Extraction d'informations textuelles au sein de documents numérisés : cas des factures

Cynthia Pitou

THÈSE

Présentée pour obtenir

LE GRADE DE DOCTEUR EN
INFORMATIQUE
DE L'UNIVERSITÉ DE LA REUNION

par

Cynthia PITOU

Extraction d'informations textuelles au sein de documents numérisés: cas des factures

Soutenue le 28 Septembre 2017 devant les membres du Jury:

Professeur	François BRUCKER	(Rapporteur)
Professeur	Pascale KUNTZ	(Rapporteur)
Professeur	Jean DIATTA	(Directeur de thèse)
Professeur	Frédéric MESNARD	(Examinateur)
Professeur	André TOTOHASINA	(Examinateur)

Thèse préparée au
Laboratoire d'Informatique et de Mathématiques (EA2525)
Parc Technologique Universitaire
Bâtiment 2
2 rue Joseph Wetzell
97 490 Sainte Clotilde

Résumé

Le traitement automatique de documents consiste en la transformation dans un format compréhensible par un système informatique, de données présentes au sein de documents et compréhensibles par l'Homme. L'analyse de document et la compréhension de documents sont les deux phases du processus de traitement automatique de documents. Étant donnée une image de document constituée de mots, de lignes et d'objets graphiques tels que des logos, l'analyse de documents consiste à extraire et isoler les mots, les lignes et les objets, puis à les regrouper au sein de blocs. Les différents blocs ainsi formés constituent la structure géométrique du document. La compréhension de documents fait correspondre à cette structure géométrique une structure logique en considérant des liaisons logiques (à gauche, à droite, au-dessus, en-dessous) entre les objets du document. Un système de traitement de documents doit être capable de : (i) localiser une information textuelle, (ii) identifier si cette information est pertinente par rapport aux autres informations contenues dans le document, (iii) extraire cette information dans un format compréhensible par un programme informatique. Pour la réalisation d'un tel système, les difficultés à surmonter sont liées à la variabilité des caractéristiques de documents, telles que le type (facture, formulaire, devis, rapport, etc.), la mise en page (police, style, agencement), la langue, la typographie et la qualité de numérisation du document.

Dans ce mémoire, nous considérons en particulier des documents numérisés, également connus sous le nom d'images de documents. Plus précisément, nous nous intéressons à la localisation d'informations textuelles au sein d'images de factures, afin de les extraire à l'aide d'un moteur de reconnaissance de caractères. Les factures sont des documents très utilisés mais non standards. En effet, elles contiennent des informations obligatoires (le numéro de facture, le numéro siret de l'émetteur, les montants, etc.) qui, selon l'émetteur, peuvent être localisées à des endroits différents. Les contributions présentées dans ce mémoire s'inscrivent dans le cadre de la localisation et de l'extraction d'informations textuelles fondées sur des régions identifiées au sein d'une image de document.

Tout d'abord, nous présentons une approche de décomposition d'une image de documents en sous-régions fondée sur la décomposition quadtree. Le principe de cette approche est de décomposer une image de documents en quatre sous-régions, de manière récursive, jusqu'à ce qu'une information textuelle d'intérêt soit extraite à l'aide d'un moteur de reconnaissance de caractères. La méthode fondée sur cette approche, que nous proposons, permet de déterminer efficacement les régions contenant une information d'intérêt à extraire.

Dans une autre approche, incrémentale et plus flexible, nous proposons un système d'extraction d'informations textuelles qui consiste en un ensemble de régions prototypes et de chemins pour parcourir ces régions prototypes. Le cycle de vie de ce système comprend cinq étapes :

1. Construction d'un jeu de données synthétiques à partir d'images de factures réelles contenant les informations d'intérêts.
2. Partitionnement des données produites.
3. Détermination des régions prototypes à partir de la partition obtenue.
4. Détermination des chemins pour parcourir les régions prototypes, à partir du treillis de concepts d'un contexte formel convenablement construit.
5. Mise à jour du système de manière incrémentale suite à l'insertion de nouvelles données.

Mots-clefs : traitement automatique de documents ; extraction d'informations textuelles ; classification supervisée ; classification non supervisée ; décomposition quadtree ; analyse formelle de concepts ; treillis de concepts.

Abstract

Document processing is the transformation of a human understandable data in a computer system understandable format. Document analysis and understanding are the two phases of document processing. Considering a document containing lines, words and graphical objects such as logos, the analysis of such a document consists in extracting and isolating the words, lines and objects and then grouping them into blocks. The subsystem of document understanding builds relationships (to the right, left, above, below) between the blocks. A document processing system must be able to: locate textual information, identify if that information is relevant comparatively to other information contained in the document, extract that information in a computer system understandable format. For the realization of such a system, major difficulties arise from the variability of the documents characteristics, such as: the type (invoice, form, quotation, report, etc.), the layout (font, style, disposition), the language, the typography and the quality of scanning.

This work is concerned with scanned documents, also known as document images. We are particularly interested in locating textual information in invoice images. Invoices are largely used and well regulated documents, but not unified. They contain mandatory information (invoice number, unique identifier of the issuing company, VAT amount, net amount, etc.) which, depending on the issuer, can take various locations in the document. The present work is in the framework of region-based textual information localization and extraction.

First, we present a region-based method guided by quadtree decomposition. The principle of the method is to decompose the images of documents in four equals regions and each regions in four new regions and so on. Then, with a free optical character recognition (OCR) engine, we try to extract precise textual information in each region. A region containing a number of expected textual information is not decomposed further. Our method allows to determine accurately in document images, the regions containing text information that one wants to locate and retrieve quickly and efficiently.

In another approach, we propose a textual information extraction model consisting in a set of prototype regions along with pathways for browsing through these prototype regions. The life cycle of the model comprises five steps:

1. Produce synthetic invoice data from real-world invoice images containing the textual information of interest, along with their spatial positions.
2. Partition the produced data.
3. Derive the prototype regions from the obtained partition clusters.
4. Derive pathways for browsing through the prototype regions, from the concept lattice of a suitably defined formal context.
5. Update incrementally the set of protype regions and the set of pathways, when one has to add additional data.

Keywords : automatic document processing ; textual iformation extraction ; supervised learning ; cluster analysis ; quadtree decomposition ; formal concept analysis ; concept lattice.

Je dédie ce mémoire à mon époux, à notre fille
et à ma grand-mère Céline (out kèr y rèt koté mwen).

Remerciements

Mes remerciements vont en premier lieu au Professeur Jean Diatta qui, en sa qualité de directeur de thèse, a été tout au long de ces années, un superviseur et un guide bienveillant pendant toutes les phases de la thèse. Je le remercie de m'avoir offert cette opportunité et d'avoir vu en moi quelqu'un capable de réussir. Ses conseils, ses propositions et sa rigueur m'ont menée jusqu'à l'objectif atteint aujourd'hui. J'espère que ces quelques mots suffiront à lui exprimer toute ma gratitude et mon respect.

Je remercie le Professeur Pascale Kuntz et le professeur François Brucker d'avoir accepté d'être rapporteurs de mes travaux. Je les remercie pour leurs commentaires avisés. Je suis très honorée que mon travail soit soumis au jugement de personnalités ayant autant œuvré, chacun dans son domaine, pour la recherche scientifique.

Je remercie également le Professeur André Totohasina et le Professeur Frédéric Mesnard d'avoir accepté de faire partie du jury. Je remercie également les membres du comité de suivi de thèse pour leur investissement. J'adresse ainsi toute ma reconnaissance à Messieurs Engelbert Mephu Nguifo, Patrice Bertