_a n'ont aucun correspondants dans V_a ($V_a = \varnothing$). Même raisonnement si V'_a est vide mais V_a est non vide.

Dans cette étape, on ne tient compte que du nombre de nœuds alignés. On obtient un ensemble de vues $Vgcand = \{Vg \in Vgcand_1 / C_f(Rep\_d, Vg) >= S_f\}$. La valeur de S_f (*seuil de filtrage*) doit être déterminée a priori par expérimentations.

D'après les structures de la figure III.21 :

- *Rep_d* est décrite par les nœuds de type élément $V_e = \{A,B,D,E\}$ et les nœuds de type attribut $V_a=\{h,t\}$,
- Vg_1 est décrite par les nœuds de type élément $V1_e = \{A,B,E,D,C\}$ et les nœuds de type attribut $V1_a=\{t\}$,
- Vg_2 est décrite par les nœuds de type élément $V2_e = \{A,B,E,R\}$ et les nœuds de type attribut $V2_a=\varnothing$,

Ainsi, la mesure de ressemblance entre *Rep_d* et ces deux vues génériques est calculée comme suit :

$$C_f(Rep\_d, Vg_1) = (\frac{4}{4} + \frac{1}{2} + \frac{4}{5} + \frac{1}{1}) / 4 = 0,82,$$

$$C_f(Rep\_d, Vg_2) = (\frac{3}{4} + \frac{0}{2} + \frac{3}{5}) / 3 = 0,45 \quad (V_a \neq \varnothing).$$

La division par 3 dans le deuxième exemple permet de pénaliser les nœuds *h* et *t* de *Rep_d* (de type attribut) qui ne sont pas appariés.

A l'issue du processus de *filtrage*, on obtient un ensemble de vues candidates $Vgcand = \{Vg \in DW_g / \Phi(Rep\_d, Vg) = vrai\}$. Cet ensemble sera retenu pour les étapes suivantes du processus de comparaison. Dans l'exemple de la figure III.21, avec un *seuil de filtrage* de 70% seule la vue représentée par le graphe Vg_1 sera retenue comme vue candidate à la comparaison : $Vgcand = \{Vg_1\}$.

La sélection des vues génériques candidates est assurée par l'algorithme « *FiltrerVueGenCand »* que nous avons introduit (Cf. Annexe page 181).

IV.2.3. Comparaison de structures

IV.2.3.1. Transformation de vues génériques

a) Principe

L'objectif de la transformation des vues est de rendre les représentants des classes plus représentatifs et par conséquent d'optimiser le volume de stockage de l'entrepôt documentaire. Cette étape consiste à rapprocher chaque vue générique *candidate* à la comparaison, du représentant *Rep_d* de la vue spécifique du document à intégrer. D'éventuels ajouts de fragments de *Rep_d*, manquants dans chacune des vues candidates, peuvent être envisagés (Cf. Annexe, page 189). Contrairement à l'approche de « résumé d'arbres » de [Dalamagas T. et al., 2004] (Cf. chapitre II, section III.3.3 page 71), dans cette étape nous respectons à la fois l'ordre des nœuds (et des arcs) et la préservation des arcs (exemple figure III.23) sans perte des composantes du graphe transformé (sans perte d'informations).

Exemple de transformation d'un graphe

Dans la figure III.23, les chemins chm_1=*A/B/H* et chm_2=*A/D/E/K* de *Rep_d* ont un degré de similarité avec les chemins respectivement chm'_1=*A/B* et chm'_2=*A/E/K* de *Vg*. Dans cet exemple, il s'agit d'ajouter les fragments : (*B*,*H*), (*A*,*D*) et (*D*,*E*) (ajout des nœuds et des arcs), de la vue représentée par *Rep_d*, qui n'existent pas dans la vue générique *Vg*.

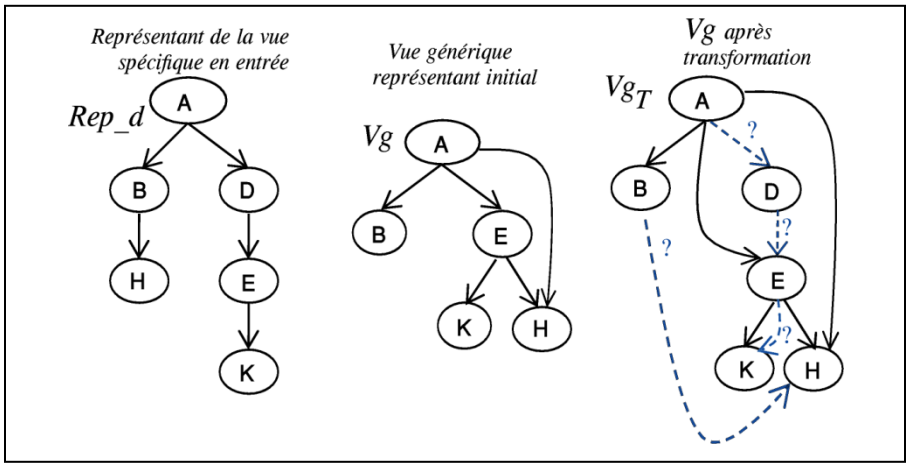


Figure III.23 - Exemple de transformation d'un graphe

Deux questions peuvent être posées à ce niveau : la première est comment enrichir les vues génériques en ajoutant des nœuds et des arcs : (1) sans perdre de l'information et (2) sans perturber la classe dont le représentant a subi une transformation ? La deuxième question est quel est l'impact de la transformation sur la qualité des classes ?

L'insertion des nœuds et des arcs ne doit pas être à l'origine de perte d'informations. Par exemple, dans le graphe *Vg* de la figure III.23, lorsqu'on a inséré le nœud *D*, nous avons conservé la relation (*A*,*E*) et par conséquent conservé les chemins *A*/*E*/*K* et *A*/*E*/*H* de *Vg*. Mais en même temps, nous avons ajouté le chemin *A*/*D*/*E*/*K*, dans la même vue Vg_T, pour que celle-ci représente le graphe *Rep_d*. Plus précisément, dans cet exemple, la transformation vise à obtenir une structure capable de représenter à la fois *Rep_d* et *Vg*.

L'insertion des nœuds et des arcs ne doit pas perturber la classe dont le représentant a subi une transformation. La vue Vg_T doit être similaire à *Vg* (la vue avant la transformation) et doit représenter tous les documents déjà représentés par *Vg* mais en même temps, elle doit représenter *Rep_d*. Pour cela, nous avons proposé d'utiliser la notion d'*arc optionnel* marqué « ? » comme information supplémentaire (cardinalité) sur l'arc (*A,D*) qui signifie que cette arc est optionnel (noté $(A,D)_?$).

Convention :

Lors de la comparaison de deux chemins, les arcs (et les nœuds) *optionnels* qui n'existent pas dans l'un des chemins ne seront pas pris en considération.

Dans l'exemple de la figure III.23, nous montrons que le graphe *Vg* est isomorphe à Vg_T:

Posons $Rep\_d = (V_1,E_1)$, $Vg = (V_2,E_2)$ et $Vg_T = (V_3,E_3)$

$V_1 = \{A,B,H,D,E,K\}$, $E_1 = \{(A,B),(B,H),(A,D),(D,E),(E,K)\}$

$V_2 = \{A,B,E,K,H\}$, $E_2 = \{(A,B),(A,E),(E,K),(E,H),(A,H)\}$

$V_3 = \{A,B,D_?,E,K,H\}$, $E_3 = \{(A,B),(B,H)_?,(A,E),(A,D)_?,(D,E)_?,(E,K),(E,H),(A,H)\}$

{*A*/*B*/*H*, *A*/*D*/*E*/*K*} l'ensemble des chemins terminaux de *Rep_d.*

{*A*/*B*,*A*/*E*/*K*,*A*/*E*/*H*,*A*/*H* } l'ensemble des chemins terminaux de *Vg.*

$\{A/B/_?H,A/E/K,A/E/H,A/_?D/_?E/_?K,A/H\}$ l'ensemble des chemins terminaux de Vg_T.

avec $X/_?Y$: signifie que l'arc (*X*,*Y*) est optionnel.

Nous avons $V_2 \subseteq V_3$ et $E_2 \subseteq E_3$ donc (Cf. chapitre II, section I.2.2 page 55) *Vg* est un sous-graphe de Vg_T ($Vg \subseteq Vg_T$).

De même, nous montrons que Vg_T est un sous-graphe de *Vg* :
En effet :
$V_3 \subseteq V_2$ (car *D* est optionnel) et $E_3 \subseteq E_2$ (car les (*B*,*H*),(*A*,*D*) et (*D*,*E*) sont optionnels)
Donc Vg_T est un sous-graphe de *Vg*, par conséquent les graphes *Vg* et Vg_T sont isomorphes.

De même, le graphe *Rep_d* est isomorphe à un sous-graphe de Vg_T.
En effet :
$V_1 \subseteq V_3$ et $E_1 \subseteq E_3$

Donc *Rep_d* est isomorphe à un sous-graphe de Vg_T ($Rep\_d \subseteq Vg_T$).

Conclusion :

La transformation d'une vue générique *Vg* vise à enrichir celle-ci en ajoutant les nœuds et les arcs de *Rep_d* qui n'existent pas dans *Vg*. Cela permet donc d'augmenter la représentativité de *Vg*. Après la transformation, le graphe Vg_T ainsi obtenu permet de représenter à la fois *Rep_d* et *Vg*.

Notre apport dans cette étape, par rapport aux travaux de [Mbarki M., 2008], est que l'ajout des nœuds ne doit pas engendrer une perte d'informations (arcs).

La figure suivante illustre un exemple d'ajout de nœuds (dans l'arbre *T* : arbre initial) selon l'approche de [Mbarki M., 2008]. L'ajout du nœud « *Langue* » a engendré la perte de la

relation (*Locuteur*,*Trans*), une information pertinente dans un processus de comparaison de structures.

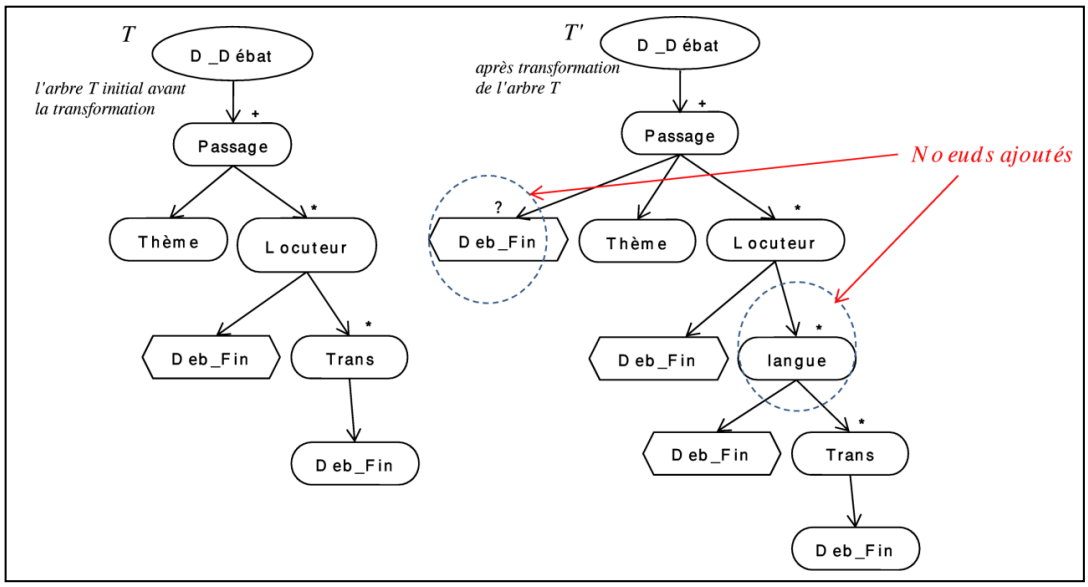


Figure III.24 - Exemple d'ajout des nœuds selon l'approche de [Mbarki M., 2008]

b) Impact de la transformation sur la qualité des classes

La qualité d'une classe dépend de la cohérence (homogénéité) des individus qui la composent. Cette cohérence peut être mesurée par la distance intra-classe. Plus cette distance est faible plus la classe est homogène. En effet, l'homogénéité de la classe traduit le lien entre les éléments qui la constituent. Plus le lien entre les individus d'une classe est fort plus la classe est homogène.

Après la classification des éléments d'un ensemble E en un ensemble $C = \{c_1,c_2,...,c_m\}$ de m classes, les classes générées doivent vérifier :

(1) $\forall c_i \in C$; $c_i \neq \varnothing$; une classe représente au moins la vue spécifique du document ayant généré la classe c_i,

(2) $\forall (c_i,c_j) \in C^2$; $i \neq j \Rightarrow c_i \cap c_j = \varnothing$; les classes sont disjointes (*séparation*)

(3) $E = \bigcup_{i=1}^{m} c_i$; la réunion des classes constitue l'ensemble E (*ensemble initial*).

Le problème de transformation des classes est lié à la question : à quel moment il faut arrêter la transformation d'une classe ?

La transformation peut engendrer un problème de rapprochement des classes (diminuer la distance inter-classe) et par conséquent perturber la classification. En effet, quand les classes sont très proches, il peut y avoir ambiguïté et appartenance d'un même document à deux classes différentes. Ce phénomène conduit à une classification dont les individus des classes sont hétérogènes : la distance inter-classe diminue en revanche la distance intra-classe augmente. Lorsque deux classes sont très proches alors il n'y a plus intérêt à les garder : elles doivent être fusionnées en une seule classe.

La séparation des classes est l'un des critères pour qualifier un classifieur. Dans [Oh,Il-Seok et al., 1999], une séparation plus large des classes implique un meilleur pouvoir discriminant. Selon [Bisson G., 2000], deux objets lointains représentent des données qui appartiennent à des groupes différents.

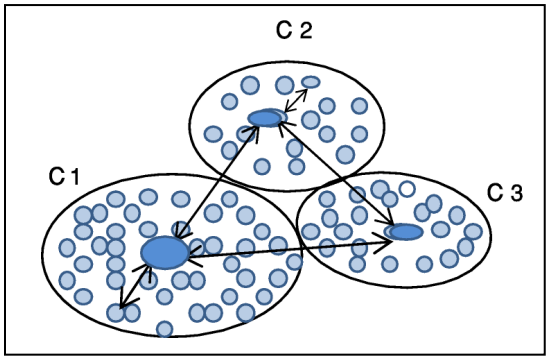


Figure III.25 - Illustration de la distance intra et inter-classe

Pour maintenir la stabilité des classes et conserver leur qualité, nous avons proposé [Idarrou A. et al., 2012b] de fixer a priori *une distance minimale* inter-classe. Augmenter la distance entre classes permet de diminuer le bruit et augmenter la précision de la classification. L'utilisation de ce paramètre est une solution au problème de rapprochement des classes.

Notre apport à ce niveau par rapport aux travaux de [Mbarki M., 2008] et [Djemal K., 2010], est la prise en compte de *la séparation des classes* (une distance minimale inter-classe est fixée a priori) lors de leurs transformations (Cf. section IV.2.3.1, page 123). Cela permet de vérifier la distance inter-classe (séparation des classes, figure III.25) des vues génériques avant et après la transformation. Le fait de ne pas tenir compte de ce paramètre, permet de continuer à transformer (faire évoluer) les classes sans cesse. A un certain moment, une (ou plusieurs) classe(s) va (vont) dominer (absorber) les autres classes. Cela peut engendrer une perturbation des classes.

Pour cela, nous avons proposé la fonction « *Separat()* » (Cf. Annexe page 185) qui permet de vérifier la séparation des classes (vues génériques) de l'entrepôt documentaire.

A la fin de cette étape, on retient l'ensemble des vues génériques transformées (et vérifiant la condition de séparation) ainsi que le coût de transformations de chacune de ces vues pour l'étape suivante.

IV.2.3.2. Pondération des vues

Pour pondérer les graphes représentant les vues documentaires, nous utilisons la fonction de pondération que nous avons définie dans la section III.2 (page 102), de ce chapitre.

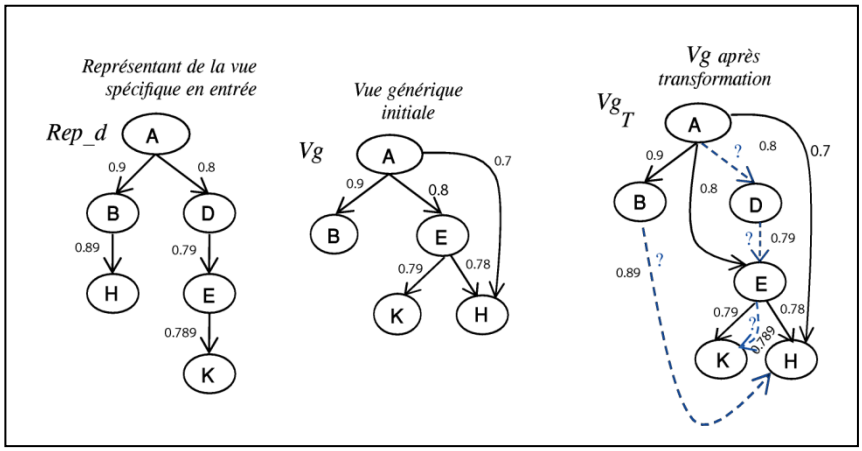


Figure III.26 - Exemple de pondération d'un graphe après sa transformation

Dans le graphe Vg_T (graphe transformé) de la figure III.26, le nœud *D* ajouté a le même ordre que le nœud *E* ce qui se traduit par le même poids des arcs (*A*,*E*) et (*A*,*D*). L'arc (*E*,*K*) possède deux poids :

- Si on considère le chemin *A/D/E/K*, le poids de (*E*,*K*) est 0.789. Les trois chiffres décimaux signifient que la profondeur du nœud *K*, dans ce chemin, est égale à 3,
- Si on considère le chemin *A/E/K*, le poids de (*E*,*K*) est 0.79. Les deux chiffres décimaux signifient que la profondeur du nœud *K*, dans ce chemin, est égale à 2.

IV.2.3.3. Calcul du score de similarité

Cette étape permet de calculer le score de similarité entre le graphe *Rep_d* (le représentant de la vue spécifique du document à intégrer) et chacun des graphes représentant les vues *génériques candidates* en utilisant la mesure de similarité proposée (Cf. section III.4, page 108).

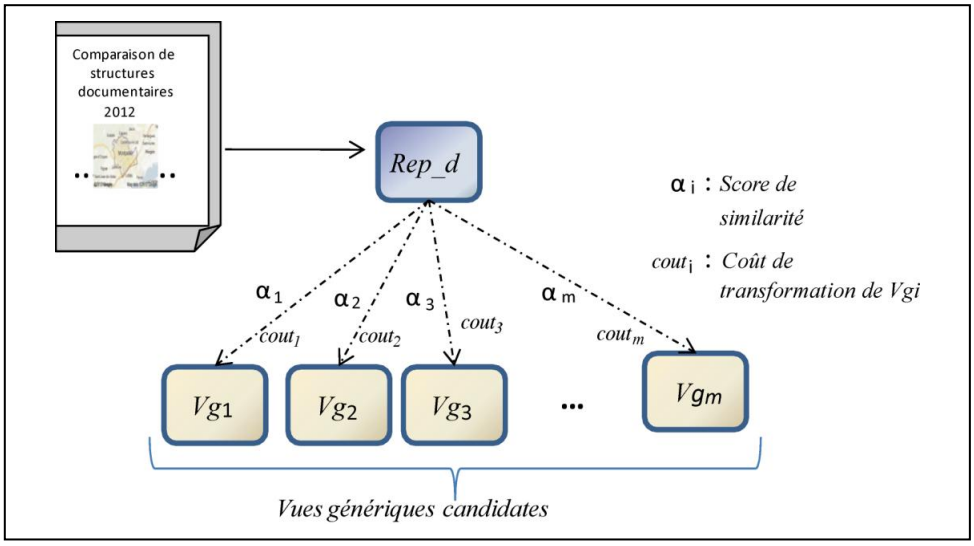


Figure III.27 - Calcul des scores de similarité entre le représentant du document et les vues génériques existantes

A l'issue de cette étape, on retient l'ensemble des vues génériques ainsi que le score de similarité et le coût de transformation de chacune de ces vues (figure III.27) avec le représentant du document à intégrer pour l'étape suivante.

IV.2.4. Décision

Une fois que toutes les vues génériques candidates ont été traitées (étapes de comparaison IV.2.3.1 à IV.2.3.3), le système passe à la dernière étape pour prendre la décision finale. Rappelons que notre problématique dans cette étape est de déterminer, parmi l'ensemble des vues génériques de l'entrepôt documentaire, la vue générique la plus similaire à la vue spécifique du document à intégrer.

a) Principe

Cette étape consiste à extraire, parmi l'ensemble des vues génériques, celle dont le *degré de similarité* avec la vue spécifique du document à intégrer est *le plus élevé*, puis à comparer ce degré avec le *seuil de similarité* (*Seuil_Sim* $\in$]0,1] un paramètre fixé par expérimentation). Deux cas se présentent :

- si ce degré de similarité est strictement inférieur à *Seuil_Sim*, une nouvelle classe sera créée à partir du représentant *Rep_d* de la vue spécifique du document à intégrer,
- si ce degré de similarité est supérieur ou égale à *Seuil_Sim* alors deux cas peuvent être envisagés :
  - une seule vue générique est similaire à la vue spécifique du document à intégrer. Dans ce cas la vue spécifique est rattachée à celle-ci.
  - plusieurs vues génériques sont similaires à la vue spécifique du document à intégrer. Dans ce cas on choisit celle pour laquelle l'intégration de cette nouvelle vue spécifique nécessitera le moins de transformations (moindre coût).

Concernant le coût de transformation d'un graphe en un autre, nous avons proposé de faire la somme des coûts des opérations élémentaires (les opérations d'ajout des fragments) :

$$\sum_{i=1} coût_i \text{ où } coût_i \text{ est le coût d'une opération d'ajout d'un arc } (u_i, v_i)$$

Avec $coût_i = \frac{\alpha_i}{k^{prof(v_i)}}$ où α_i et k sont deux paramètres (Cf. section III.2, page 100) qui reflètent les aspects hiérarchique et contextuel des éléments structurels des graphes comparés.

Dans l'exemple de la figure III.26 (*k=10*) le coût de transformation de *Vg* en Vg_T est :

$$coût_1 + coût_2 + coût_3 + coût_4 = \frac{1}{100} + \frac{2}{10} + \frac{1}{100} + \frac{1}{1000} = 0.221$$

- $coût_1$: coût de l'opération d'ajout de l'arc (*B,H*) (α_1 = *ordre(H)= 1 et prof(H)=2*)

- $coût_2$: coût de l'opération d'ajout de l'arc (*A,D*) (α_2 = *ordre(D)= 2 et prof(D)=1*)
- $coût_3$: coût de l'opération d'ajout de l'arc (*D,E*) (α_3 = *ordre(E)= 1 et prof(E)=2*)
- $coût_4$: coût de l'opération d'ajout de l'arc (*E,K*) (α_4 = *ordre*(*K*)=1 et *prof*(*K*)=3)

Une classe notée C_i représentée par Vg_i, représentant *n* vues spécifiques, peut être définie formellement comme suit :

$$C_i = \{Vsp_k / k \in [1,n] ; Sim(Rep\_d_k, Vg_i) >= Seuil\_Sim\}$$

Où Vsp_k est une vue spécifique rattachée à la vue générique Vg_i (exemple figure III.3 page 94), Rep_d_k est le représentant de la vue spécifique Vsp_k (Cf. section IV.2.1, page 115) et *Sim* est la fonction de similarité structurelle que nous avons définie dans la section III.4 de ce chapitre.

b) Le choix du seuil de similarité

Concernant le seuil de similarité *Seuil_Sim*, d'autres tests ont été réalisés afin de déterminer la valeur optimale [Idarrou A., 2010]. Dans ces travaux, nous avons fait des tests en faisant varier le *Seuil_Sim*. Nous avons constaté que l'augmentation de la valeur de *Seuil_Sim* entraîne la création de nombreuses classes. En revanche, la diminution de cette valeur implique la croissance du nombre de documents rattachés à chaque classe ce qui suscite une hétérogénéité entre documents d'une même classe. Nous avons constaté que la valeur de 0.80 (80% de similarité) a donné des résultats satisfaisants (des classes homogènes).

c) Séparation des classes

Au fur et à mesure de la construction des classes, les vues génériques sont transformées. Ces transformations peuvent amener à un rapprochement de ces classes, voire un chevauchement. Pour conserver le pouvoir discriminant des vues génériques, il faut s'assurer qu'elles soient suffisamment distantes (Cf. section IV.2.3.1, page 123).

Rappelons que la fonction de similarité entre deux graphes *G* et *G'* (Cf. section III.4, page 108) a été définie comme suit : $Sim(G,G') = 1 - Dist(G,G')$.

Soit $C = \{c_1, c_2, \ldots c_n\}$ l'ensemble des classes où une classe c_i est représentée par Vg_i. Celle-ci regroupe un ensemble de vues spécifiques structurellement proches.

Soit *X* le représentant d'une vue spécifique en **entrée** et soit $i \in [1,n]$, deux cas peuvent être envisagés :

- *X* est similaire à Vg_i donc $Sim(X,Vg_i) >= Seuil\_Sim$

  c'est à dire $1 - Dist(X,Vg_i) >= Seuil\_Sim$

  d'où $\mathbf{Dist(X,Vg_i) <= 1 - Seuil\_Sim}$ [32]

- *X* est non similaire à Vg_i donc $Sim(X,Vg_i) < Seuil\_Sim$

  c'est-à-dire $1 - Dist(X,Vg_i) < Seuil\_Sim$

  d'où $\mathbf{1 - Seuil\_Sim < Dist(X,Vg_i)}$ [33]

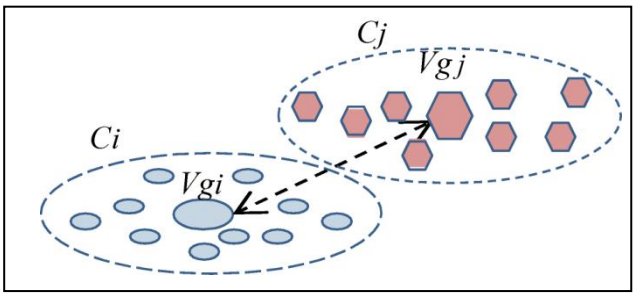


Figure III.28 – Illustration de la distance inter-classe

Comme nous l'avons évoqué précédemment au cours des ce chapitre (Cf. section IV.2.3.1, page 123), les classes doivent être suffisamment séparées. Dans ce contexte, nous proposons de fixé a priori une *distance minimale* inter-classe.

Formellement, les classes doivent vérifier :

$\forall\ i,j \in [1,n]\ ;\ i \neq j\ ;\ Sim(Vg_i,Vg_j) < Seuil\_Sim$ (sinon les classes c_i et c_j sont similaires)

c'est-à-dire $\forall\ i,j \in [1,n]\ ;\ i \neq j\ ;\ 1 - Dist(Vg_i,Vg_j) < Seuil\_Sim$

d'où $\forall\ i,j \in [1,n]\ ;\ i \neq j\ ;\ \mathbf{1 - Seuil\_Sim < Dist(Vg_i,Vg_j)}$ [34]

V. Conclusion

La classification des objets structurés nécessite d'utiliser un modèle de représentation riche et expressif et un outil de comparaison en adéquation avec la nature de ces objets. Dans notre approche, nous avons choisi d'utiliser les graphes pour représenter les documents multimédias à structures multiples afin de les comparer d'un point de vue structurel. La comparaison des documents se traduit par la comparaison des graphes qui leur sont associés. Pour évaluer la ressemblance (ou la différence) entre les graphes, et plus généralement les objets structurés, il devient nécessaire d'avoir une mesure de similarité ou de distance.

Au cours de ce chapitre, nous avons définit dans un premier temps notre approche de classification documentaire. Ensuite, nous avons introduit une mesure de similarité structurelle basée sur l'isomorphisme de sous-graphes. L'originalité de cette mesure réside dans :

- la définition d'une fonction de pondération qui permet d'attribuer un poids à chacun des arcs composant un graphe donné. Ce poids traduit les aspects hiérarchique et contextuel de ces composants (la position des composantes : ordre, profondeur),
- le fait que cette mesure reflète à la fois la structure et le sens des documents comparés dans le sens où elle repose sur la distance d'alignement des chemins : tenir compte des relations structurelles. En effet le sens d'un document ne dépend pas seulement des éléments structurels qui le composent mais il dépend également des relations, porteuses d'informations supplémentaires, entre ces éléments. La mesure proposée retourne une valeur continue, entre 0 et 1, à partir de laquelle il est possible de répondre à des questions du genre :

- deux structures sont identiques (similaires),
- une structure est totalement incluse dans une autre,
- une structure est partiellement incluse dans une autre,
- deux structures sont disjointes ou non.

Dans un deuxième temps, nous avons présenté notre approche de classification structurelle non-supervisée dans le cadre d'un entrepôt de documents. Cette approche est composée globalement de deux phases : (1) l'extraction de la vue spécifique d'un document à intégrer et (2) le rattachement de cette vue spécifique à la vue générique la plus similaire. L'étape (2) passe par un processus de comparaison de la vue spécifique de l'étape (1) avec l'ensemble des vues génériques (les classes déjà générées) de l'entrepôt de documents. Notre approche de classification est caractérisée par :

- le fait qu'un document n'est plus vu comme un ensemble de composants mais plutôt comme un ensemble organisé de granules reliés les uns aux autres de façon cohérente,
- l'utilisation des graphes (intégrant les structures et leurs liens) comme modèle universel et flexible pour représenter les documents multimédias à structures multiples et l'exploitation de la théorie des graphes pour comparer ces structures,
- la définition d'une fonction (mesure de ressemblance : formule [31]) permettant de réduire l'espace de comparaison des structures (Cf. section IV.2.2 page 121), en sélectionnant les vues génériques de l'entrepôt susceptibles d'être similaires à la vue spécifique du document à intégrer. Cela permet d'optimiser le temps de réponse de nos algorithmes de comparaison,
- la transformation des vues génériques qui permet de déterminer la vue générique communes à un ensemble de documents proches (optimiser le volume de stockage) avec la prise en compte du coût de cette transformation,
- l'utilisation d'un *seuil de similarité* permettant de fixer le degré de similarité entre un document et le représentant de la classe à laquelle il est rattaché,
- l'utilisation d'une *distance inter-classe* pour conserver la séparation des classes.

VI. Bibliographie

[Bisson G., 2000] Bisson G., La similarité : une notion symbolique/ numérique. Apprentissage symbolique-numérique. Eds Moulet, Brito, Cepadues Edition, 2000.

[Dalamagas T. et al., 2004] Dalamagas T., Cheng T., Winkel K-J. and Sellis T., « Clustering XML Documents Using Structural Summaries », In *EDBT Workshops* p 547-556, 2004.

[Djemal K. 2010] Djemal Karim, « De la modélisation à l'exploitation des documents à structures multiples », Thèse de Doctorat de l'Université de Paul Sabatier. - Toulouse, France, 2010.

[Gentner D., 1983] Gentner D., "Structure-mapping: A theoretical framework for analogy". *Cognitive Science*, 7, 155-170. (Reprinted in A. Collins & E. E. Smith (Eds.), *Readings in cognitive science: Aperspective from psychology and artificial intelligence*. Palo Alto, CA: Kaufmann), 1983.

[Gentner D. et al. ,1989] Gentner D. "*The mechanisms of analogical learning". In* S. Vosniadou et A. Ortony (dir.), *Similarity and analogical reasoning,* p. 199-241. Cambridge: Cambridge University Press. 1989.

[Hummel J.E., 2000] Hummel, J.E. "*Where view-based theories break down: The role of structure in shape perception and object recognition",* In E. Dietrich and A. Markman (Eds.). Cognitive Dynamics: Conceptual Change in Humans and Machines. Hillsdale, NJ: Erlbaum, 2000.

[Hummel J.E, 2001] Hummel, J.E. "*Complementary solutions to the binding problem in vision: Implications for shape perception and object recognition" Visual Cognition,* 8, p. 489-517, 2001.

[Idarrou A., 2010] Idarrou A., « Classification de documents multi-structurés : Comparaison des structures » Dans : *Colloque International Francophone sur l'Écrit et le Document (CIFED 2010), Sousse (Tunisie), 18/03/2010-20/03/2010,* Cépaduès, p. 501-506, 2010.

[Idarrou A. et al., 2010a] Ali Idarrou, Driss Mammass, Chantal Soulé-Dupuy, Nathalie Vallés-Parlangeau. « Classification of multi-structured documents: A comparison based on media image » *Lecture Notes in Computer Science***,** Vol. LNCS 6134, Springer, p. 428-438, 2010.

[Idarrou A. et al., 2010b] Ali Idarrou, Driss Mammass, Chantal Soulé-Dupuy, Nathalie Vallés-Parlangeau. "A generic Approach to the Classification of Multimedia Documents: a Structures Comparison". Dans : ICGST International Journal on Graphics, Vision and Image Processing, The International Congress for global Science and Technology, Vol. Volume 10 N. Issue VI, p. 13-18, december 2010.

[Idarrou A. et al., 2012b] Ali Idarrou and Driss Mammass, "*Structural Clustering Multimedia Documents*: *An Approach based on Semantic Sub-graph Isomorphism*". *International Journal of Computer Applications* 51(1):14-21, August 2012. Published by

Foundation of Computer Science, USA, Vol. 51 N. 1, August 2012. Accès : http://www.ijcaonline.org/archives/volume51/number1/8005-1343.

[Laroum S., et al., 2009] Sami Laroum, Nicolas Béchet, Hatem Hamza et Mathieu Roche, "Classification automatique de documents bruités à faible contenu textuel", Manuscrit auteur, publié dans RNTI : Revue des Nouvelles Technologies de l'Information 1, 2009.

[Mbarki M. 2008] Mbarki Mohamed, « De la modélisation à l'exploitation des documents à structures multiples », Thèse de Doctorat de l'Université Paul Sabatier. - Toulouse, France , 2008.

[Oh,Il-Seok et al., 1999] Oh,Il-Seok, Lee J., Suen C., Analysis of Class Separation and Combination of Class Dependent Features for Handwriting Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 10, October 1999, 1089-1094.

[Portier P.E., 2010] Portier Pierre-Edouard « Construction des Documents Multistructurés dans le Contexte des Humanités Numériques », Thèse de Doctorat de l'INSA De Lyon France, 2010.

[Schlieder T. et al., 2002] Schlieder T., Meuss M., « Querying and Ranking XML Documents », *Special Topic Issue of the Journal of the American Society of Information Science on XML and Information Retrieval*, 2002.

[Sorlin S., 2006] Sorlin S. « Mesurer la similarité de graphes », Thèse de Doctortat de l'Université de Claude Bernard Lyon I France, 2006.

Chapitre VI - Implantation et résultats expérimentaux

Résumé du chapitre :

L'objectif de ce chapitre est la validation de notre approche. Il présente l'implantation des propositions de ce mémoire de thèse au travers du prototype MMDOCREP « Multistructured Multimedia Document REPository ». Ce prototype est constitué d'une application Java qui interagit avec le Système de Gestion de Base de Données objet relationnel Oracle 10g2. Au travers de ce prototype, nous validons notre classification et nous montrons la faisabilité des démarches de comparaison et de classification structurelle de documents multimédias à structures multiples. Nous présentons un ensemble d'expérimentations qui va permettre d'évaluer nos propositions.

Sommaire du chapitre IV

I. Introduction

Pour valider les propositions présentées dans ce mémoire de thèse, nous avons développé un outil intitulé : *MMDocRep* « *Multistructured Multimedia Document REPository* ». Cet outil permet l'intégration des documents multimédias à structures multiples dans le cadre d'un entrepôt de documents. Nous détaillons dans ce chapitre, la mise en œuvre de l'approche de classification structurelle non-supervisée, décrite dans le chapitre précédent (Cf. chapitre III section IV.2, page 113).

Globalement, notre prototype permet :

- la création d'un entrepôt de documents multimédias à structures multiples,
- la classification structurelle non-supervisée de ces documents. Celle-ci permet d'associer la structure d'un document à sa classe structurelle.

MMDocRep est basé sur une architecture client-serveur (figure IV.1). Il repose d'une part, sur le Système de Gestion de Base de Données Oracle 11g2 pour le stockage des documents (structures et contenus), d'autre part sur une application développée en Java. Celle-ci utilise des interfaces graphiques permettant la communication et l'interaction avec l'utilisateur afin de faciliter l'exploitation des différents traitements : de l'alimentation de l'entrepôt de documents à la visualisation des résultats.

Le serveur de données communique avec l'application Java via l'interface API (*Application Programming Interface*) JDBC (*Java DataBase Connectivity*) qui assure la connexion avec une base de données et supporte les fonctionnalités de base du langage SQL (*Structured Query Language*). L'application Java permet d'exécuter des scripts SQL générés à partir d'un générateur de scripts SQL (figue IV.1).

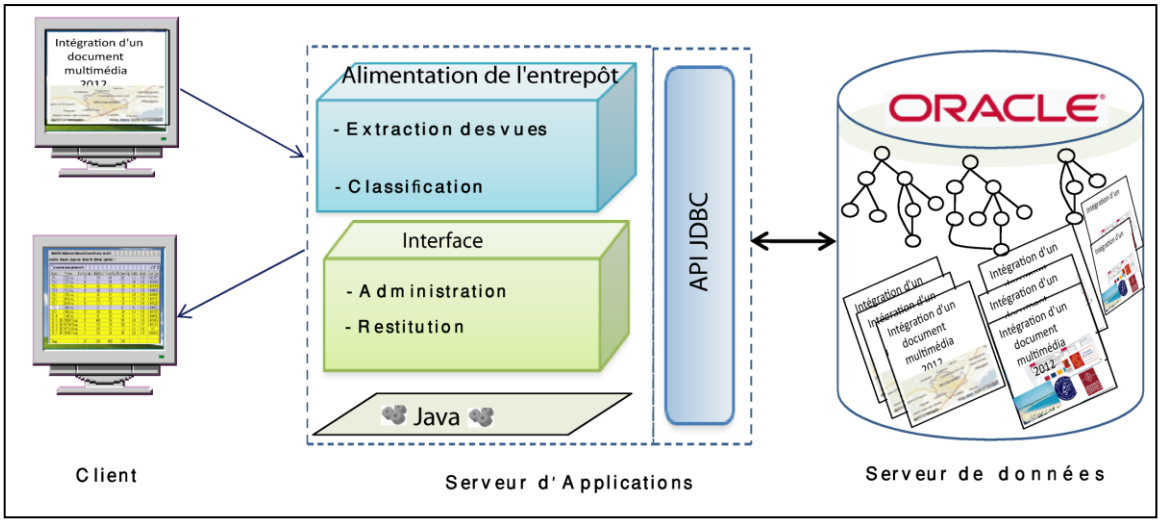


Figure IV.1 - Architecture globale de *MMDocRep*

Pour l'alimentation de l'entrepôt de documents, nous avons proposé l'algorithme *AlimenterDW()* (Cf. Annexe page 183) qui permet d'intégrer automatiquement un ensemble de documents. La partie Interface n'est pas l'objet de cette thèse, elle a été développée dans [Djemal K., 2010].

II. Alimentation de l'entrepôt documentaire

Le but du processus d'intégration (Cf. figure IV.2) est d'ajouter des documents dans l'entrepôt documentaire. Ce qui consiste à insérer le contenu et la vue spécifique du document (les étapes 3 et 4) en rattachant chaque granule (partie qui présente une information cohérente) à un fragment d'une vue générique bien déterminée. Avant qu'il ne soit inséré, le contenu du document doit être séparé de sa vue (étape 1). Dans le processus d'intégration de l'entrepôt, la prise en compte de la vue spécifique d'un document (étape 2) est une phase assez importante car c'est elle qui détermine le rattachement de la vue spécifique à une vue générique. Ainsi, la flexibilité dans l'alimentation de notre entrepôt est assurée à travers la souplesse de ces étapes d'insertion et de transformation de vues.

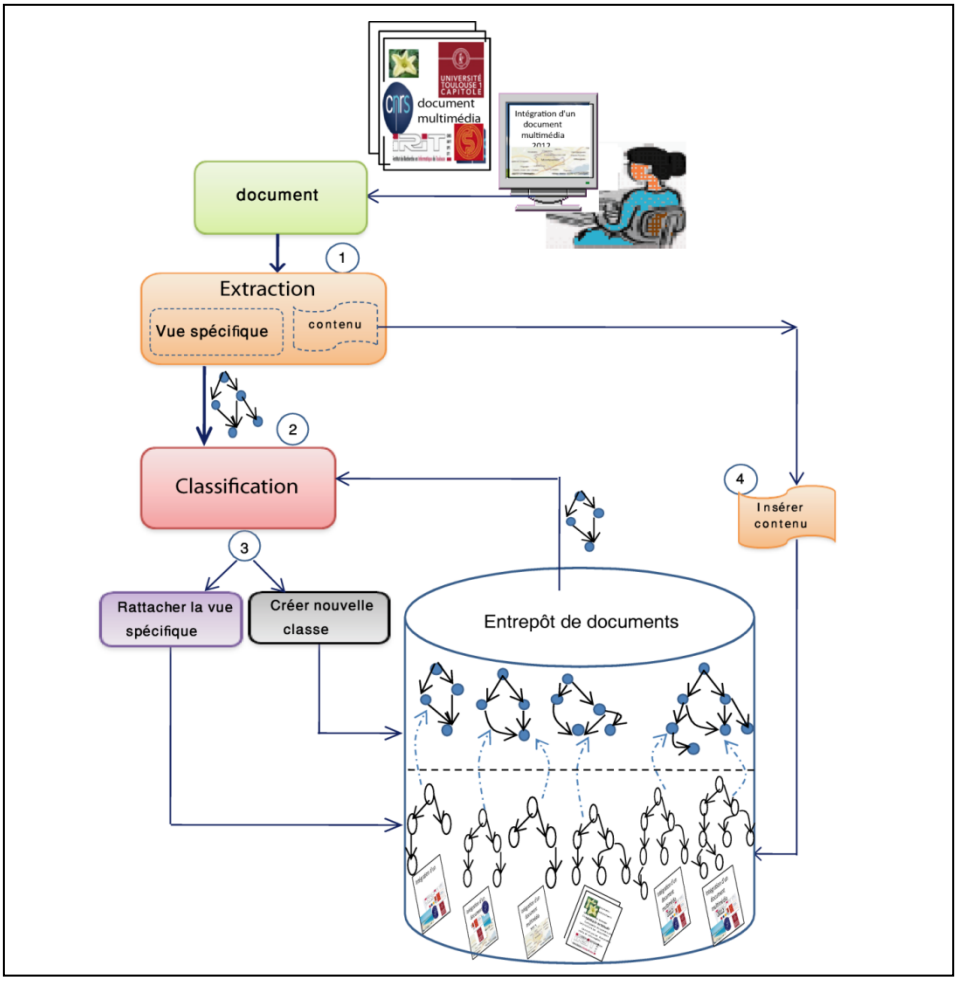


Figure IV.2 - Alimentation de l'entrepôt de documents

Le processus d'intégration de documents permet la construction incrémentale de l'entrepôt de documents suite à l'ajout de nouveaux documents issus de sources différentes et hétérogènes (Cf. Annexe, page 189). Il est composé de deux sous-modules : (1) l'extraction de la vue spécifique du document à intégrer, (2) classification structurelle.

II.1. Extraction de la vue spécifique d'un document

L'alimentation de l'entrepôt passe par un processus d'extraction de la structure du document. Comme nous l'avons indiqué (Cf. chapitre I, section IV page 31), les approches de parseurs basées sur la norme DOM « *Document Object Model* » imposent de représenter la globalité du document en mémoire sous forme d'arbre. Cela présente un inconvénient lorsqu'il s'agit de traiter des documents de grande taille.

La norme SAX « *Simple API for XML* » utilise *des événements* pour traiter les documents XML. L'avantage de cette approche est qu'on peut générer seul les éléments dont on a besoin sans être obligé à construire en mémoire l'intégralité de la structure du document manipulé. L'API SAX est basée sur une approche événementielle qui permet de réagir à un événement comme le début et la fin d'un document, le début et la fin d'un élément structurel, etc (Cf. Annexe page 185). Ces événements permettent de piloter un document XML en réalisant le traitement approprié à chaque événement. Ensuite, les résultats peuvent être renvoyés à l'application utilisant cette API.

Pour extraire la vue spécifique d'un document XML, nous avons développé deux parseurs en Java basés sur la norme SAX. Le premier renvoie toutes les balises rencontrées sur le document. Ensuite ces balises sont interceptées, par un deuxième parseur qui permet de les analyser, les filtrer et faire des transformations afin d'obtenir la vue représentant le document. La vue *Rep_d* (exemple chapitre III, figure 18, page 117) ainsi obtenue est utilisée comme représentant d'un document, dans notre processus de classification.

II.2. Classification

Pour valider notre démarche de classification (Cf. chapitre III, section IV.2 page 113), nous avons utilisé *MMDocRep* au travers un ensemble d'expérimentations. Nous rappelons que cette approche est basée sur une *mesure de similarité structurelle* que nous avons proposée (Cf. chapitre III, section III.4 page 108).

La classification d'un document, représenté par *Rep_d* (étape précédente), passe par un processus qui permet de comparer celui-ci avec l'ensemble des vues génériques de l'entrepôt et rattacher ensuite la structure spécifique de ce document à la classe la plus similaire :

- *FiltrerVueGenCand()* : qui permet de sélectionner les vues génériques, de l'entrepôt documentaire, candidates à la comparaison (Cf. Annexe page 181),
- *TransformationGraph() :* qui permet de transformer un graphe en un autre (Cf. Annexe page 182),
- *Ponderation()* : qui permet de pondérer les graphes (Cf. Annexe page 182) représentant les vues génériques
- *Dist_graph()* : qui permet de calculer la distance d'alignement entre deux vues (Cf. Annexe page 184),
- *Separat()* : qui permet de vérifier la séparation des classes (Cf. page 185),

Nous décrivons dans la section suivante les expérimentations que nous avons menées dans ce travail et nous discutons les résultats obtenus.

III. Expérimentations

Nous rappelons que pour chaque document à intégrer, une vue spécifique est générée à partir de laquelle un représentant générique de celle-ci est crée. Ce représentant doit être comparé avec l'ensemble des graphes représentant les vues génériques (les classes) existantes de l'entrepôt documentaire. Cette comparaison qui se traduit par la recherche d'isomorphisme de sous-graphes (problème combinatoire) engendre un problème lié au temps de calculs de nos algorithmes de comparaison. Pour optimiser ce temps de calculs, nous avons proposé un processus de f*iltrage* qui permet de réduire l'espace de comparaison (Cf. chapitre III, section IV.2.2 page 121) lors de l'exploration des vues génériques susceptibles d'être similaires au document à intégrer sans pour autant perdre la qualité des classes générées.

Cette section est composée de deux parties :

- dans la première partie, nous évaluons l'impact du *sous-processus de filtrage* à la fois sur la qualité de la classification et sur les performances de nos algorithmes de comparaison en termes de temps de réponse,
- dans la deuxième partie, nous étudions l'influence du *seuil de similarité* sur la qualité des classes générées par notre processus de classification automatique.

III.1. Evaluation de l'impact du sous-processus de filtrage

III.1.1. Conditions expérimentales

Corpus et classification de référence

Nous avons vu (Cf. chapitre II, section III.4.3 page 76) que pour évaluer les résultats d'une classification automatique, une des solutions est de comparer les résultats avec une classification de référence (préexistante) ou d'utiliser un corpus de référence [Yosr N. et Sinaoui, 2009]. Afin que nous puissions faire une classification manuelle, qui construira la *classification de référence*, nous avons utilisé un corpus relativement petit composé de 207 documents multimédias. Ce corpus est extrait aléatoirement du corpus *INEX 2007* et d'un corpus composé de notices descriptives de livres au format XML issues de la bibliothèque de l'Université Toulouse 1 Capitole (tableau IV.1).

Nombre de documents	**207**
Nombre total de nœuds	9862
Nombre total d'éléments	5381
Nombre total d'attributs	4481
Nombre moyen de nœuds/*Vsp*	47.89
Nombre moyen de chemins /*Vsp*	17.89
Profondeur moyenne /*Vsp*	5.76

Tableau IV.1 - Description du corpus

La classification manuelle a été établie sur la base d'un seuil de similarité de 80%. Nous avons obtenu 14 classes de documents. Nous présentons dans le tableau IV.2 pour chacune des classes, le nombre de vues spécifiques qui leur sont rattachées :

Classes	Nb *Vsp* / Classe
$C_{1\text{-}Ref}$	46
$C_{2\text{-}Ref}$	19
$C_{3\text{-}Ref}$	18
$C_{4\text{-}Ref}$	39
$C_{5\text{-}Ref}$	7
$C_{6\text{-}Ref}$	14
$C_{7\text{-}Ref}$	13
$C_{8\text{-}Ref}$	5
$C_{9\text{-}Ref}$	13
$C_{10\text{-}Ref}$	5
$C_{11\text{-}Ref}$	17
$C_{12\text{-}Ref}$	5
$C_{13\text{-}Ref}$	2
$C_{14\text{-}Ref}$	4

Tableau IV.2 - Résultats obtenus par classification manuelle : *classif_Ref*

Dans les expériences qui suivent, nous utiliserons le terme de *classif_Ref* pour la classification manuelle de référence.

Description des mesures utilisées

Une classification de qualité est une classification qui correspond au mieux à la *classification de référence*. Pour valider les résultats d'une classification, on peut utiliser le critère interne qui permet de mesurer l'écart entre la structure engendrée et les données [Genane Y., 2004]. Les classes générées automatiquement doivent être identiques aux classes de *classif_Ref*. Aussi, nous discuterons pour chacune des expériences, du *nombre de classes obtenues*, le contenu de ces classes et donc du *nombre de vues spécifiques mal classées*. De plus, les classes construites doivent être les plus homogènes possibles. Cette homogénéité peut se mesurer grâce à la *similarité moyenne intra-classe et l'écart-type intra-classe*. Nous utilisons également le rappel et précision pour comparer les classifications.

Ainsi, les tableaux de chacune de nos expériences feront apparaître les mesures suivantes :

- *Nb_ Vsp* : le nombre de vues spécifiques rattachées,
- *Nb_ Elt* : le nombre de nœuds de type éléments,
- *Nb_Att* : le nombre de nœuds de type attributs,
- *Nb_Chem* : le nombre de chemins,
- *ProfMy* : la moyenne des profondeurs des vues spécifiques rattachées,
- *SimMy* : la similarité moyenne intra-classe,
- *Ecart_Type* : l'écart type intra-classe,
- *Mal classées* : le nombre de vues mal classées.

III.1.2. Impact du filtrage sur la qualité des classes obtenues

Afin d'étudier l'impact du seuil de *filtrage* à la fois sur la qualité des classes obtenues et sur le temps de réponse de nos algorithmes, nous avons mené trois séries de tests avec des seuils de *filtrage* différents. Nous fixons le **seuil de similarité à 80%** et nous faisons varier le *seuil de filtrage* à 70%, 66% et 64%. Nous avons aussi effectué une classification *sans filtrage*.

Les tableaux IV.3, IV.4 et IV.5 présentent les résultats des classifications *classif_70, classif_66 et classif_64*.

Seuil de filtrage à 70% : classif_70

Les 207 documents du corpus sont regroupés en 15 classes.

La comparaison avec *classif_Ref* est donnée dans la figure IV.3. Nous avons donné pour chacune des classes sa correspondance avec les classes de *classif_Ref*.
Nous avons un taux de vues mal classées de 7.24%. Ce taux s'explique en partie par la création d'une classe supplémentaire, la classe $C_{3\text{-}70}$. Après examen de cette classe, nous avons constaté que la création de cette classe est due à une erreur de filtrage. Les cinq vues rattachées à $C_{3\text{-}70}$ sont essentiellement : trois vues structurellement proches à la classe $C_{3\text{-}Ref}$ et deux vues structurellement proches à la classe $C_{4\text{-}Ref}$.
L'examen des autres classes et des vues spécifiques qui leur sont rattachées a montré que deux vues de la classe $C_{1\text{-}70}$ sont structurellement proches à $C_{4\text{-}Ref}$. Une vue de la classe $C_{2\text{-}70}$ est structurellement proche à classe $C_{4\text{-}Ref}$. Une vue de la classe $C_{9\text{-}70}$ est structurellement proche à $C_{4\text{-}Ref}$ et une vue de la même classe est structurellement proche à $C_{9\text{-}Ref}$. Quatre vues de La classe $C_{14\text{-}70}$ sont structurellement proches à $C_{12\text{-}Ref}$.
Ces erreurs sont dues au fait que la bonne classe à laquelle devrait être rattachée la vue spécifique, lors de l'intégration du document qu'elle représente, n'a pas été filtrée.

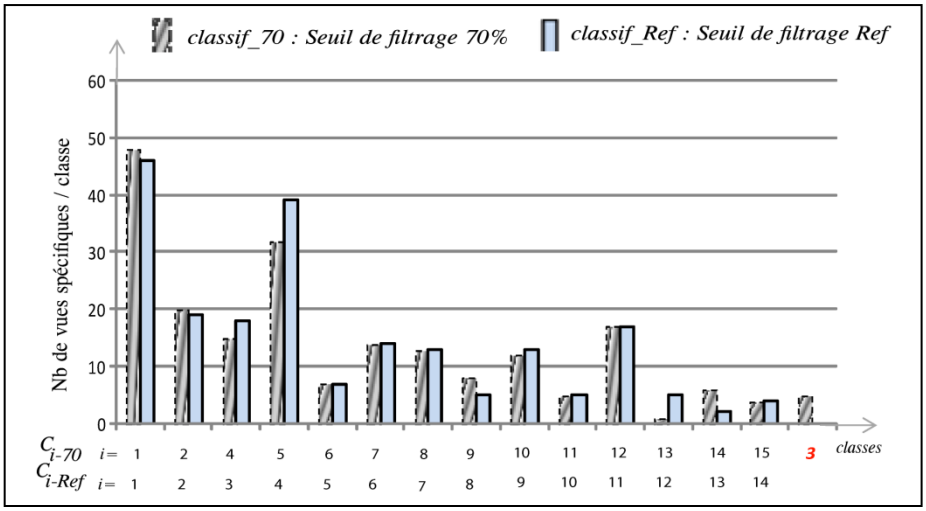


Figure IV.3 - Représentation graphique des résultats de *classif_70* et *classif_Ref*

La qualité de la classification est discutée aussi en examinant le tableau IV.3.

Seuil Sim=0.80 et Seuil Filtrage=0.70 : Le nombre de classes générées est 15

Classes	Nb Vsp/Clas	Nb Elt/Clas	Nb Att/Clas	Nb Chem/...	ProfMy	SimMy	Ecart-Type	Mal classées
C1-70	48	601	680	446	5,06	0,97	0,02	2
C2-70	20	285	307	208	5,10	0,97	0,02	1
C3-70	5	79	72	60	5,20	0,95	0,04	5
C4-70	15	216	210	166	6,07	0,97	0,01	0
C5-70	32	853	916	476	6,12	0,96	0,18	0
C6-70	7	94	89	77	6,29	0,98	0,01	0
C7-70	14	295	315	171	5,36	0,98	0,01	0
C8-70	13	217	176	155	7,00	0,98	0,01	0
C9-70	8	116	107	77	4,62	0,96	0,17	3
C10-70	12	323	286	174	6,50	0,98	0,01	0
C11-70	5	89	107	58	5,40	0,95	0,02	0
C12-70	17	1401	769	991	6,18	0,98	0,01	0
C13-70	1	70	38	58	6,00	1,00	0,00	0
C14-70	6	417	225	345	6,00	0,97	0,02	4
C15-70	4	325	184	241	6,25	0,95	0,01	0
Totaux	207	5381	4481	3703				15

Tableau IV.3 - Résultats de la classification : *classif_70*

Nous constatons (tableaux IV.3) que les classes $C_{1\text{-}70}$, $C_{2\text{-}70}$, $C_{3\text{-}70}$, $C_{9\text{-}70}$ et $C_{14\text{-}70}$ qui représentent des vues spécifiques mal classées sont plus hétérogènes par rapport aux autres classes : *écart-type intra-classe* plus élevé.

Seuil de filtrage à 66% : classif_66

Les 207 documents du corpus sont regroupés en 14 classes.

La comparaison avec c*lassif_Ref* est donnée dans la figure IV.4. Nous avons donné pour chacune des classes sa correspondance avec les classes de *classif_Ref*.
Nous avons un taux de vues mal classées de 1.93%**.** Ce taux d'erreurs, moins important que celui engendré par *classif_70*, s'explique par le fait que quatre vues spécifiques sont mal classées. En effet, deux vues spécifiques de la classe $C_{1\text{-}70}$ sont structurellement proches à la classe $C_{4\text{-}Ref}$ une vue spécifique de $C_{2\text{-}70}$ est structurellement proche à la classe $C_{4\text{-}Ref}$ et une vue spécifique de $C_{8\text{-}70}$ est structurellement proche à la classe $C_{4\text{-}Ref}$.

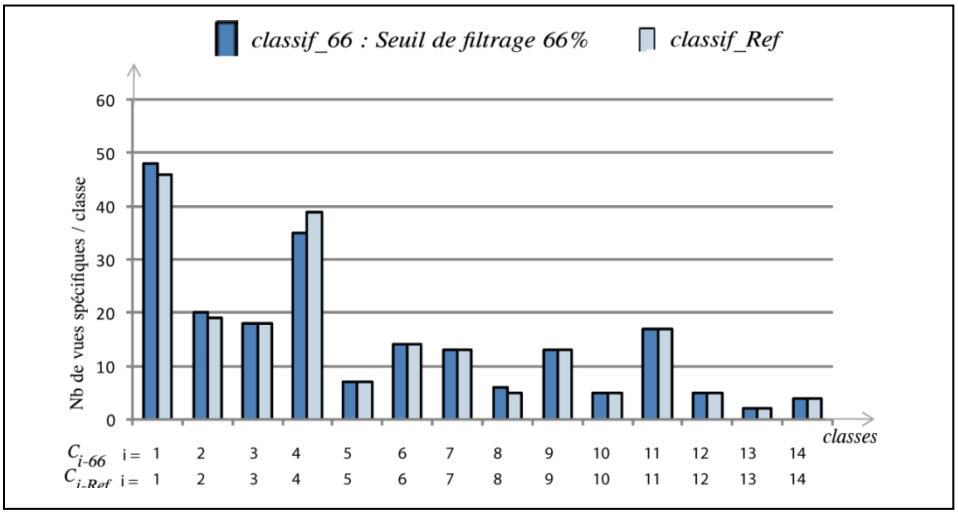


Figure IV.4 - Représentation graphique des résultats de *classif_66* et *classif_Ref*

La qualité de la classification est discutée aussi en examinant le tableau IV.4.

Seuil Sim=0.80 et Seuil Filtrage=0.66 : Le nombre de classes générées est 14

Classes	Nb Vsp/Clas	Nb Elt/Clas	Nb Att/Clas	Nb Chem/...	ProfMy	SimMy	Ecart-Type	Mal classées
C1-66	48	601	680	446	5,06	0,97	0,02	2
C2-66	20	285	307	208	5,10	0,97	0,02	1
C3-66	18	262	253	198	5,94	0,97	0,01	0
C4-66	35	903	949	510	6,06	0,98	0,02	0
C5-66	7	94	89	77	6,29	0,98	0,01	0
C6-66	14	295	315	171	5,36	0,98	0,01	0
C7-66	13	217	176	155	7,00	0,98	0,01	0
C8-66	6	77	85	56	4,50	0,97	0,04	1
C9-66	13	345	304	189	6,46	0,98	0,01	0
C10-66	5	89	107	58	5,40	0,95	0,02	0
C11-66	17	1401	769	991	6,18	0,98	0,01	0
C12-66	5	353	193	288	6,00	0,97	0,02	0
C13-66	2	134	70	115	6,00	0,99	0,01	0
C14-66	4	325	184	241	6,25	0,95	0,01	0
Totaux	207	5381	4481	3703				4

Tableau IV.4 - Résultats de la classification : *classif_66*

Nous constatons (tableaux IV.4) la disparition de la classe créée par erreur de filtrage (*classif_70*) et que les classes $C_{8\text{-}66}$ et $C_{13\text{-}66}$ sont plus homogènes par rapport à leur classes homologues respectivement $C_{9\text{-}70}$ *et* $C_{14\text{-}70}$ (tableau IV.3) : *écart-type intra-classe* plus optimum.

Seuil de filtrage à 64% : classif_64

Les 207 documents du corpus sont regroupés en 14 classes.

La comparaison avec c*lassif_Ref* est donnée dans la figure IV.5. Nous avons donné pour chacune des classes sa correspondance avec les classes de *classif_Ref.*
Après examens de chacune des classes et des vues spécifiques qu'elles représentent, nous avons constaté que les résultats obtenus sont exactement les mêmes que ceux de *classif_Ref* (tableau IV.2). Par conséquent, toutes les vues spécifiques intégrées sont rattachées à la bonne classe.

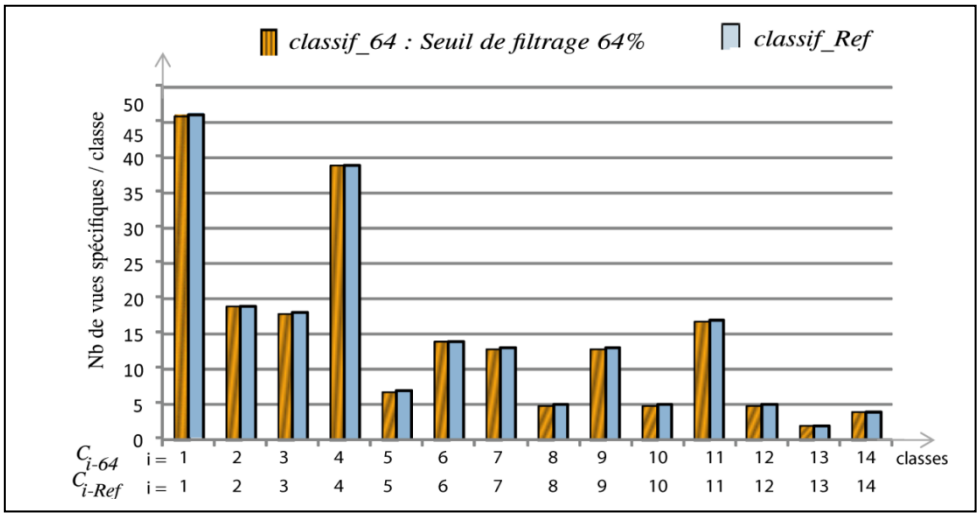


Figure IV.5 - Représentation graphique des résultats de *classif_64* et *classif_Ref*

La qualité de la classification est discutée aussi en examinant le tableau IV.5.

Seuil Sim=0.80 et Seuil Filtrage=0.64 : Le nombre de classes générées est 14

Classes	Nb Vsp/Clas	Nb Elt/Clas	Nb Att/Clas	Nb Chem/...	ProfMy	SimMy	Ecart-Type	Mal classées
C1-64	46	570	661	426	5,00	0,98	0,02	0
C2-64	19	281	281	213	5,05	0,98	0,01	0
C3-64	18	262	253	198	5,94	0,97	0,01	0
C4-64	39	948	1002	537	6,00	0,98	0,00	0
C5-64	7	94	89	77	6,29	0,98	0,01	0
C6-64	14	295	315	171	5,36	0,98	0,01	0
C7-64	13	217	176	155	7,00	0,98	0,01	0
C8-64	5	67	77	44	4,40	0,97	0,00	0
C9-64	13	345	304	189	6,46	0,98	0,01	0
C10-64	5	89	107	58	5,40	0,95	0,02	0
C11-64	17	1401	769	991	6,18	0,98	0,01	0
C12-64	5	353	193	288	6,00	0,97	0,02	0
C13-64	2	134	70	115	6,00	0,99	0,01	0
C14-64	4	325	184	241	6,25	0,95	0,01	0
Totaux	207	5381	4481	3703				0

Tableau IV.5 - Résultats de la classification : *classif_64*

Nous constatons que les trois classes qui ont subi des modifications : les lignes 1, 2 et 8 du tableau IV.5 sont plus homogènes par rapport à *classif_66* (tableau IV.4) : l'écart-type intra-classe est plus optimum.

Au final, après examen des classes générées par *classif_64* et *classif_Ref* et des vues spécifiques qu'elles représentent, nous avons remarqué que nous obtenions les mêmes classes qu'avec la *classification de référence* : *classif_Ref* (classification manuelle : tableau IV.2) :

$\forall\ i \in [1,14]$; les classes $C_{i\text{-}Ref}$ et $C_{i\text{-}64}$ représentent les mêmes vues spécifiques.

La figure IV.6 représente une comparaison graphique des résultats des quatre classifications : *classif_70, classif_66, classif_64* et *classif_Ref.*

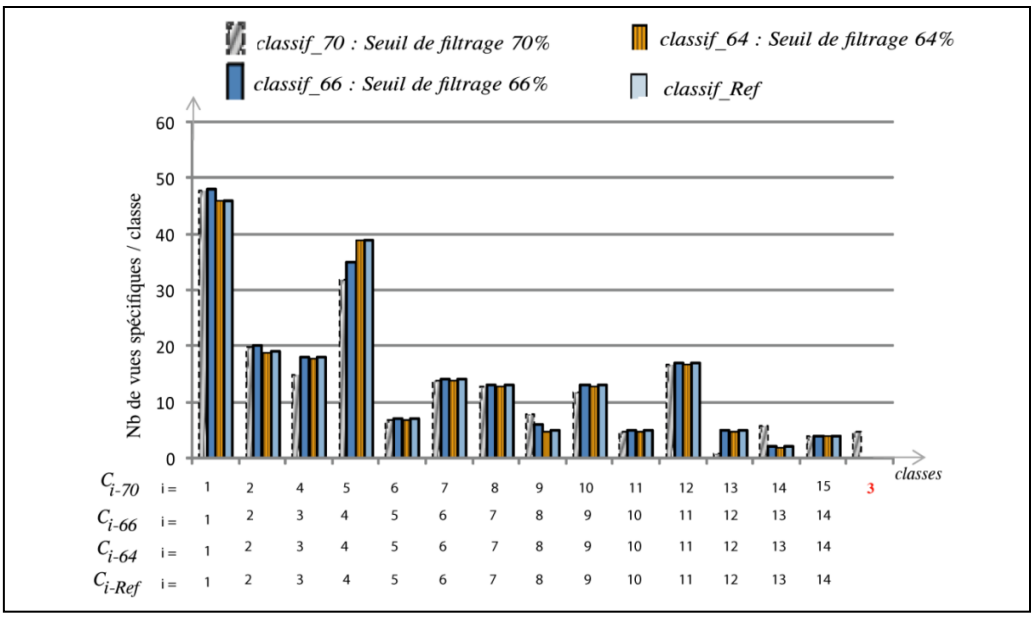


Figure IV.6 - Représentation graphique des résultats de *classif_70, classif_66, classif_64* et *classif_Ref*

San seuil de filtrage : classif_SansFiltr

Les 207 documents du corpus sont regroupés en 14 classes.

Nous avons donné pour chacune des classes sa correspondance avec les classes de *classif_Ref*. Après examens de ces classes et des vues spécifiques qu'elles représentent, nous avons constaté que les résultats obtenus sont exactement les mêmes que ceux de *classif_Ref*.

Utilisation du rappel et précision pour comparer les classifications

Nous avons vu (Cf. chapitre II, section III.4.4 page 76), que la précision permet d'évaluer l'homogénéité des classes tandis que le rappel évalue l'exhaustivité des contenus des classes. Un classifieur est plus efficace quand la précision et le rappel sont proches de 1.

Dans le tableau IV.6, nous rappelons les formules [19], [20], [21] et [22] (Cf. chapitre II, section III.4.4 page 77) respectivement de la précision P_i d'une classe c_i, de la précision P de la classification, du rappel R_i d'une la classe c_i et du rappel R de la classification :

Précision d'une classe c_i	$P_i = \frac{Nombre\ de\ documents\ correctement\ attribués\ à\ la\ classe\ c_i}{Nombre\ de\ documents\ attribués\ à\ c_i}$
Précision de la classification	$P = \frac{\sum_{i=1}^{n} P_i}{n}$ où n est le nombre de classes
Rappel d'une classe c_i	$R_i = \frac{Nombre\ de\ documents\ correctement\ attribués\ à\ la\ classe\ c_i}{Nombre\ de\ documents\ \in\ c_i}$
Rappel de la classification	$R = \frac{\sum_{i=1}^{n} R_i}{n}$ où n est le nombre de classes

Tableau IV.6 - Précision et rappel d'une classification

Dans le tableau IV.7, nous présentons les rappels et précision des classes de trois séries de tests c*lassif_70*, c*lassif_66* et c*lassif_64* que nous avons menées.

	Classif_70	
Classes	Rappel	Précision
$C_{1\text{-}70}$	1	0.96
$C_{2\text{-}70}$	0.91	1
$C_{3\text{-}70}$	0	0
$C_{4\text{-}70}$	0.83	1
$C_{5\text{-}70}$	0.82	1
$C_{6\text{-}70}$	1	1
$C_{7\text{-}70}$	1	1
$C_{8\text{-}70}$	1	1
$C_{9\text{-}70}$	1	0.63
$C_{10\text{-}70}$	0.92	1

	classif_66	
Classes	Rappel	Précision
$C_{1\text{-}66}$	1	0.96
$C_{2\text{-}66}$	1	0.95
$C_{3\text{-}66}$	1	1
$C_{4\text{-}66}$	0.90	1
$C_{5\text{-}66}$	1	1
$C_{6\text{-}66}$	1	1
$C_{7\text{-}66}$	1	1
$C_{8\text{-}66}$	1	0.83
$C_{9\text{-}66}$	1	1
$C_{10\text{-}66}$	1	1

	classif_64	
Classes	Rappel	Précision
$C_{1\text{-}64}$	1	1
$C_{2\text{-}64}$	1	1
$C_{3\text{-}64}$	1	1
$C_{4\text{-}64}$	1	1
$C_{5\text{-}64}$	1	1
$C_{6\text{-}64}$	1	1
$C_{7\text{-}64}$	1	1
$C_{8\text{-}64}$	1	1
$C_{9\text{-}64}$	1	1
$C_{10\text{-}64}$	1	1

$C_{11\text{-}70}$	1	1
$C_{12\text{-}70}$	1	1
$C_{13\text{-}70}$	0.20	1
$C_{14\text{-}70}$	1	0.33
$C_{15\text{-}70}$	1	1
Totaux	R =0,91	P=0.92

$C_{11\text{-}66}$	1	1
$C_{12\text{-}66}$	1	1
$C_{13\text{-}66}$	1	1
$C_{14\text{-}66}$	1	1
	R=0.99	P=0.98

$C_{11\text{-}64}$	1	1
$C_{12\text{-}64}$	1	1
$C_{13\text{-}64}$	1	1
$C_{14\text{-}64}$	1	1
	R=1	P=1

Tableau IV.7 - Rappels et précisons de *classif_70, classif_66* et *classif_64*

Nous remarquons que la classification *classif_64* est plus efficace que les classifications *classif_70* et *classif_66* : le rappel et la précision sont égaux à 1.

Dans nos travaux antérieurs [Idarrou A. et al., 2012a], nous avons fait une étude comparative, concernant l'impact du *processus de filtrage* sur la qualité des classes générées, avec l'approche de [Mbarki M., 2008]. Nous avons montré à travers cette étude que notre *mesure de filtrage* (Cf. la formule [31] page 121) est plus efficace et que cette mesure tient compte des types (élément, attribut ou métadonnée) des éléments structurels comparés.

Nous étudions dans la section suivante l'influence du *seuil de filtrage* sur le temps de réponse de nos algorithmes.

III.1.3. Impact du filtrage sur le temps de réponse

Le temps de réponse est l'un des critères qui permettent de mesurer les performances d'un algorithme. Nous étudions, dans cette section, l'impact du filtrage sur le temps de réponse de nos algorithmes de comparaison. Le tableau IV.8 présente le temps moyen (en secondes), du sous-processus de *comparaison* des vues sur lequel repose notre approche de classification structurelle, en fonction du nombre de chemins qu'elles contiennent. Nous avons mesuré les temps, pour chacune des quatre tests, en regroupant les vues spécifiques en quatre intervalles de nombres de chemins. Ce temps peut varier d'une machine à l'autre. Cela dépend des performances en termes de processeur, mémoire, etc de la machine utilisée.

Nombre de chemins	1 à 5	6 à 10	11 à 15	16 à 20
Temps avec filtrage (70%)	0.30	0.89	1.42	1.76
Temps avec filtrage (66%)	0.36	0.99	1.57	1.93
Temps avec filtrage (64%)	0.42	1.10	1.71	2.11
Temps sans filtrage	0.49	1.25	1.92	2.40

Tableau IV.8 - Temps moyen de réponse par tranche de chemin

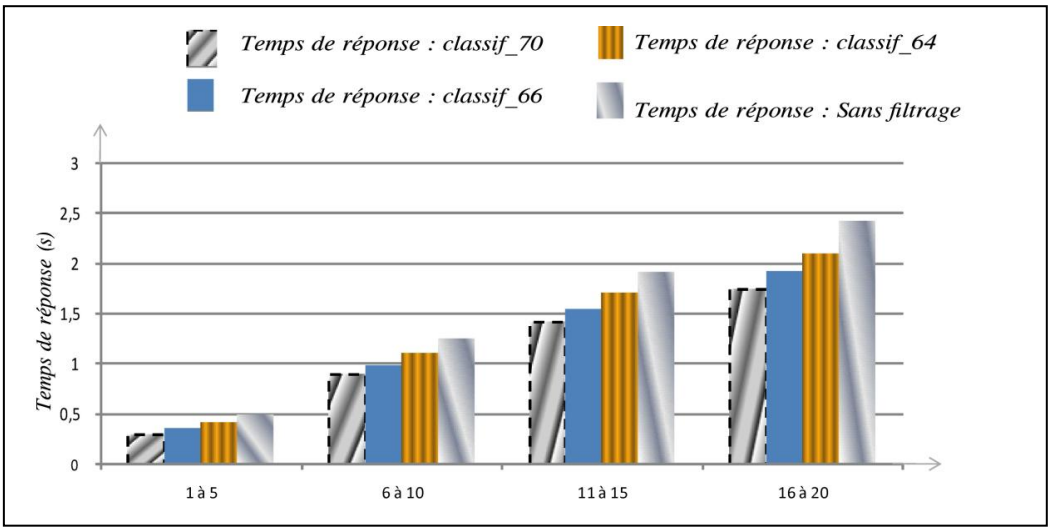


Figure IV.7 - Temps moyen (en s) de réponse avec et sans filtrage par tranche de chemins

La figure IV.7 représente, pour chaque tranche de chemin, les temps de réponse pour les quatre séries de tests. Nous remarquons que le gain de temps augmente avec le nombre de chemins (exemple tranches 16-20). Le gain le plus notable est observé pour les tranches supérieures à 16 chemins avec environ une diminution du temps de 26.68% entre *classif_70* et c*lassif_SansFiltr* et de 12.08% entre *classif_64* et c*lassif_SansFiltr*.

III.1.4. Bilan et synthèse

Le tableau IV.9 présente une synthétise des résultats des quatre séries de tests. Nous présentons dans ce tableau, pour chaque série de tests, le nombre de classes obtenues, la moyenne de vues spécifique par classe, la moyennes des similarités moyennes intra-classe, la moyenne des écart-types intra-classe et le pourcentage des vues spécifiques mal classées.

	classif_70	***classif_66***	***classif_64***	***classif_Sans Filtr***
Nombre de classes générées	15	14	14	14
Moyenne de vues spécifique par classe	13.80	14.78	14.78	14.78
Moyenne des similarités moyennes	0.97	0.97	0.97	0.97
Moyennes des écart-types	0.03	0.19	0.01	0.01
Pourcentage des vues spécifiques mal classées (%)	7.24	1.93	0	0
Rappels et précisions des quatre classifieurs	*R*=0.91	*R*=0.99	*R*=1	*R*=1
	P=0.92	*P*=0.98	*P*=1	*P*=1

Tableau IV.9 - Synthétise des résultats de *classif_70*, *classif_66*, *classif_64* et *classif_SansFiltr*

Le *filtrage* permet de restreindre l'espace de comparaison, il joue un rôle important dans l'optimisation de temps de réponse de nos algorithmes. En revanche son impact sur la

qualité de classification est non négligeable. Dans les quatre tests que nous avons effectués, nous avons noté qu'avec un *seuil de filtrage* de 70% le taux des vues spécifiques mal classées est de 7.24%. Ce taux diminue, avec *un seuil de filtrage* de 66% : il est égal à 1.93%. Il devient nul, à partir d'un seuil de filtrage de 64%.

Il suffit donc de trouver un compromis entre la qualité des classes et la performance des algorithmes de comparaison, en termes de temps de réponse. Nous remarquons, à travers les expériences effectuées dans [Idarrou A. et al., 2012a] et celles menées dans ces travaux, qu'un *seuil de filtrage* de 64% donne de bons résultats : un gain remarquable de temps de calcul (surtout pour comparer des graphes ayant un nombre de chemins important) sans pour autant altérer la qualité de classification.

Après avoir étudié l'impact du sous-processus de filtrage sur à la fois la qualité de la classification et sur le temps de réponse de nos algorithmes, nous étudions dans la section suivante l'influence du seuil de similarité sur les classes obtenues.

III.2. Influence du seuil de similarité sur la classification

III.2.1. Description de l'expérience

Corpus utilisé

Dans cette deuxième partie de nos expérimentations, nous étudions l'impact du *seuil de similarité* sur la qualité des classes générées par notre processus de classification. Pour cela, nous avons mené quatre séries de tests sur un même corpus composé de 1606 documents extraits de manière aléatoire du corpus *INEX 2007* et d'un corpus composé de notices descriptives de livres au format XML issues de la bibliothèque de l'Université Toulouse 1 Capitole (tableau IV.10).

Nombre de documents	**1606**
Nombre total de nœuds	38138
Nombre total d'éléments	21814
Nombre total d'attributs	16324
Nombre moyen de nœuds / *Vsp*	23.75
Nombre moyen de chemins / *Vsp*	8.86
Profondeur moyenne / *Vsp*	6.06

Tableau IV.10 - Description du corpus pour la seconde expérimentation

Dans les quatre séries de tests, nous fixons le *seuil de filtrage* à 64% et nous faisons varier le *seuil de similarité* à 78%, 80%, 82% et 83%.

Description des mesures utilisées

En l'absence d'une classification de référence, nous utilisons le critère interne pour discuter la validité des classes obtenues par chacune des classifications. Aussi, nous discuterons pour chacune des expériences, du *nombre de classes obtenu*, le contenu de ces classes. Nous discutons également l'homogénéité des classes construites. Cette homogénéité peut se mesurer grâce à *l'écart-type intra-classe.*

Ainsi, les tableaux de chacune de nos expériences feront apparaître les mesures suivantes :

- *Nb_ Vsp* : le nombre de vues spécifiques rattachées,
- *Nb_ Elt* : le nombre de nœuds de type éléments,
- *Nb_Att* : le nombre de nœuds de type attributs,
- *Nb_Chem : le nombre de chemins,*
- *ProfMy : la moyenne des profondeurs des vues spécifiques rattachées,*
- *SimMy : la similarité moyenne intra-classe,*
- *Ecart_Type : l'écart type intra-classe.*

Seuil de similarité à 78% : classif78

Les 1606 documents du corpus sont regroupés en 40 classes (avec une distance minimale inter-classe de 22%).
Le tableau IV.11 ci-dessous présente les résultats de la classification c*lassif78.*

Seuil Sim=0.78 : Le nombre de classes générées est 40

Classes	Nb Vsp/clas	Nb Elt/Clas	Nb Att/Clas	NbChem/Clas	ProfMy	SimMy	Ecart-Type
C1-78	177	2285	1565	1431	6,61	0,98	0,00
C2-78	34	417	312	310	5,47	0,86	0,02
C3-78	186	2252	1970	1770	5,41	0,86	0,01
C4-78	21	263	208	193	5,71	0,97	0,02
C5-78	30	344	277	246	5,23	0,88	0,06
C6-78	20	197	170	135	5,10	0,95	0,01
C7-78	22	232	204	218	5,18	0,83	0,03
C8-78	23	668	80	98	4,04	0,98	0,01
C9-78	85	828	686	607	5,00	0,98	0,00
C10-78	40	473	467	364	4,48	0,95	0,03
C11-78	105	1938	541	583	4,95	0,98	0,02
C12-78	67	553	503	419	3,93	0,98	0,01
C13-78	13	197	122	121	5,38	0,96	0,02
C14-78	42	536	548	518	5,02	0,95	0,03
C15-78	56	563	681	395	4,16	0,97	0,02
C16-78	30	339	315	251	4,77	0,83	0,03
C17-78	18	298	169	181	6,33	0,89	0,02
C18-78	6	79	64	70	4,33	0,91	0,01
C19-78	33	496	314	327	6,58	0,95	0,02
C20-78	29	260	263	194	4,17	0,98	0,02
C21-78	26	234	244	174	4,08	0,98	0,01
C22-78	18	232	193	186	5,56	0,95	0,01
C23-78	34	433	394	248	5,15	0,99	0,01
C24-78	30	291	248	216	5,00	0,99	0,02
C25-78	29	263	266	189	4,00	0,98	0,01
C26-78	7	70	63	63	4,00	0,98	0,01
C27-78	13	135	146	86	4,15	0,99	0,00
C28-78	22	226	248	156	4,18	0,98	0,01
C29-78	8	98	93	53	4,00	0,98	0,01
C30-78	29	309	336	220	4,14	0,98	0,00
C31-78	42	445	583	305	4,10	0,98	0,00
C32-78	72	828	571	508	5,04	0,96	0,02
C33-78	145	1643	1696	1273	4,56	0,96	0,02
C34-78	10	128	104	124	5,40	0,91	0,01
C35-78	12	131	124	130	4,75	0,95	0,02
C36-78	44	917	340	242	4,02	0,98	0,01
C37-78	17	1401	769	991	6,18	0,98	0,01
C38-78	5	353	193	288	6,00	0,97	0,02
C39-78	2	134	70	115	6,00	0,99	0,01
C40-78	4	325	184	241	6,25	0,95	0,01
Totaux	1606	21814	16324	14239			

Tableau IV.11 - Résultats de la classification : *classif78*

Seuil de similarité à 80% : classif80

Les 1606 documents sont groupés en 42 classes (avec une distance minimale inter-classe de 20%).

La comparaison avec *classif78* est donnée dans la figure IV.8. Nous avons donné pour chacune des classes sa correspondance avec les classes de *classif78*.

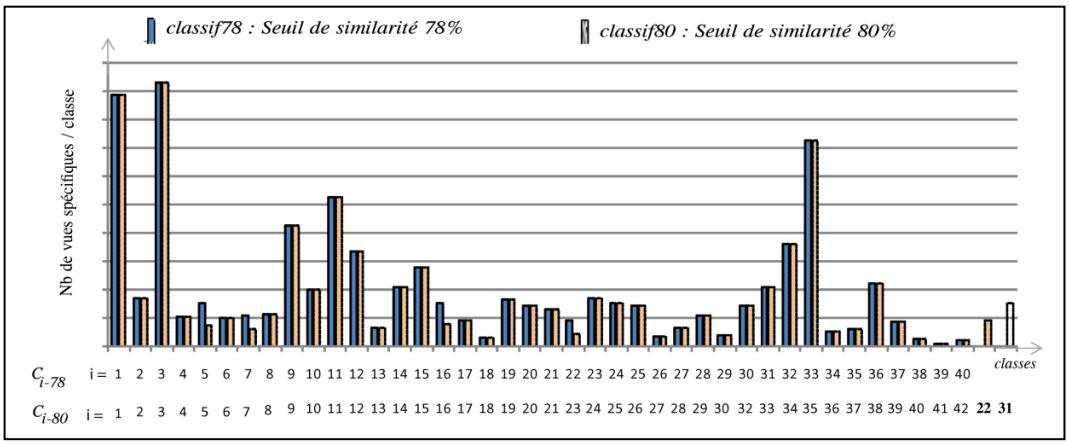


Figure IV.8 - Représentation graphique des résultats de *classif78* et *classif80*

La qualité de la classification est discutée aussi en examinant le tableau IV.12.

Seuil Sim=0.80 : Le nombre de classes générées est 42

Classes	Nb Vsp/clas	Nb Elt/Clas	Nb Att/Clas	NbChem/Clas	ProfMy	SimMy	Ecart-Type
C1-80	177	2285	1565	1431	6,61	0,98	0,00
C2-80	34	417	312	310	5,47	0,86	0,02
C3-80	186	2252	1970	1770	5,41	0,86	0,01
C4-80	21	263	208	193	5,71	0,97	0,02
C5-80	15	190	154	130	5,47	0,95	0,01
C6-80	20	197	170	135	5,10	0,95	0,01
C7-80	12	147	134	147	5,33	0,94	0,01
C8-80	23	668	80	98	4,04	0,98	0,01
C9-80	85	828	686	607	5,00	0,98	0,00
C10-80	40	473	467	364	4,48	0,95	0,03
C11-80	105	1938	541	583	4,95	0,98	0,02
C12-80	67	553	503	419	3,93	0,98	0,01
C13-80	13	197	122	121	5,38	0,96	0,02
C14-80	42	536	548	518	5,02	0,95	0,03
C15-80	56	563	681	395	4,16	0,97	0,02
C16-80	16	174	187	124	4,19	0,96	0,01
C17-80	18	298	169	181	6,33	0,89	0,02
C18-80	6	79	64	70	4,33	0,91	0,01
C19-80	33	496	314	327	6,58	0,95	0,02
C20-80	29	260	263	194	4,17	0,98	0,02
C21-80	26	234	244	174	4,08	0,98	0,01
C22-80	9	104	90	89	5,11	0,97	0,01
C23-80	18	232	193	186	5,56	0,95	0,01
C24-80	34	433	394	248	5,15	0,99	0,01
C25-80	30	291	248	216	5,00	0,98	0,02
C26-80	29	263	266	189	4,00	0,98	0,01
C27-80	7	70	63	63	4,00	0,98	0,01
C28-80	13	135	146	86	4,15	0,99	0,00
C29-80	22	226	248	156	4,18	0,98	0,01
C30-80	8	98	93	53	4,00	0,98	0,01
C31-80	30	300	231	225	5,17	0,97	0,01
C32-80	29	309	336	220	4,14	0,98	0,00
C33-80	42	445	583	305	4,10	0,98	0,00
C34-80	72	828	571	508	5,04	0,96	0,02
C35-80	145	1643	1696	1273	4,56	0,96	0,02
C36-80	10	128	104	124	5,40	0,91	0,01
C37-80	12	131	124	130	4,75	0,95	0,02
C38-80	44	917	340	242	4,02	0,98	0,01
C39-80	17	1401	769	991	6,18	0,98	0,01
C40-80	5	353	193	288	6,00	0,97	0,02
C41-80	2	134	70	115	6,00	0,99	0,01
C42-80	4	325	184	241	6,25	0,95	0,01
Totaux	1606	21814	16324	14239			

Tableau IV.12 - Résultats de la classification : *classif80*

Après examen des résultats des deux classifications c*lassif78* et c*lassif80*, nous avons constaté l'émergence de deux nouvelles classes : $C_{22\text{-}80}$ et $C_{31\text{-}80}$ (tableau IV.12). La classe $C_{22\text{-}80}$ regroupe 9 vues spécifiques : 2 vues spécifiques de la classe $C_{7\text{-}78}$ (tableau IV.11) et 7 vues spécifiques de la classe $C_{16\text{-}78}$ (tableau IV.11). La deuxième classe $C_{31\text{-}80}$ regroupe 30 vues spécifiques : 15 vues spécifiques de la classe $C_{5\text{-}78}$ (tableau IV.11), 8 vues spécifiques de la classe $C_{7\text{-}78}$ (tableau IV.11) et 7 vues spécifiques de la classe $C_{16\text{-}78}$ (tableau IV.11).

En comparaison avec les résultats de *classif78* (tableau IV.11), nous remarquons une amélioration des similarités moyennes intra-classe et une optimisation considérable de l'écart-type intra-classe des classes ayant subi des modifications : lignes 5, 7 et 16 du tableau IV.12.

Seuil de similarité à 82% : classif82

Les 1606 documents sont groupés en 43 classes (avec une distance minimale inter-classe de 18%).

La comparaison avec *classif80* est donnée dans la figure IV.9. Nous avons donné pour chacune des classes sa correspondance avec les classes de *classif80*.

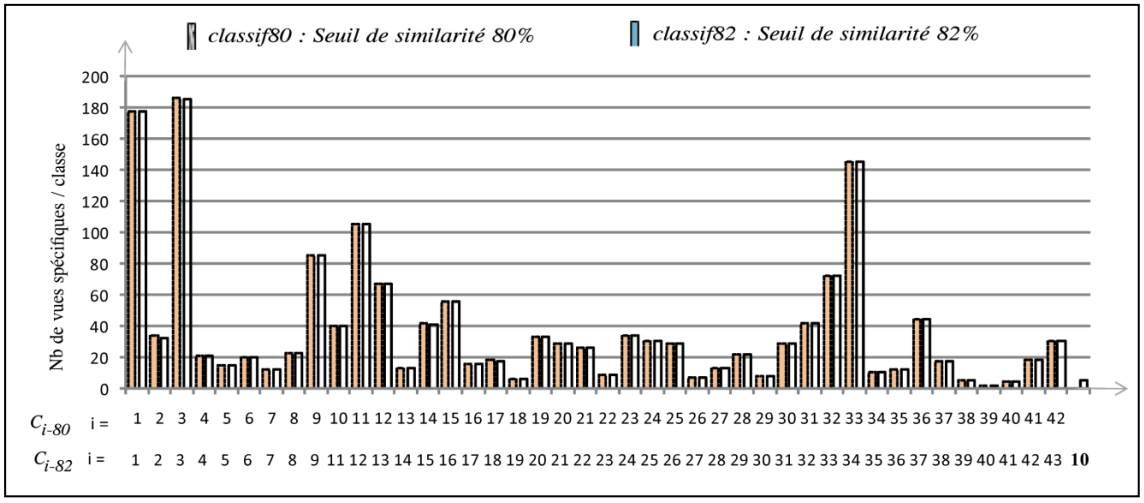


Figure IV.9 - Représentation graphique des résultats de *classif80* et *classif82*

Nous discutons les résultats de la classification *classif82* en examinant le tableau IV.13.

Seuil Sim=0.82 : Le nombre de classes générées est 43

Classes	Nb Vsp/clas	Nb Elt/Clas	Nb Att/Clas	NbChem/Clas	ProfMy	SimMy	Ecart-Type
C1-82	177	2285	1565	1431	6,61	0,98	0,00
C2-82	32	388	290	293	5,44	0,86	0,01
C3-82	185	2230	1968	1763	5,41	0,86	0,01
C4-82	21	263	208	193	5,71	0,97	0,02
C5-82	15	190	154	130	5,47	0,95	0,01
C6-82	20	197	170	135	5,10	0,95	0,01
C7-82	12	147	134	147	5,33	0,94	0,01
C8-82	23	668	80	98	4,04	0,98	0,01
C9-82	85	828	686	607	5,00	0,98	0,00
C10-82	5	78	42	46	6,00	0,86	0,04
C11-82	40	473	467	364	4,48	0,95	0,03
C12-82	105	1938	541	583	4,95	0,98	0,02
C13-82	67	553	503	419	3,93	0,98	0,01
C14-82	13	197	122	121	5,38	0,96	0,02
C15-82	41	524	537	507	5,02	0,95	0,02
C16-82	56	563	681	395	4,16	0,97	0,02
C17-82	16	174	187	124	4,19	0,96	0,01
C18-82	17	283	162	170	6,24	0,89	0,01
C19-82	6	79	64	70	4,33	0,91	0,01
C20-82	33	496	314	327	6,58	0,95	0,02
C21-82	29	260	263	194	4,17	0,98	0,02
C22-82	26	234	244	174	4,08	0,98	0,01
C23-82	9	104	90	89	5,11	0,97	0,01
C24-82	18	232	193	186	5,56	0,95	0,01
C25-82	34	433	394	248	5,15	0,99	0,01
C26-82	30	291	248	216	5,00	0,98	0,02
C27-82	29	263	266	189	4,00	0,98	0,01
C28-82	7	70	63	63	4,00	0,98	0,01
C29-82	13	135	146	86	4,15	0,99	0,00
C30-82	22	226	248	156	4,18	0,98	0,01
C31-82	8	98	93	53	4,00	0,98	0,01
C32-82	30	300	231	225	5,17	0,97	0,01
C33-82	29	309	336	220	4,14	0,98	0,00
C34-82	42	445	583	305	4,10	0,98	0,00
C35-82	72	828	571	508	5,04	0,96	0,02
C36-82	145	1643	1696	1273	4,56	0,96	0,02
C37-82	10	128	104	124	5,40	0,91	0,01
C38-82	12	131	124	130	4,75	0,95	0,02
C39-82	44	917	340	242	4,02	0,98	0,01
C40-82	17	1401	769	991	6,18	0,98	0,01
C41-82	5	353	193	288	6,00	0,97	0,02
C42-82	2	134	70	115	6,00	0,99	0,01
C43-82	4	325	184	241	6,25	0,95	0,01
Totaux	1606	21814	16324	14239			

Tableau IV.13 - Résultats de la classification : *classif82*

Après l'examen des classes de chacune des classifications *classif80* et *classif82*, nous remarquons l'émergence d'une nouvelle classe $C_{10\text{-}82}$ qui regroupe 5 vues spécifiques. Nous avons ensuite exploré ces vues et nous avons constaté que : deux vues spécifiques sont rattachées à $C_{2\text{-}80}$, une vue spécifique est rattachée à $C_{3\text{-}80}$, une vue spécifique est rattachée à $C_{14\text{-}80}$, et une vue spécifique est rattachée à $C_{17\text{-}80}$.

Seuil de similarité à 83% : classif83

Les 1606 documents sont groupés en 43 classes.

La comparaison avec *classif82* a montré que les classes obtenues par *classif83* sont les mêmes que celles présentées par le tableau IV.13 (*classif82*). Après exploration des vues spécifiques rattachées à chacune des classes, nous avons remarqué que les résultats obtenus sont exactement les mêmes que ceux de *classif82*.

III.2.2. Bilan et synthèse

Le tableau IV.14 présente une synthèse des résultats des trois séries de tests : avec des seuils de similarité de 78%, 80% et 82%. Pour chaque série de tests, nous présentons : le

nombre de classes obtenues, la moyenne des vues spécifiques par classe, la moyenne des similarités moyennes intra-classe, la moyenne des écart-types intra-classe.

	classif78	***classif80***	***classif82***
Nombre de classes	40	42	43
Moyenne des vues spécifique par classe	40.15	38.23	37.34
Moyenne des similarités moyennes	0.95	0.96	0.96
Moyennes des écart-types	0.02	0.01	0.01

Tableau IV.14 - Synthèse des résultats de *classif78, classif80* et *classif82*

Nous remarquons que l'augmentation de la valeur du *Seuil de similarité* entraîne un surnombre de classes générées. En revanche, la diminution de cette valeur implique une réduction du nombre de classes. En conséquence le nombre de vues rattachées à chaque classe augmente, ce qui implique une hétérogénéité entre les individus d'une même classe (diminution de l'homogénéité intra-classe). Il faut donc trouver un compromis entre le nombre de classes générées et l'homogénéité intra-classe.

Les résultats de synthèse, des trois séries de tests, représentés par le tableau IV.14 montrent que les classifications *classif80* et *classif82* sont proches (les classifications *classif82* et *classif83* sont identiques). Nous constatons, au travers des expériences effectuées dans nos travaux antérieurs [Idarrou A. et al., 2012a] et [Idarrou A. et al., 2012b] et celles menées dans ces travaux, qu'un seuil de similarité de 80% donne des résultats satisfaisants.

Comme nous l'avons évoqué (Cf. chapitre III, section IV.2.3.1 page 123), les classes (vues génériques) peuvent subir des transformations au fur et à mesure de la classification et que cela peut engendrer le problème de rapprochement des classes. Au cours de ces séries de tests que nous avons menées, nous avons noté que de tel phénomène ne c'est pas produit. Cela est dû au fait que nous avons fixé a priori pour chaque test une *distance minimale* inter-classe (tableaux IV.11, IV.12 et IV.13).

Afin de s'assurer que chaque vue spécifique documentaire est rattachée à la bonne classe, nous avons proposé de recalculer les classes à la fin de chaque classification.

IV. Conclusion

Nous avons présenté dans ce chapitre le prototype *MMDOCREP*. Ce prototype nous a permis de valider les propositions de cette thèse de l'intégration d'un document à la visualisation des résultats d'une classification structurelle.

Plus précisément, le prototype *MMDocRep* nous a permis l'évaluation de notre approche de classification. Cette approche est basée sur une *mesure de similarité structurelle* que nous avons proposée. Une mesure basée sur l'isomorphisme de sous-graphes. Elle repose sur une fonction de pondération qui reflète les aspects hiérarchique et contextuel des composants des graphes. Elle est paramétrée par un seuil d*e similarité* permettant de définir a priori le degré de similarité entre le représentant de chaque classe générée et les individus de cette classe. Cela permet d'assurer une cohésion intra-classe.

L'augmentation, de la valeur du seuil de similarité entraine une forte homogénéité des classes. En contre partie cela permet la création d'un surnombre de classes. En revanche, la diminution de cette valeur permet de réduire le nombre de classes (tableau IV.14). Plus

précisément, lorsque la valeur du seuil de similarité diminue, le nombre de vues rattachées à chaque classe augmente, ce qui engendre une hétérogénéité entre les individus d'une même classe. Il faut donc trouver un compromis entre le nombre de classes générées et l'homogénéité intra-classe.

Les expériences menées (Cf. section III.1.2, page 149) ont montré l'intérêt du *sous-processus de filtrage* : réduire l'espace de comparaison des structures et par conséquent optimiser le temps de réponse de nos algorithmes. Avec un *seuil de filtrage* de 64%, nous avons obtenu les mêmes résultats qu'avec la classification manuelle (que nous avons considérée comme *classification de référence*) avec un gain de temps considérable.

La séparation des classes est l'un des critères pour qualifier un classifieur. La prise en compte de *la séparation des classes* permet de conserver le pouvoir discriminant de ces classes afin d'éviter leur rapprochement et leur chevauchement. Nous avons constaté au cours des expériences que nous avons menées (Cf. section III.2, page 151) que le problème de rapprochement des classes ne s'est pas posé.

Lorsque les classes sont suffisamment séparées, le problème d'appartenance d'une même vue spécifique à plusieurs classes ne sera pas envisagé. Un problème auquel nous n'avons pas été confrontés au cours des séries de tests que nous avons menées. Augmenter la distance inter-classe permet de *diminuer le bruit* et *augmenter la précision* de la classification.

V. Bibliographie

[Genane Y., 2004] Genane Youness « Contributions à une méthodologie de comparaison de partitions » Thèse de Doctorat de l'université de Paris 6 France, 2004.

[Idarrou A., 2010] Idarrou A., « Classification de documents multi-structurés : Comparaison des structures » Dans : *Colloque International Francophone sur l'Écrit et le Document (CIFED 2010), Sousse (Tunisie), 18/03/2010-20/03/2010*, Cépaduès, p. 501-506, 2010.

[Idarrou A. et al. 2010b] Ali Idarrou, Driss Mammass, Chantal Soulé-Dupuy, Nathalie Vallés-Parlangeau. "A generic Approach to the Classification of Multimedia Documents: a Structures Comparison". Dans : ICGST International Journal on Graphics, Vision and Image Processing, The International Congress for global Science and Technology, Vol. Volume 10 N. Issue VI, p. 13-18, december 2010.

[Idarrou A. et al., 2012a] Ali Idarrou, Chantal Soulé-Dupuy, Nathalie Vallés-Parlangeau. *Classification structurelle des documents multimédias basée sur l'appariement des graphes (regular paper).* Dans : *INFormatique des Organisations et Systemes d'Information et de Decision (INFORSID 2012), Montpellier (France), 29/05/2012-31/05/2012*, Association INFORSID, p. 539-554, 2012.

[Idarrou A. et al., 2012b] Ali Idarrou and Driss Mammass, "*Structural Clustering Multimedia Documents*: *An Approach based on Semantic Sub-graph Isomorphism*". *International Journal of Computer Applications* 51(1) : 14-21, August 2012. Published by Foundation of Computer Science, USA, Vol. 51 N. 1, August 2012. Accès : http://www.ijcaonline.org/archives/volume51/number1/8005-1343.

[Mbarki M., 2008] Mbarki Mohamed, « De la modélisation à l'exploitation des documents à structures multiples », Thèse de Doctorat de l'Université Paul Sabatier. - Toulouse, France, 2008.

[Yosr N. et al., 2009] Yosr Naïja, Sinaoui Kaouthar Blibech: "A novel measure for validating clustering results applied to road traffic". KDD Workshop on Knowledge Discovery from Sensor Data 2009: 105-113, (Paris, France, June 28 - 28, 2009).

Conclusion générale

I. Bilan et synthèse de nos propositions

Dans cette thèse, nous nous sommes intéressés à la classification structurelle des documents multimédias à structures multiples dans le cadre d'un entrepôt documentaire. Nous avons présenté une approche de *classification structurelle* basée sur un processus de comparaison de structures documentaires.

Dans la littérature, les approches qui ont abordé la classification se distinguent par le modèle utilisé pour représenter les objets et par la méthode utilisée pour classer ces objets.

Nous faisons partie de la catégorie des travaux qui ont abordé la classification *structurelle* (de structures de documents) *non-supervisée*, considérant que la structure est un facteur discriminant intéressant pour la classification documentaire. Nous avons utilisé les graphes (intégrant les structures et leurs liens) comme modèle universel et flexible pour représenter les documents multimédias à structures multiples toute en s'appuyant sur la théorie des graphes pour comparer ces structures.

Pour évaluer la proximité entre deux graphes, nous avons proposé une mesure de *similarité structurelle* basée sur la recherche d'isomorphisme de sous-graphes et reposant sur une fonction de pondération des graphes que nous avons introduit. Cette dernière permet d'exprimer des contraintes liées aux aspects hiérarchique et contextuel, des composantes (nœuds et arcs), dans la mesure où elle prend en considération la répartition de ces composantes : position d'un nœud dans un chemin mais aussi sa position par rapport aux nœuds frères. La similarité entre deux graphes est donc une agrégation entre ces contraintes.

La mesure proposée permet de montrer :

- qu'un graphe est inclus dans un autre,
- que deux graphes sont isomorphes (structurellement identique),
- que deux graphes sont disjoints ou non, en exprimant le degré de leur intersection.

Notre approche de classification structurelle *non-supervisée* est composée globalement de deux phases : (1) l'extraction de la vue spécifique d'un document à intégrer et (2) le rattachement de cette vue spécifique à la vue générique (représentant de la classe) la plus similaire. L'étape (2) passe par un processus de comparaison de la vue spécifique avec l'ensemble des vues génériques (les classes déjà générées) de l'entrepôt de documents. L'approche proposée est caractérisée par :

- le fait qu'un document n'est plus vu comme un ensemble de composants mais plutôt comme un ensemble organisé de granules reliés les uns aux autres de façon cohérente,
- la définition d'une fonction (mesure de ressemblance) permettant de réduire l'espace de comparaison des structures (Cf. chapitre III, section IV.2.2.2 page 121), en sélectionnant les vues génériques de l'entrepôt susceptibles d'être similaires à la vue spécifique du document à intégrer. Cela permet d'optimiser le temps de réponse de nos algorithmes de comparaison sans pour autant altérer la qualité de la classification,

- la transformation, en tenant compte du coût de cette transformation, des vues génériques afin d'augmenter leur représentativités. Cela permet d'enrichir les représentants des classes au fur et à mesure de la classification,
- le fait qu'elle repose sur une *mesure structurelle* qui reflète le sens et la structure des documents comparés, dans le sens où celle-ci repose sur une fonction de pondération qui traduit l'importance des composantes, d'un point de vu structurel, des graphes comparés,
- l'utilisation d'un *seuil de similarité* permettant de fixer le degré de similarité entre la classe générée et les individus de celle-ci. Cela permet d'assurer une cohésion intra-classe,
- la prise en compte de la *séparation des classes* qui permet d'assurer la stabilité des classes. Plus précisément, conserver une distance entre les classes est une solution qui permet d'éviter les problèmes de rapprochement et de chevauchement de ces classes.

Les expérimentations menées ont montré des résultats encourageants :

- dans la première expérience, nous avons obtenu les mêmes résultats qu'avec la classification manuelle,
- dans la deuxième expérience, les écart-types moyens sont faibles ce qui montre l'homogénéité des classes générées.

Nous pensons que la mesure de similarité entre graphes, que nous avons proposée, s'avère bien adaptée à la problématique de la classification structurelle des documents à structures multiples.

II. Perspectives

Nos algorithmes de classification semblent avoir des propriétés qui les rendent performant dans la mesure où ils reposent sur une mesure de *similarité structurelle* paramétrée par : (1) un *seuil de similarité* qui permet de définir le degré de similarité entre les individus d'une classe et le représentant de cette classe et (2) une distance de *inter-classe* permettant d'assurer la cohésion et la stabilité des classes générées. Toutefois, une étude plus fine doit être faite afin de déterminer la combinaison la mieux appropriée de ces deux paramètres.

Il serait intéressant de comparer nos résultats avec d'autres approches mais en l'absence d'une **base de référence de documents multimédias à structures multiples**, cette tâche reste non triviale. Il serait aussi utile de compléter les expérimentations sur des corpus de documents multi-structurés.

Nous aimerions :

- Poursuivre les expérimentations :
  - sur des corpus plus importants de documents multi-structures,
  - sur la qualité de la classification,
- Compléter la démarche de comparaison :
  - par une etude approfondie de la combinaison des seuils (filtrage, similarite)

  - en integrant des ressources terminologiques pour traiter les aspects semantiques. Cela peut être utile aussi bien en recherche d'information qu'en classification documentaire.

- Tester notre mesure sur d'autres problèmes et dans d'autres domaines dont les objets sont représentables en graphes :
  - en reconnaissance des formes, pour comparer les objets,
  - en chimie organique pour la classification des structures ou pour déterminer les sous structures qui composent une structure donnée,
  - en classification d'images,
  - en recherche d'information : en représentant les requêtes et les documents à l'aide des graphes. La mesure proposée permet ensuite d'évaluer l'inclusion ou l'intersection entre une requête et un document donnés.

Dans l'optique de l'application de nos algorithmes, il serait utile de travailler avec des chercheurs dans ces domaines afin d'élargir le champ d'application de notre approche.

Bibliographie générale

A

[Abascal R., 2003] Abascal R., Beigbeder M., Benel A., Calabretto S., Chabbat B., Champin P. A., Chatti, N., Jouve, D., Prie, Y., et Rumpler, B. (2003). « Modéliser la structuration multiple des documents » H2PTM, Hermès, Paris, France, 253-258.

[Abascal R. et al., 2004] Abascal R., Beigbeder M., Bénel A., Calabretto S., Chabbat B., Champin P. A., Chatti N., Jouve, D., Prié, Y., et Rumpler, B. (2004). « Documents à structures multiples ». *SETIT 2004*.

[Abascal R. et al., 2005] Abascal R., Rumpler B., Berisha-Bohé S. « Proposition d'une nouvelle structure de document pour améliorer la recherche d'information », *Proceedings of the CORIA'05*, ISBN: 2-9523810-0-3, IMAG, pp. 389-404, 2005.

[Abascal R. et al., 2007] Abascal-Mena R., B. Rumpler, « Accès au contenu des thèses numériques par leur structure sémantique » Document Numérique 10(2/200):9-35, Lavoisier - Hermes, ISBN 978-2-7462-2023, 2007.

[Afnor, 1987] AFNOR, Références bibliographiques : contenu, forme et structure. ISO 690 1987. Paris, 1987.

[Allen 1991] Allen J. F. (1991). "Time and time again: The many ways to represent time". International Journal of Intelligent Systems, 6(4).

[Aïtelhadj A. et al., 2009] Ali Aïtelhadj, « Classification de Structures Arborescentes : Cas de Documents XML », *CORIA, 6th French Information Retrieval Conference, Presqu'île* de *Giens*, France, May 5-7, 2009. Proceeding. LSIS-USTV, ISBN 2-9524747-1-0 page 301-317, 2009.

[Allen 1983] Allen J. F. (1983). "Maintaining knowledge about temporal intervals.‖ Communication ACM", 26(11), 837-843.

[Ambauen R. et al., 2003] Ambauen R., Fischer S. and BUNKE H. "Graph Edit Distance with Node Splitting and Merging, and Its Application to Diatom Identification". In *IAPR-TC15*, 2003.

[Anderberg M.R., 1973] Anderberg M.R. "Cluster Analysis for Applications", Academic Press,1973.

[Atteneave F., 1950] Atteneave F., "Dime,sons of Similarity", American Journal of Psychology, Vol. 65 pp 516-556, 1950.

B

[Bachimont B, 1999] Bachimont B., "Bibliothèques numériques audiovisuelles : des enjeux scientifiques et techniques". *Document Numérique, Numéro Spécial "Les Bibliothèques Numériques"*, 2(3-4):219-242, Janvier 1999.

[Bachimont B, 1998] Bachimont B. « Bibliothèques numériques audiovisuelles : Des enjeux scientifiques et techniques. » Revue Document numérique, 2(3), 219-242, 1998.

[Bachimont B, 2004] Bachimont B., et Crozat, S. « Instrumentation numérique des documents : pour une séparation fonds/forme. » Information-Interaction-Intelligence I3, 4(1), 95-104, 2004.

[BÄRECKE T., 2009] BÄRECKE Thomas « Isomorphisme inexact de graphes par optimisation évolutionnaire », Thèse de Doctorat de l'université de Paris VI, 2009.

[Bassok M. et al., 1997] Bassok, M. and Medin, D.L. "Birds of a feather flock together: Similarity judgments with semantically rich stimuli". In journal of Memory & Language, vol. 36, p. 311-336, 1997.

[Beigbeder M., 2004] Beigbeder M. (2004). « Les temps du document et la recherche d'information. » *Document numérique*, (2004/4), 55–64.

[Ben Aouicha M., 2009] Ben Aouicha Mohamed, « Une approche algébrique pour la recherche d'information structurée », Thèse de doctorat de l'Université de Paul Sabatier Toulouse 2009.

[Beney J., 2006] Beney Jean « *Classification supervisée de documents : théorie et pratique* », Hermes Science, février 2008, 184p.

[Berkhin P., 2002] Berkhin P. "*Survey of clustering data mining techniques"*, Accrue Software, 2002.

[Bisson G., 2000] Bisson G., « La similarité : une notion symbolique/ numérique. Apprentissage symbolique-numérique ». Eds Moulet, Brito, Cepadues Edition, 2000.

[Bres S., 1999] Bres S., Champin P.A., Heraud J.M., Herilier V., Jolion J.M., Loupias E., *TeleSUN A world wide multimedia TELEteaching System for UNiversities*, 1999.

[Bringay S., 2004] Bringay, S., Barry, C., et Charlet, J. (2004). « Les documents et les annotations du dossier patient hospitalier » *Revue I3 : Information-Interaction-Intelligence*, 4(1), 191-211.

[Bruno E. et al., 2007] Bruno E., Calabretto S., Murisasco E., « Documents textuels multistructurés : un état de l'art ». *Revue Information - Interaction - Intelligence (I3)* 7 (1) 2007.

[Bruno E. et Murisasco, 2006] Bruno, E., et Murisasco, E. (2006). "MSXD: A Model and a Schema for Concurrent Structures Defined over the Same Textual Data". *Database and Expert Systems Applications*, 172-181.

[Bruno E. et al., 2004] Bruno E., J. LeMaitre et E.Murisasco, « Temporalisation d'un document XML », In *Document Numérique*, volume 8(4), pages 125–141, 2004.

[Bryan M., 1998] Bryan, 1998 Bryan M. "Guidelines for using XML for Electronic Data Interchange", 1998 http://www.xmledi-group.org/xmledigroup/guide.htm.

[Bunke H. et Shearer k., 1998] Bunke, H. et Shearer K. "A graph distance metric based on the maximal common subgraph". Pattern Recogn. Letters 19 (3-4), 255–259, 1998.

[Bunke H., 1997] Bunke, H. (1997). "On a relation between graph edit distance and maximum common subgraph". Pattern Recogn. Letters 18 (9), 689–697.

[Bunke H., 1999] Bunke, H. 1999, "Error Correcting Graph Matching: On the Influence of the Underlying Cost Function". *IEEE Transactions on Pattern Analysis and Mach Intelligence*, 21(9): 917–922,1999.ISSN 0162-8828. doi:http: // dx.doi.org /10.1109 / 34.790431.

[Boeres M. et al., 2004] Boeres M., Ribeiro C. & Bloch I. (2004). "A randomized heuristic for scene recognition by graph matching". In *WEA 2004*, p. 100–113.

[Bouveyron C., 2006] Bouveyron C. « Modélisation et classification des données de grande dimension application à l'analyse d'mages » Thèse de Doctorat de Joseph Fourier, 2006.

[Bruno E. et Murisasco, 2006] Bruno, E., et Murisasco, E. (2006). MSXD: A Model and a Schema for Concurrent Structures Defined over the Same Textual Data. Database and Expert Systems Applications, 172-181.

[Bui Thi M-P., 2003] Minh Phung BUI THI, « La structuration sémantique des contenus des documents audiovisuels selon les points de vue de la production », Thèse de doctorat de l'Université de Paris VIII, 2003.

C

[Chabbat B., 1997] Chabbat B. *Modélisation Multiparadigme de textes réglementaires*. Thèse de doctorat, LISI. Lyon, décembre 1997, 392 p.

[Champclaux Y., 2009] Champclaux Y., « Un modèle de recherche d'information basé sur les graphes et les similarités structurelles pour l'amélioration du processus de recherche d'information », Thèse de Doctorat de l'Université de Toulouse, 2009.

[Champin P-A., Solnon C., 2003] Champin P-A., Solnon C., "Measuring the similarity of labeled graphs". Dans 5th Int. Conf. On Case-Based Reasoning (ICCBR 2003), Kevin D. Ashley and Derek G. Bridge ed. Trondheim (NO). pp. 80-95. LNAI 2689. Springer Berlin. 2003.

[Caro S., 2003] CARO Stéphane, Manuscrit auteur, publié dans « Techniques de l'ingénieur Documents numériques Gestion de contenu », 2003.

[Chatti N. et al., 2007] Chatti, N., Calabretto, S., Pinon, J. M., et Kaouk, S. « MultiX: an XML-based formalism to encode multi-structured documents », *Proceedings of Extreme Markup Languages 2007*, 2007.

[Chatti N. et al., 2006] Chatti N., Kaouk S., Calabretto S., and Jean-Marie Pinon. "MultiX : an XML-based formalism to encode multi-structured documents". In Proceedings of Extreme Markup Languages'2006, Montréal (Canada), August 2006.

[Chaudiron et al., 2000] S. Chaudiron, F. Role, M. Ihadjadene, 2000. *CodeX : un système pour la définition de vues multiples guidée par les usages*, CIDE 2000, Lyon, FR, 2000, pp. 71-81.

[Celeux G. et al., 1989] Celeux, G., E. Diday, G. Govaert, Y. Lechevallier, et H. Ralambondrainy (1989). « *Classification Automatique des Données, Environnement statistique et informatique* ». Dunod informatique, Bordas, Paris.

[Conte D. et al., 2004] Conte D., Foggia P., Sansone C. et Vento M., "Thirty Years of Graph Matching in Pattern Recognition". Int. Journal of Pattern Recognition and Artificial Intelligence, 18(3):265–298, 2004.

[Cordella L.P. et al., 2001] Cordella, L.P., Foggia, P., Sansone, C. et Vento, M. 2001. "An Improved Algorithm for Matching Large Graphs", *Proc. 3rd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, pp. 149 -159.

[Costa G. et al., 2004] Costa, G., G. Manco, R. Ortale, et A. Tagarelli. "A Tree-Based Approach to Clustering XML Documents by Structure". In *PKDD*, pp. 137–148, 2004.

[Chrisment C. et al., 2002] Chrisment C., F. Sedes. *Media annotations: toward a unified representation.* Chapitre du livre Multimedia mining, octobre 2002. Kluwer Academic Publisher.

D

[Dalamagas T. et al., 2004] Dalamagas T., Cheng T., Winkel K-J. and Sellis T., « Clustering XML Documents Using Structural Summaries », In *EDBT Workshops* p 547-556, 2004.

[Dalamagas T. et al., 2006] Dalamagas T., Cheng T., Winkel K-J, Sellis T.K., "A methodology for clustering XML documents by structure". Information Systems 31(3): 187-228 (2006).

[Djemal K. 2010] Djemal Karim, « De la modélisation à l'exploitation des documents à structures multiples », Thèse de Doctorat de l'Université de Paul Sabatier. - Toulouse, France, 2010.

[Davies D.L., 2000] Davies D.L., Bouldin D.W.(2000): "A cluster separation measure". IEEE Trans. Pattern Anal. Machine Intell, 1(4), 224-22.

[Del Razo L.F. et al., 2006] Del Razo Lopez F., Laurent A., Poncelet P., Teisseire M., « Recherche de sous-structures fréquentes pour l'intégration de schémas XML », In Conférence Extraction et Gestion des Connaissances (EGC 2006), Lille, Janvier 2006, volume II, p 487-498.

[Demirci M. F. et al., 2006] Demirci M. F., A. Shokoufandeh, Y. Keselman, L. Bretzner, et S. Dickinson (2006). "Object recognition as many-to-many feature matching", International Journal of Computer Vision 69(2), 203–222.

[Denoyer L., 2004] Denoyer L., « Apprentissage et Inférence statistique dans les bases de documents structurés: Application aux corpus de documents textuels », Thèse de Doctorat Paris 6, 2004.

[Doucet A. et al., 2002] Doucet A. et Ahonen-Myka H., "Naïve Clustering of a large XML Document Collection", In INEX Workshop, pp 81-87, 2002.

[Durusau P. et al., 2002] Durusau, P., et O'Donnell, M. B. (2002). "Concurrent markup for XML documents". *Proc. XML Europe*.

[Dunn J., 1974] Dunn J. (1974): "Well Separated clusters and optimal fuzzy partitions ", Journal of Cybernetics, 4, 95-104.

E,F

[El moubarki L., 2009] El moubarki L. « Décomposition et évaluation des mesures de stabilité d'un partitionnement » Thèse de Doctorat de l'université de Paris-Dauphine, 2009.

[Elghazel H., 2007] Elghazel Haytham, « Classification et Prévision des Données Hétérogènes : Application aux Trajectoires et Séjours Hospitaliers », Thèse de Doctorat de l'Université Claude Bernard Lyon 1 France, 2007.

[Egyed Z.E., 2003] Egyed Zsigmond E., « *Gestion des connaissances dans une base de documents multimédias* », Thèse de doctorat en informatique, INSA de Lyon, octobre 2003.

[Ercegovac Z., 1999] Ercegovac Z., "Introduction to the Special Topic Issue, Integrating Multiple Overlapping Metadata Standards.". dans Journal of the American Society for Information Science, Vol. 50, n°. 13, p. 1165-1170, novembre 1999.

[El moubarki L., 2009] El moubarki L. « Décomposition et évaluation des mesures de stabilité d'un partitionnement » Thèse de Doctorat de l'université de Paris-Dauphine, 2009.

[Fourel, F., 1998] Fourel, F. « Modélisation, indexation et recherche de documents structurés » Thèse de doctorat, Université Joseph Fourier, Grenoble, 1998.

[Francesca F.D., et al., 2003] Francesca F. D., Gordano G., Ortale R., Tagarelli A "Distance-based Clustering of XML Documents", *In Proceedings of the First International Workshop on Mining Graphs,Trees and Sequences,* 2003, pp 75-78.

G,H

[Garey M., 1979] Garey M. et Johnson D., "Computers and Intractability : A Guide to the Theory of NP-Completeness", W. H. Freeman, New-York, 1979.

[Gardenfors P., 2000] Gardenfors P. Conceptual spaces : "the geometry of though", Cambridge, MIT Press.

[Genane Y., 2004] Genane Youness « Contributions à une méthodologie de comparaison de partitions » Thèse de Doctorat de l'université de Paris 6 France, 2004.

[Gentner D., 1983] Gentner D., "Structure-mapping: A theoretical framework for analogy", *Cognitive Science*, 7, 155-170. (Reprinted in A. Collins & E. E. Smith (Eds.), *Readings in cognitive science: A perspective from psychology and artificial intelligence*. Palo Alto, CA: Kaufmann), 1983.

[Gentner D. et al. ,1989] Gentner D. "*The mechanisms of analogical learning"*. *In* S. Vosniadou et A. Ortony (dir.), *Similarity and analogical reasoning,* p. 199-241. Cambridge: Cambridge University Press. 1989.

[Goldfarb C.F. et al., 1990] Goldfarb C.F., Rubinsky Y., "*The SGML handbook"*. Clarendon Press, Oxford, 1990.

[Harchaoui Z. et Bach, 2007] Harchaoui Z. et Bach F., "Image Classification with Segmentation Graph Kernels", Dans CVPR. IEEE, 2007.

[Holyoak K.J. et al., 1989] Holyoak, K. J. et Thagard, P. "Analogical mapping by constraint satisfaction", Cognitive Science, 13, p. 295-355, 1989.

[Hidovi D. et al., 2004] Hidovi D., Pelillo M. (2004), "*Metrics for Attributed Graphs Based on the Maximal SimilarityCommon Subgraph*". International Journal of Pattern Recognition and Artificial, 7/2004.

[Hopcroft J.E. et al., 1974] Hopcroft J. E. et Wong, J. K, "Linear Time Algorithm for Isomorphism of Planar Graphs", *In Proceedings 6th ACM Symposium on Theory of Computing*, pp. 172 -184, 1974.

[Hu H. et al., 2005] Hu H., Hang Y., Han J., et X. Zhou (2005). "Mining coherent dense subgraphs across massive biological network for fonctional discovery", Bioinformatics 1(1), 1–9.

[Hummel J.E., 2000] Hummel, J.E. "*Where view-based theories break down: The role of structure in shape perception and object recognition"*, In E. Dietrich and A. Markman (Eds.). Cognitive Dynamics: Conceptual Change in Humans and Machines. Hillsdale, NJ: Erlbaum, 2000.

[Hummel J.E., 2001] Hummel, J.E. "*Complementary solutions to the binding problem in vision: Implications for shape perception and object recognition" Visual Cognition*, 8, p. 489-517, 2001.

I,J,K

[Idarrou A., 2010] Idarrou A., « Classification de documents multi-structurés : Comparaison des structures » Dans : *Colloque International Francophone sur l'Écrit et le Document (CIFED 2010), Sousse (Tunisie), 18/03/2010-20/03/2010*, Cépaduès, p. 501-506, 2010.

[Idarrou A. et al., 2010a] Ali Idarrou, Driss Mammass, Chantal Soulé-Dupuy, Nathalie Vallés-Parlangeau. « Classification of multi-structured documents: A comparison based on media image » *Lecture Notes in Computer Science* **,** Vol. LNCS 6134, Springer, p. 428-438, 2010.

[Idarrou A. et al. 2010b] Ali Idarrou, Driss Mammass, Chantal Soulé-Dupuy, Nathalie Vallés-Parlangeau. "A generic Approach to the Classification of Multimedia Documents: a Structures Comparison". Dans : ICGST International Journal on Graphics, Vision and Image Processing, The International Congress for global Science and Technology, Vol. Volume 10 N. Issue VI, p. 13-18, december 2010.

[Idarrou A. et al., 2012a] Ali Idarrou, Chantal Soulé-Dupuy, Nathalie Vallés-Parlangeau. *Classification structurelle des documents multimédias basée sur l'appariement des graphes (regular paper).* Dans : *INFormatique des Organisations et Systemes d'Information et de Decision (INFORSID 2012), Montpellier (France), 29/05/2012-31/05/2012*, Association INFORSID, p. 539-554, 2012.

[Idarrou A. et al., 2012b] Ali Idarrou and Driss Mammass, "*Structural Clustering Multimedia Documents*: *An Approach based on Semantic Sub-graph Isomorphism*". *International Journal of Computer Applications* 51(1):14-21, August 2012. Published by Foundation of Computer Science, USA, Vol. 51 N. 1, August 2012. Accès : http://www.ijcaonline.org/archives/volume51/number1/8005-1343

[Jain A.k., 1999] Jain A. k., Murty M. N. and Flynn: P. J., "*Data Clustering: A Review"*, ACM Computing Surveys,31, pp. 264-323, 1999.

[Jedidi A., 2005] Jedidi A., « Modélisation générique de documents multimédia par métadonnées : mécanismes d'annotation et d'interrogation », Thèse de Doctorat de L'université de Paul sabatier Toulouse, France. 2005.

[Kalakech M., 2011] Kalakech Mariam, « Sélection semi-supervisée d'attributs : Application à la classification de textures couleur », Thèse de Doctorat, Université de Lille 1, 2011.

[Kanal L., 1993] Kanal L. N., "On pattern, categories, and alternate realities", *Pattern Recognition Letters"*, Vol. 14, pp. 241-255, 1993.

[Klinger S. et Austin J., 2005] Klinger Stefan, Austin Jim : "Chemical similarity searching using a neural graph matcher", ESANN 2005: 479-484.

[Kriegel H.P. et al., 2003] KRIEGEL H.P. ET SCHÖNAUER S., "Similarity Search in Structured Data, Lecture Notes in Computer Science", N° 2737, 2003, pp. 309-319.

[Kutty S., 2008] Kutty S., Tran, T., Nayak, R., et Li, Y. (2008). "Clustering XML Documents Using Closed Frequent Subtrees" : A Structural Similarity Approach‖. Lecture Notes In Computer Science, 183-194.

L

[Laborie S., 2008] Laborie S., adaptation sémantique de documents multimédia, Doctorat de l'Université de Joseph Fourier-Grenoble 1, 2008.

[Laufer R. et Scavetta D., 1992] Laufer R. et Scavetta D., « Texte, hypertexte, hypermédia, Paris, PUF, 1992, p.3

[Landow G. P., 1992] Landow G. P. (1992): Hypertext: the convergence of contemporary critical theory and technology, *John Hopkins University Press*.

[Laroum S. et al., 2009] Sami Laroum, Nicolas Béchet, Hatem Hamza et Mathieu Roche, "Classification automatique de documents bruités à faible contenu textuel", Manuscrit auteur, publié dans RNTI : Revue des Nouvelles Technologies de l'Information 1, 2009.

[Lementini E.C. et al., 1993] Lementini E.C., Felice P., Oosterom D. et Van P., "A Small Set of Formal Topological Relationships Suitable for End-User Interaction", dans le Proc. of 3rd International Symposium on Advances in Spatial Databases, Berlin Heidelberg New York, juin 1993, Springer Verlag, LNCS n° 692, p. 277-295.

[Luc C., 2001] LUC Christophe, « Une typologie des énumérations basée sur les structures rhétoriques et architecturales du texte », *Actes de TALN 2001, Université de Tours*, juillet 2001, 263-272.

[Lavoisier, 1789] Lavoisier, dans son ouvrage : "Traité élémentaire de chimie présenté dans un ordre nouveau et d'après les découvertes modernes" ,1789.

[Lebart L. et al., 1982] Lebart L., Maurineau A., Piron M. (1982): « Traitement des données statistiques », Dunod, Paris.

[Levenshtein V., 1966] Levenshtein V., "Binary Codes Capable of Correcting Deletions, Insertions and Reversals", Soviet Physics Doklady, vol. 10, p. 707, 1966.

[Lin D. 1998] Lin D., "An information-theoretic definition of similarity", In Proc. 15th International Conf. on Machine Learning, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.

[Lord P et al., 2003] Lord P, Stevens R, Brass A, Goble C. "Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation". Bioinformatics;19:1275–1283, 2003.

M,N

[Mann W.C. et al., 1998] Mann, W.C., Thompson, S.A. 1988. Rhetorical Structure Theory : Toward a functional theory of text organization. Text, 8(3). 243-281.

[Manning C.D., 2008] Manning C.D., Raghavan P., Schütze H., "Introduction to Information Retrieval", Cambridge University Press, 2008.

[Marcoux Y., 1994] Marcoux, Y. (1994). « Les formats normalisés de documents électroniques » ICO. Intelligence artificielle et sciences cognitives au Québec, 6(1-2), 56-65.

[Mbarki M. 2008] Mbarki M., Gestion de l'hétérogénéité documentaire : le cas d'un entrepôt de documents multimédias., Thèse de Doctorat de l'Université de Paul Sabatier, Toulouse 3 France, 2008.

[Medin D. et al., 1993] Medin, D., Goldstone, R. L. et Gentner, D., "Respect for similarity. *Psychological Review", vol. 2,* p. 254-278, 1993.

[Metzger J. P. et al., 2004] Metzger J. P., et Lallich-Boidin, G. (2004). « Temps et documents numériques ». *Document numérique,* (2004/4), 11–21.

[Messmer B.T. et al., 1999] Messmer, B.T., et Bunke, H. 1999. "A Decision Tree Approach to Graph and Subgraph Isomorphism Detection", *Pattern Recognition*, 32, pp. 1979 – 1998.

[Minsky M., 1991] Minsky M., "Logical versus analogical, or symbolic versus connectionist, or neat versus scruffy", *Artificial Intelligence Magazine*, Vol. 12 (2), pp. 34-51, 1991.

[Nanard M., 1996] Nanard M., and all. La métaphore du généraliste : acquisition et utilization de la connaissance macroscopique sur une base de documents techniques. In Acquisition et Ingénierie des Connaissances - Tendances actuelles. N.Aussenac-Gilles, P. Laublet, C. Reynaud. Toulouse : CEPADUES, pages 285–304, 1996.

[Nassr N., 1999] N. Nassr, "*Oragnisation et Indexation automatique de documents multilingue"*. Rapport de DEA 2IL de l'université Paul Sabatier. 1999.

[Nierman A. et Jagadish, 2002] Nierman A., Jagadish H. V., "Evaluating Structural Similarity in XML Documents" , *In Proceedings of the Fifth International Workshop on the Web and Databases*, WebDB, 2002, Madison,Wisconsin, USA.

O,P

[Oh,Il-Seok et al., 1999] Oh,Il-Seok, Lee J., Suen C., Analysis of Class Separation and Combination of Class Dependent Features for Handwriting Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 10, October 1999, 1089-1094.

[Papadias D. et al., 1997] Papadias, D., et Theodoridis , Y. (1997). "Spatial relations, minimum bounding rectangles, and spatial data structures". *International Journal of Geographical Information Science,* 11(2), 111–138.

[Poullet L., 1997] L. Poullet. « Formaliser *la sémantique des documents – Un modèle unificateur »,* Actes de la Xème Conférence INFORSID'1997, Toulouse, juillet 1997. pp. 339-352.

[Portier P.E., 2010] Pierre-Edouard PORTIER « Construction des Documents Multistructurés dans le Contexte des Humanités Numériques », Thèse de Doctorat de l'INSA De Lyon France, 2010.

[Poullet L. et al., 1997] Poullet L., Pinon J.M., Calabretto S.. Semantic Structuring of Documents. Proceedings of the Third Basque International Workshop on Information Technology, BIWIT'97, Biarritz, July 1997, pp. 118–124.

R

[Ramel J-Y., 2006] Jean-Yves Ramed **:** Habilitation à diriger les recherches UNIVERSITE FRANCOIS RABELAIS DE TOURS, 2006.

[Razo F. D. et al., 2005] Razo F. D., A. Laurent, et Teisseire M. « Représentation efficace des arborescences pour la recherche des sous-structures fréquentes », In *Actes de l'atelier Fouille de données complexes, Conférence Extraction et Gestion des Connaissances (EGC 2005)*, pp. 113.120.

[Roux M., 1986] Roux M. « Algorithmes de Classification, Masson »,1986.

[Roxin I. et Mercier D., 2004**]** Roxin Ioan, Mercier D., "Multimédia, les fondamentaux : Introduction à la représentation numérique", Vuibert, 2004.

[Roger T. et al., 2003] Roger T. « Pédauque. Document : forme, signe et médium, les reformulations du numérique ». Working paper. Version 3 du 08 juillet 2003.

[Roisin C., 1999] Roisin Cécile. *Document structurés multimédias*. Habilitation à diriger les recherches. Institut National Polytechnique de Grenoble, septembre 1999.

[Roisin C., 1998] Roisin Cécile : "Authoring structured multimedia documents". In Proceedings of the Conference on Current Trends in Theory and Practice of Informatics, pages 222-239, 1998.

[Ros J. et al, 2005] Julien Ros, Christophe Laurent, Jean-Michel Jolion and Isabelle Simand, "Comparing string representation and distances in a natural images classification task. In Graphbased representations" in Pattern Recognition, pages 72-81, 2005.

S

[Saleem K., 2008] Saleem Khalid. (2008). "schema matching and integration in large scale snario". Thèse de Doctorat de L'Université Montpellier II, France, 2008.

[Schlieder T. et al., 2002] Schlieder T., Meuss M., « Querying and Ranking XML Documents », *Special Topic Issue of the Journal of the American Society of Information Science on XML and Information Retrieval*, 2002.

[Shang H., 2010] Shang H., K. Zhu, X. Lin, Y. Zhang, et R. Ichise (2010). "Similarity search on supergraph containment". In Proc. of ICDE, pp. 637–648.

[Salton G., 1971] Salton G. (1971). The SMART Retrieval System – experiments. *in automatic document processing. U Perntice-Hall, Inc., Englewood Cliffs, NJ.*

[Scuturici M., 2002] Scuturici M., *Contrinbution aux techniques orientée objet de gestion des séquences vidéo pour les serveurs Web*, PhD, INSA Lyon, 2002, 118 p.

[Sanz I. et al., 2005] Sanz I., Mesiti M., Guerrini G., and Berlanga Llavori R. (2005). "Approximate Subtree Identification in Heterogeneous XML Documents Collections". Lecture notes in computer science, 2005.

[Sigogne A., 2008] Sigogne Anthony, « Classification incrémentale et regroupement de documents Web », Stage de Master de recherche, Université Paris-Est Marne-la-Vallée, 2008.

[Smeulders A.W. et al., 2000] Smeulders, A. W., Worring, M., Santini, S., Gupta, A., and Jain, R. 2000. "Content-based image retrieval at the end of the early years". IEEE Trans. Pattern Analysis and Machine Intelligence 22, 12, 1349{1380.

[Sokal R.R. et Sneath, 1963] Sokal R.R. and Sneath P.H., "Principles of Numerical Taxonomy". W. H. Fressman and Compagny, 1963.

[Sorlin S., 2006] Sorlin S. « Mesurer la similarité de graphes », Thèse de Doctortat de l'Université de Claude Bernard Lyon I France , 2006.

[Sorlin S. et al., 2006] Sorlin S., Sammoud O., Solnon C., Jolion JM., « Mesurer la similarité de graphes », Dans Extraction de COnnaissance à partir d'Images (ECOI 2006), Atelier de Extraction et Gestion de Connaissances (EGC 2006), Nicole VINCENT et Nicolas LOMENIE ed. ed. Lille. pp. 21-30, 2006.

[Sorlin S. et Solnon C., 2005] Sorlin S., C. Solnon. "Reactive tabu search for measuring graph similarity". In 5th IAPR-TC-15 workshop on Graph-based Representations in Pattern Recognition, Luc Brun, Mario Vento ed. Poitiers. pp. 172-182. Springer-Verlag. 2005.

[Soulé-Dupuy C., 2001]Soulé-Dupuy, Chantal. « Bases d'informations textuelles : des modèles aux applications », Habilitation à diriger des recherches, Université Paul Sabatier, France 2001.

[Sylvain L., 2009] Sylvain Lamprier, Tassadit Amghar, Bernard Levrat, and Frédéric Saubion. « Clustering en recherche d'information : Concentration vs. Distribution de l'information pertinente ». In CORIA'09 : 6ième Conférence en Recherche d'Information et Applications, pages 115-130, 2009.

[STEICHEN O. et al., 2006] STEICHEN O., DANIEL-LE BOZEC C., THIEU M., ZAPLETAL E. et JAULENT M.-C. (2006). « Computation of semantic similarity within an ontology of breast pathology to assist inter-observer consensus», Comput Biol Med, 36(7-8), 768–788.

[Suard F. et al., 2007] Suard F. A. Rakotomamonjy, **«** Mesure de similarité de graphes par noyau de sacs de chemins », 21ième colloque GRETSI sur le traitement du signal et des images, Troyes, septembre 2007.

T

[Tagarelli A., 2010] Andrea Tagarelli, Sergio Greco: "Semantic clustering of XML documents". ACM Trans. Inf. Syst. 28(1), 2010.

[Tannier X., 2006] Tannier X., « Traitement automatique du langage naturel pour l'extraction et la recherche d'information ». Technical report, Ecole Nationale Sup_erieure des Mines de Saint-Etienne, July 2006. 16, 17.

[Thuong T.T., 2003] Tien TRAN THUONG, « *Modélisation et traitement du contenu des médias pour l'édition et la présentation de documents multimédias* ».Thèse de Doctorat de l'Institut National Polytechnique de Grenoble, 2003.

[Thibaut J.P, 1997] Thibaut, J.-P. « Similarité et catégorisation ». L'année psychologique, 97, p. 701-736, 1997.

[Tennison J. et al., 2002] Tennison, J., et Piez, W. (2002). « The Layered Markup and Annotation Language » (LMNL). Extreme Markup, Montreal.

[Termier A. et al., 2002] Termier A., Rousset, M. C., et Sebag, M. (2002). "TreeFinder: a First Step towards XML Data Mining", Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02), IEEE Computer Society Washington, DC, USA, 450.

[Tversky A., 1977] Tversky, A. (1977), "Features of Similarity". In *Psychological Review*, Volume 84, pp. 327–352.American Psychological Association Inc.

V,Y

[Van C., 1994] Van Cutsem, "Classification and Dissimilarity Analysis", Springer Verlag, 1994.

[Vazirgiannis M. et al., 1998] Vazirgiannis M., Theodoridis, Y., and Sellis, T. K., "Spatio-Temporal Composition and Indexing for Large Multimedia Applications Multimedia Systems", 6(4), pp. 284-298, 1998.

[Vilain M. et al., 1986] Vilain M., Kautz H. A. (1986). Constraint propagation algorithms for temporal reasoning. In AAAI-86. p. 132-144.

[Vellucci L., 1998] Vellucci L, "Metadata", Annual Review of Information Science and Technology, Vol. 33, 1998, p 187-222.

[Vercoustre A.M. et al., 2006] Vercoustre, A. M., Fegas, M., Lechevallier, Y., Despeyroux, T., et Rocquencourt, I. (2006). « Classification de documents XML à partir d'une représentation linéaire des arbres de ces documents ». Paris, France, 433–444.

[Yi J. et Sundaresan N., 2002] Yi, J., et Sundaresan, N. (2000). "A classifier for semi-structured documents". *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM New York, NY, USA, 340-344.

[Yosr N. et al., 2009] Yosr Naïja, Sinaoui Kaouthar Blibech: "A novel measure for validating clustering results applied to road traffic". KDD Workshop on Knowledge Discovery from Sensor Data 2009: 105-113, (Paris, France, June 28 - 28, 2009).

W,Z

[Wallis W.D. et al., 2001] Wallis W.D., Shoubridge P., Kraetz M., Ray D., "Graph distances using graph union". Pattern Recognition Letters 22 (2001).

[Weiss M.A., 1998] Weiss M. A, "*Data Structures And Algorithm Analysis*", *In C*, 1998.

[Wieczorek S., 2009], Wieczorek S., Gilles Bisson, Sylvaine Rou « Atelier AGS : Apprentissage et Graphes pour les Systèmes complexes », Hammamet, 25 Mai 2009 Dans le cadre de la plate-forme AFIA,CAp'09.

[Wisniewski G., 2005] Wisniewski G., Denoyer L., Gallinari P, « Classification automatique de documents structurés. Application au corpus d'arbres étiquetés de type XML » CORIA 2005: 167-184.

[Witten I.H., 1999] WITTEN I. H., MOFFAT A. et BELL T. C. (1999). "Managing gigabytes: compressing and indexing documents and images". Morgan Kaufmann.

[Woodhead N., 1991] Woodhead N., "Hypertext and hypermedia: theory and applications", London: Sigma Press and Addison-Wesley Co, 231 p, 1991.

[Zhang N. et al., 2006] Zhang, N., T. Özsu, I. Ilyas, et A. Aboulnaga (2006). "Fix: Feature-based indexing technique for xml documents". In Proc. of VLDB, pp. 259–270.

Annexe

Sommaire de l'annexe

Dans la suite nous considérons *:*
Seuil_Filtrage (le seuil de *filtrage*), *Seuil_Sim* (le seuil de *similarité*) *et* d_{min} (distance minimale inter-*classe*) : trois paramètres fixés a priori (réels),
DWg : l'ensemble des vues génériques,
Vgcand : l'ensemble des vues génériques candidates à la comparaison,
cout_T : coût de transformation d'un graphe en un autre (nombre réel).

I. Algorithme de filtrage : *FiltrerVueGenCand()*

// FilterVueGenCand : algorithme de présélection des vues génériques candidates à la comparaison.*

Soit Rep_d=(V,E) ;
entier NbNod_elt_Rep_d ; // nombre de nœuds de Rep_d de type élément*
entier NbNod_elt_Vg ; // nombre de nœuds de Vg de type élément*
*entier NbNod_att_Rep_d ; //*nombre de nœuds de Rep_d de type attribut (ou métadonnée)*
*entier NbNod_att_Vg ; //*nombre de nœuds de Vg de type attribut (ou métadonnée)*
*entier NbNod_elt_Com ; //*nombre de nœuds communs à Rep_d et Vg de type élément*
entier NbNod_att_Com;// nombre de nœuds communs à Rep_d et Vg de type attribut (ou métadonnée)*
réel Coeff_f ; // Coeff_f le coefficient de filtrage prédefini a priori*

Début
Vgcand= ∅ ; // ensemble des vues candidates à la comparaison
NbNod_elt_Rep_d =0 ; NbNod_att_Rep_d = 0;
pour chaque u de V
si type(u)= "Elt" alors NbNod_elt_Rep_d = NbNod_elt_Rep_d + 1
sinon NbNod_att_Rep_d = NbNod_att_Rep_d + 1
finpour // calcul du nombre de nœuds de V de type éléments et de type attributs*
pour chaque Vg de Dw_g *//* pour chaque vue générique de l'entrepôt*
// si Vg vérifie l'ordre des nœuds frères*
entier NbNod_elt_Com=0, NbNod_att_Com=0; NbNod_elt_Vg=0;
Vg=(V',E') ; NbNod_att_Vg =0;
pour chaque u' de V'
si type(u')= "Elt" alors NbNod_elt_Vg= NbNod_elt_Vg + 1
sinon NbNod_att_Vg = NbNod_att_Vg + 1
finpour // calcul du nombre de nœuds de V' de type éléments et de type attributs*
pour chaque u de V // calcul du nombre de nœuds communs (éléments et attributs)*
pour chaque u' de V'
si (type(u)= type(u') et u similaire à u') alors
si type(u)= "Elt" alors NbNod_elt_Com = NbNod_elt_Com + 1
Sinon NbNod_att_Com = NbNod_att_Com + 1
finsi
finsi
finpour
finpour

si (*NbNod_att_Rep_d* ≠ *0 et NbNod_att_Vg* ≠ *0*) *alors*

$$Coeff\_f = \left[\frac{NbNod\_elt\_Com}{NbNod\_elt\_Rep\_d} + \frac{NbNod\_elt\_Com}{NbNod\_elt\_Vg} + \frac{NbNod\_att\_Com}{NbNod\_att\_Rep\_d} + \frac{NbNod\_att\_Com}{NbNod\_att\_Vg} \right] / 4$$

sinon
si (*NbNod_att_Rep_d = 0 et NbNod-att_Vg* =0) *alors*

$$Coeff\_f = \left[\frac{NbNod\_elt\_Com}{NbNod\_elt\_Rep\_d} + \frac{NbNod\_elt\_Com}{NbNod\_elt\_Vg} \right] / 2$$

sinon

$$Coeff\_f = \left[\frac{NbNod\_elt\_Com}{NbNod\_elt\_Rep\_d} + \frac{NbNod\_elt\_Com}{NbNod\_elt\_Vg} \right] / 3$$

finsi
finsi
si (Coeff_f >= Seuil_Filtage) alors Vgcand= Vgcand $\cup$ *{Vg}*
finpour
Fin

II. Algorithme de pondération d'un graphe

Ponderation*(Graphe G)*
Soit k le nombre maximum de nœuds fils (des nœuds de G)

Début
entier i, N2prof, k=10; réel*, α, pd, pdRp*;
pour chaque relation *Ri* de *G*
Ri = (*N1*,*N2*) //**N1*, et *N2* sont respectivemt l'origine et l'extrémité de la relation *Ri*
si (type(N2)= "Elt" alors α = ordre(*N2*) *sinon α = 1*
si (*prof*(*N2*) = 1) *alors*
pd =1 – *α / k*
sinon
Np = père(*N1*) //* *Np : nœud père de N1*
Rp = (*Np*,*N1*) //* *la relation père de Ri*
pdRp = P_e(*Rp*) //* *le poids de la relation Rp avec* P_e *(formule [25])*
N2prof = prof(*N2*) // **profondeur de N2*
pd = pdRp – *α* / k^{N2prof}
finsi
//* *attribuer le poids pd à la relation (l'arc) Ri*
finpour
Fin

III. Algorithme de transforamtion d'un graphe en un autre

TransformationGraph*(graph G, graph G')*

Soit CH_G*={*chm_1*,* chm_2*, ……,*chm_n*} l'ensemble de chemins de G,*
Soit $CH_{G'}$*={*chm'_1*,* chm'_2*, ……,*$chm'_{n'}$*} l'ensemble de chemins de G',*
réel d ; entier c,k ; boolean e_existe;
Début
d = 1 ; cout_T = 0 ; *k* = 10 // **10 nœuds fils max par nœud*
pour chaque chm_i *de* CH_G
pour chaque chm'_j *de* $CH_{G'}$ //*recherche du chemin de $CH_{G'}$ le plus similaire*
si ((*Dist_Chm*(chm_i, chm'_j) < *d*) *alors* //* *à* chm_i
d = Dist_Chm(chm_i, chm'_j)
c = j
finsi
finpour //* *d distance minimale*

```
si 0 < d < 1 alors   //* d<1 donc ∃ c tel que chm'c (chm'c ∈ CHG') est partielement
                                  //*similaire à chmi
 pour chaque e de chmi  //*Ajout nœuds et arcs de chmi qui n'existent pas dans chmc
      e = (u,v)
     e_existe = faux
    pour chaque e' de chm'c
      if (e similaire à e') alors e_existe=vrai   //* quitter
    finpour
   si (e_existe = faux) alors
      //*Ajout de l'arc e dans chm'c
     si (Type(v)= "Elt" alors α =ord(v) sinon α = 1
     cout_T = cout_T + α / k^prof(v)
   finsi
  finpour
 sinon si d=1 alors      //* Ajout de tout le chemin
  pour chaque e de chmi
      e = (u,v)
      //*Ajout de l'arc e dans le grpahe G'
    si (Type(v)= "Elt" alors α =ord(v) sinon α = 1
    cout_T = cout_T + α / k^prof(v)
  finpour
 finsi
finpour
Fin
```

IV. Algorithme d'alimentation de l'entrepôt de documents

//* ***AlimenterDW*** ()

Soit $D=\{d_1,d_2,.....d_n\}$ un ensemble de documents
Soit C l'ensemble des classes

```
Début
 réel α ; entier j ;
 pour chaque di de D
  Extraction de la vue spécifique Vsp puis Rep_d de di
  si card(C)=0 alors          //* créer la première classe représentée par C1
       C = {C1}          //* C1 représentée par Rep_d
     sinon
       α = 0 ;
    pour chaque Vg de Vgcand   //* pour chaque Vg candidate
       TransformationGraph(Rep_d,Vg) ;
       si (Separat(Vg)) alors
        si (Sim(Rep_d,Vg) > α) alors
              α = Sim(Rep_d,Vg)   //* Sim(Rep_d,Vg)=1-Dist_Graph(Rep_d,Vg)
        finsi    //*consever α et coût_T pour chaque Vg
       finsi
    finpour   //* max des α
    si ∃k / Sim(Rep_d,Vgk)=α>= Seuil_Sim alors  //* avec Vgk (coût_T min)
                        rattachement de Vsp à Vgk
                 sinon   j= card(C)
                                        //* créer une nouvelle classe Cj+1
                C = C ∪ {Cj+1}
```

```
        finsi
    finsi
finpour
            //* recaluler les classes
Fin
```

Algorithme *Dist_Chm*() : *qui permet de calcul la distance d'alignement entre deux chemins.*

```
réel  Dist_ Chm (Chemin chm_i,Chemin chm'_j)
réel d_1,d_2,SomPd = 0 ;

Début
    d_1=0 ;
pour chaque e de chm_i
    d_2 = P_e(e) ; SomPd = SomPd + P_e(e) ;
    pour chaque e' de chm'_j
        si (e  similaire à e') alors d_2 = |P_e(e)-P_e(e')|  //* quitter la boucle
    finpour
    d_1= d_1+ d_2 ;
finpour
retourner (d_1/ SomPd)    //* SomPd non nul car le chemin chm_i contient au moins un arc
Fin
```

Algorithme *Dist_Graph*() : *qui permet d'évaluer la distance entre deux graphes.*

```
réel  Dist_Graph(graph G, graph G')
Soit CH_G={chm_1, chm_2, ……,chm_n } l'ensemble de chemins de G,
Soit CH_G'={chm'_1, chm'_2, ……,chm'_n'} l'ensemble de chemins de G',
réel  d, d_GG', d_G'G ;

Début
d_GG'=0 ;
  pour chaque chm_i  de  CH_G            //* i ∈[1,n]
      d=1;
    pour chaque  chm'_k  de  CH_G'    //*k ∈[1,n']
        si (Dist_chm(chm_i, chm'_k) < d) alors d = Dist_chm(chm_i, chm'_k)
    finpour
     d_GG'= d_GG' + d;     //*d = min (Dist_chm(chm_i, chm'_k)) k ∈[1,n']
  finpour
   d_G'G=0 ;
  pour chaque chm'_i de  CH_G'
       d=1 ;
    pour chaque  chm_k  de  CH_G
        si (Dist_chm (chm'_i, chm_k) < d) alors d = Dist_chm (chm'_i, chm_k)
    finpour
    d_G'G= d_G'G + d   //*d = min (Dist_chm(chm'_i, chm_k)) k ∈[1,n]
  finpour
  retourner (d_GG'+ d_G'G)/2 ;
Fin
```

Pour vérifier la séparation d'une vue (générique) avec l'ensemble des vues génériques de l'entrepôt, nous proposons la fonction suivante :

boolean ***Separat****(graph V1)*

Début
  boolean View_sep=vrai
  pour chaque V2 de DW_g
    si (Dist_Graph(V1,V2) < d_{min} *et V1 ≠ V2) alors*
      *View_sep=faux //** d_{min} *paramètre fixé a priori par l'utilisateur*
        // quitter la boucle*
    finsi
  finpour
  return View_sep
Fin

V. Quelques événements de base de SAX

Parmi les événements de SAX, permettant de piloter un document XML, on peut citer :

- *startDocument()* : lié à l'ouverture du document,
- *startElement()* : lié à la rencontre d'un nouvel éléments de la structure du document,
- *characters()* : lié au contenu et qui permet de retourner les caractères rencontrés,
- *endElement()* : lié à la fin d'un élément de la structure du document,
- *endDocument()* : lié à la fin du document.

Exemple

Soit le document XML :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Labo>
  <libelle> IRIT, Toulouse France et IRF-SIC Agadir Maroc</libelle>
   <etudiant>
     <nom>A.Idarrou</nom>
     <sujet>entreposage de documents : comparaison de structures</sujet>
   </etudiant>
  <etudiant>……. </etudiant>
</Labo>
```

Cette séquence d'instructions va déclencher les événements suivants :

startDocument()
startElement() : « Labo »
characters() : « *IRIT, Toulouse France et IRF-SIC Agadir Maroc*»
startElement() : « etudiant »
startElement() : « nom »
characters() : « *A.Idarrou* »
startElement() : « sujet »
characters() : « *entreposage de documents : comparaison de structures* »
endElement () : «sujet»
endElement() : «nom»
endElement() : «etudiant»
……
endElement() : «Labo»

endDocument()

Exemple de document extrait du corpus de la bibliothèque de l'Université Toulouse 1 Capitole :

```
<?xml version="1.0" encoding="UTF-8"?>
<BiblioRecord Language="fre" id="PPN060710985">
  <Meta>
   <CreationDate Value="20020429"/>
   <TransactionDate Value=""/>
   <Status Value="c"/>
   <RecordType Value="a"/>
   <BibliographicLevel Value="m"/>
   <Completeness Value="0"/>
   <Origin Role="Issuer" System="">
     <Country>FR</Country>
     <Agency>Abes</Agency>
     <TransactionDate Value="20040330"/>
     <CataloguingRules>AFNOR</CataloguingRules>
   </Origin>
  </Meta>
  <Description>
   <LanguageInfo TitleScript="ba"/>
   <TitleAndResponsibility Significant="True">
     <Work>
      <TitleGroup>
        <Title>Palestra sagrada, o memorial de santos de Cordoba, con notas, y reflexione
      </TitleGroup>
     </Work>
   </TitleAndResponsibility>
   <IntellectualResponsibility>
     <PersonalName Role="Primary" FormOfName="Surname" AuthorityRecord="060711248">
      <Entry>Sanchez de Feria</Entry>
      <OtherPart>Bartolome</OtherPart>
      <Relationship>070</Relationship>
     </PersonalName>
     <PersonalName Role="Secondary" FormOfName="Surname" AuthorityRecord="060711353">
     <Entry>Rodriguez</Entry>
     <OtherPart>Juan</OtherPart>
     <NameAddition>libraire imprimeur</NameAddition>
     <Date>17 -17 ?</Date>
     <Relationship>610</Relationship>
    </PersonalName>
    <CorporateName Role="Secondary" Type="Corporate" AuthorityRecord="056535104">
     <Entry>Couvent des Capucins</Entry>
     <Subdivision>bibliothÃ¨que</Subdivision>
     <NameAddition>Toulouse, Haute-Garonne</NameAddition>
    </CorporateName>
  </IntellectualResponsibility>
  <PublicationGroup>
    <Publication>
     <Publisher>
       <Place>En Cordoba</Place>
       <Name>en la oficina de Juan Rodriguez</Name>
     </Publisher>
     <Date>1772</Date>
    </Publication>
  </PublicationGroup>
  <PublicationDate>
    <MonographDate Status="Uncertain">
     <Year>1772</Year>
     <Year>1772</Year>
    </MonographDate>
  </PublicationDate>
  <PhysicalDescription>
    <PhysicalItem>
     <ItemDescription>
       <Material>[28], 448, [8] p.</Material>
       <Dimensions>in-4</Dimensions>
       </ItemDescription>
      </PhysicalItem>
    </PhysicalDescription>
```

```
      </ItemDescription>
      <ItemDescription>
        <Material>[28], 448, [8] p.</Material>
        <Dimensions>in-4</Dimensions>
      </ItemDescription>
    </PhysicalItem>
  </PhysicalDescription>
 </Description>
 <CodedValues>
  <cdMonographic Illustration="a" FormOfContents="z" Conference="0" Festschrift="0" Index="1" Literature="
 </CodedValues>
 <Notes>
  <Note Type="General">Ouvrage en 4 volumes</Note>
  <Note Type="PhysicalDescription">Lettres ornÃ©es, bandeaux et culs-de-lampe gravÃ©s sur bois</Note>
  <CopyInHandNote Institution="315552102:150614241">Reliure en parchemin, lacets et fermoirs en perle</Cop
  <ProvenanceNote Institution="315552102:150614241">Cachet : BibliothÃ¨que des capucins de Toulouse</Prove
  <Note Type="Reproduction">Texte numÃ©risÃ© d'aprÃ¨s l'ex. Res Cap B 8621 SAN (1) de la BibliothÃ¨que uni
 </Notes>
 <Subjects>
  <TargetAudience Code="k"/>
  <TopicalName System="rameau" AuthorityRecord="027826279027520447027797376">
    <Entry>Saints</Entry>
    <TopicalSubdivision>Biographies</TopicalSubdivision>
    <TopicalSubdivision>Ouvrages avant 1800</TopicalSubdivision>
    <GeographicalSubdivision>Cordoue (Espagne)</GeographicalSubdivision>
  </TopicalName>
  <PlaceAccess>
  <OtherClass System="brp-sys">
    <Class>531AND</Class>
  </OtherClass>
 </Subjects>
 <LocalData/>
</BiblioRecord>
```

Figure 1 : Exemple de document XML extrait du corpus utilisé dans nos expérimentations

VI. Quelques interfaces de Multistructured Multimedia Document REPository

a) L'interface permettant le choix du *seuil de filtrage* :

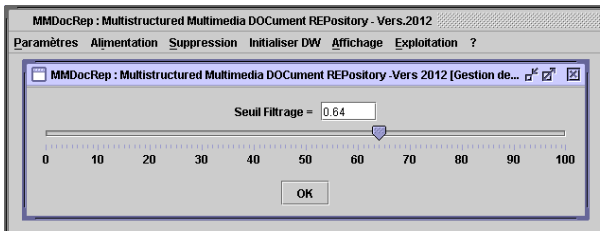


Figure 2 : Choix du seuil de filtrage : 64%

b) L'interface permettant le choix du *seuil de similarité* :

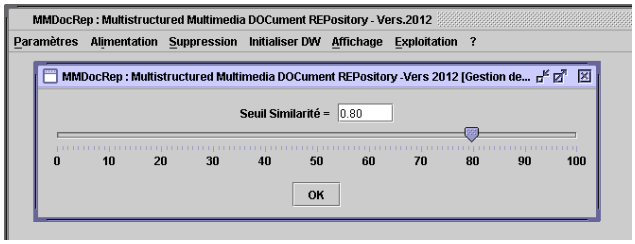


Figure 3 : Choix du seuil de similarité : 80%

c) L'interface qui permet l'intégration des documents :

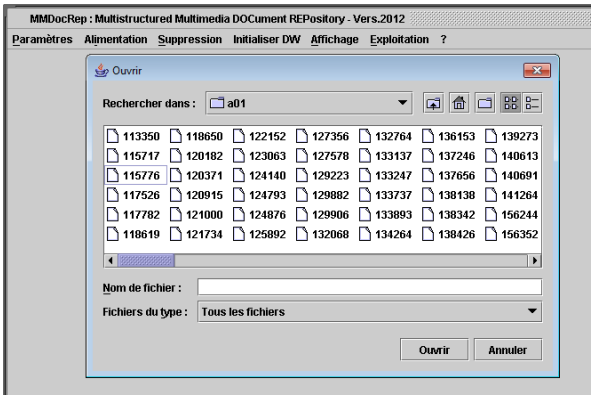


Figure 4 : Intégration d'un ensemble de documents

d) L'interface suivante montre les étapes d'alimentation de l'entrepôt de documents de l'intégration d'un document à la classification de celui-ci :

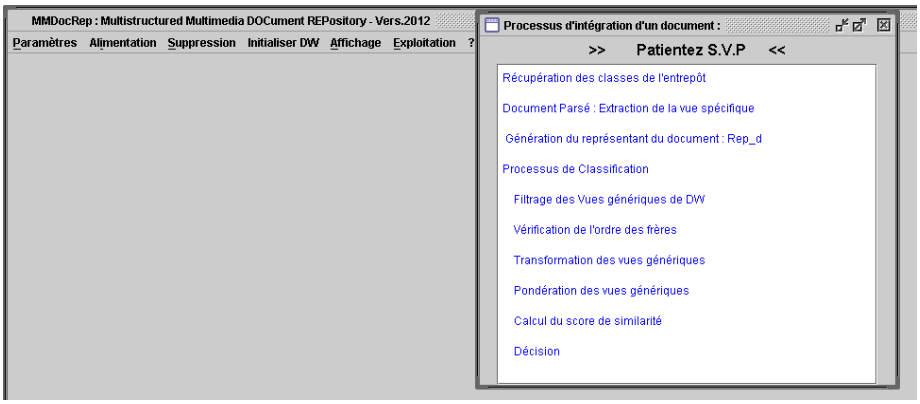


Figure 5 : Les étapes du processus de classification

e) Exemple de construction des classes (vues génériques)

Le représentant de la vue du premier document « doc1.xml » inséré dans la base sert de premier représentant de la première classe. Les classes sont ensuite construites par agrégation des documents structurellement proches :

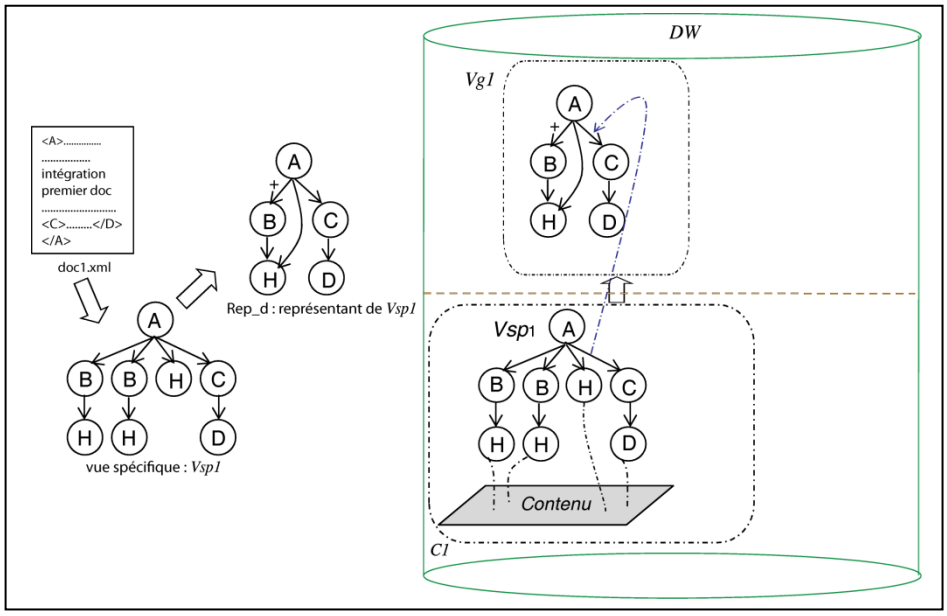


Figure 6 : Intégration du premier document

L'intégration du deuxième document « doc2.xml » nécessite la comparaison de son représentant *Rep_d* avec la vue générique Vg_1 (le représentant de la seule classe existante). La vue spécifique Vsp_2 est similaire à Vg_1, par conséquent Vsp_2 est rattachée à Vg_1 (figure 7). Dans cet exemple la vue générique a subi des transformations (Cf. chapitre III, section VI.2.3.1 page 123) : ajout des fragments *C/P et A/D*. La classe Vg_1 regroupe deux documents structurellement proches.

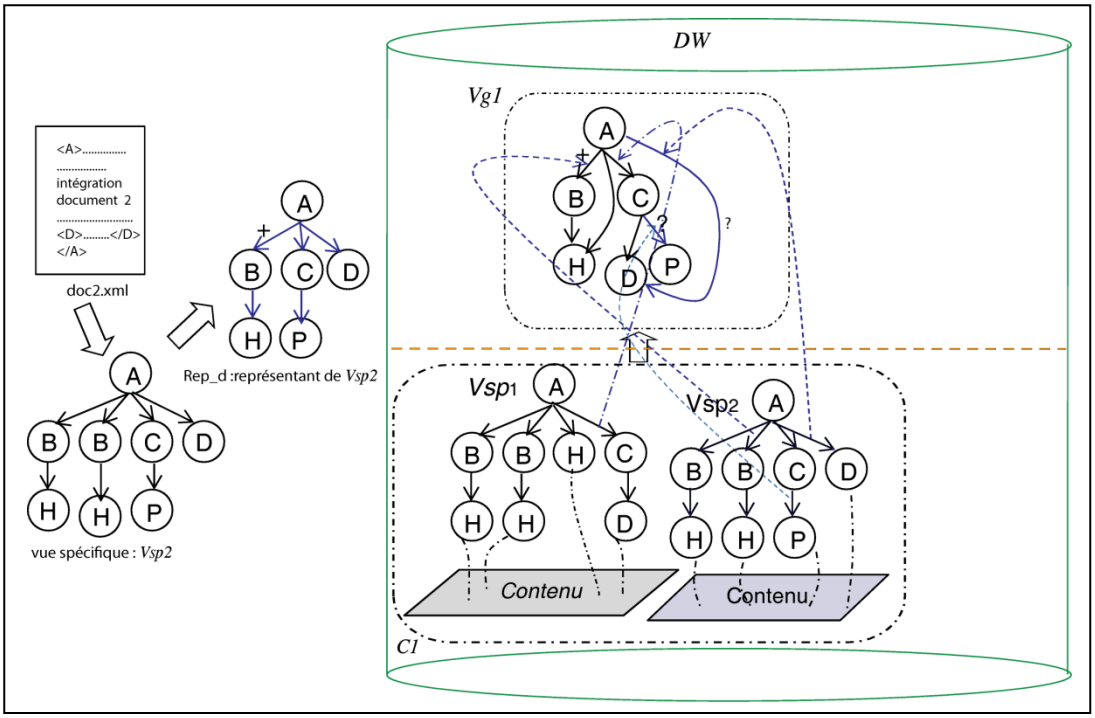


Figure 7 : Intégration du deuxième document

L'intégration du troisième document « doc3.xml » a suscité la création de la deuxième classe représentée par Vg_2. En effet le représentant *Rep_d* du document « doc3.xml » n'est

pas similaire à Vg_1, par conséquent une nouvelle classe Vg_2 (Cf. chapitre III, section IV.2.4 page 128) est crée (figure 8).

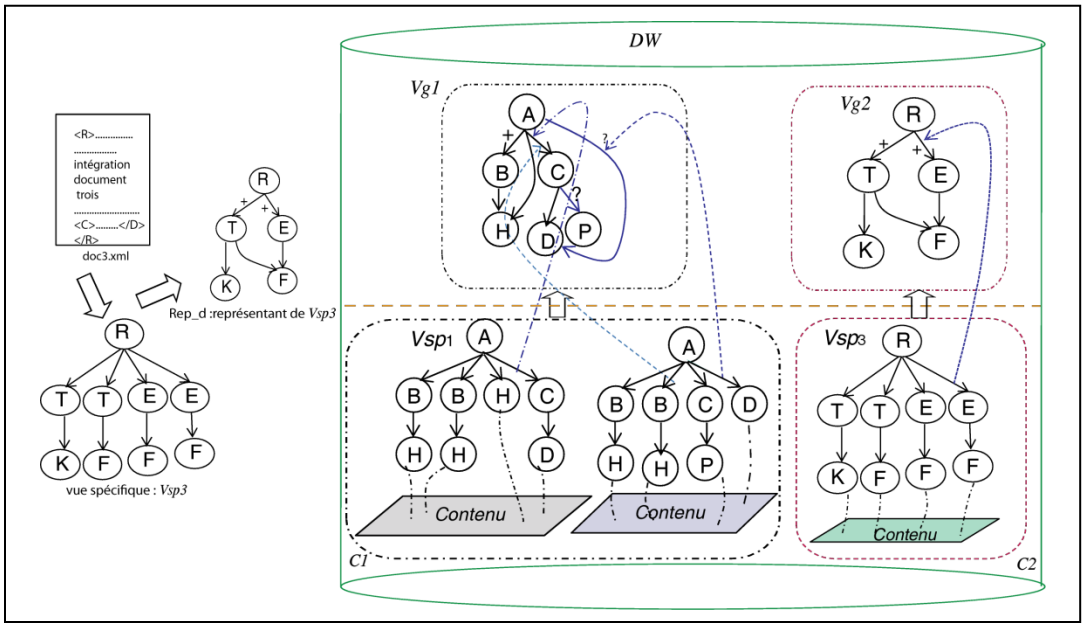


Figure 8 : Intégration du troisième document

Plus généralement, la construction des classes (vues génériques) sont construites d'une façon automatique au fur et à mesure par agrégation des documents structurellement proches. Les vues génériques sont représentées sous forme de graphes enracinés. Cette représentation est engendrée par une superposition d'arbores structurellement proches (ou identiques) rattachées à une même (vue générique) classe. Ces graphes sont enrichis au fur et à mesure de la classification (Cf. chapitre III, section 123 page 123).

f) L'interface suivante permet de sélectionner un module de restitution des résultats d'une classification (résultats d'une classification, liste des vues spécifiques par classe ou liste des vues spécifiques intégrées) :

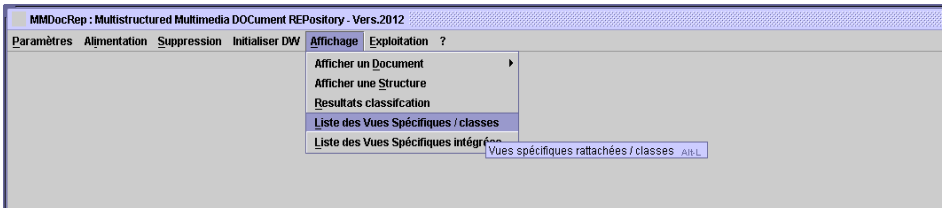


Figure 9 : Restitution des résultats

Le tableau suivant représente l'ensemble des vues spécifiques des documents intégrés :

Liste des vues spécifiques intégrées est 1606

NumVsp	VueSp	Nb Elt/Vsp	Nb Att/Vsp	Nb Chem/Vsp	ProfSp	Sim
1	C:\b1\139684.xml	10	11	7	5	1.0
2	C:\b1\139767.xml	9	10	6	5	0.98
3	C:\b1\139768.xml	8	7	6	5	0.98
4	C:\b1\139877.xml	12	3	7	6	0.87
5	C:\b1\139895.xml	8	8	6	5	0.86
6	C:\b1\140477.xml	9	10	6	5	0.86
7	C:\b1\140568.xml	17	17	17	5	1.0
8	C:\b1\140581.xml	12	8	11	5	0.86
9	C:\b1\140583.xml	16	12	12	7	0.86
10	C:\b1\141441.xml	9	8	7	5	0.96
11	C:\b1\141665.xml	11	10	8	5	0.94
12	C:\b1\142065.xml	9	8	7	5	1.0
13	C:\b1\142203.xml	10	9	8	5	0.98
14	C:\b1\142220.xml	8	5	7	5	0.86
15	C:\b1\142232.xml	11	6	8	6	0.87
16	C:\b1\142264.xml	10	9	7	5	1.0
17	C:\b1\142533.xml	11	2	6	5	0.96
18	C:\b1\142609.xml	9	8	7	5	0.99
19	C:\b1\142610.xml	10	9	7	5	0.99
20	C:\b1\142619.xml	10	9	7	5	0.99
21	C:\b1\142661.xml	10	9	7	5	0.99
22	C:\b1\142725.xml	9	11	7	4	0.86
23	C:\b1\142931.xml	11	10	8	5	0.94
24	C:\b1\143063.xml	16	6	9	5	0.86
25	C:\b1\143175.xml	8	8	7	4	0.86
26	C:\b1\143530.xml	9	9	10	5	0.86
27	C:\b1\143755.xml	15	16	10	6	0.92
28	C:\b1\143860.xml	10	7	4	4	0.95
29	C:\b1\14418.xml	8	5	7	5	0.95
30	C:\b1\144199.xml	11	13	8	4	0.94
31	C:\b1\144320.xml	12	8	11	7	0.96
32	C:\b1\144328.xml	15	15	12	7	0.86
33	C:\b1\144683.xml	9	6	8	5	1.0
34	C:\b1\144687.xml	11	6	7	6	0.95
35	C:\b1\144955.xml	8	4	6	5	0.86
36	C:\b1\145173.xml	11	17	7	4	0.95
37	C:\b1\145235.xml	10	11	14	5	0.94
38	C:\b1\145333.xml	11	14	10	5	0.86
39	C:\b1\145411.xml	10	11	7	4	0.95
40	C:\b1\145588.xml	13	17	10	5	0.97
41	C:\b1\145819.xml	17	8	9	6	0.95
42	C:\b1\145861.xml	11	16	8	6	0.96
43	C:\b1\146031.xml	13	6	8	6	0.96

Figure 10: Liste des vues spécifiques des documents intégrés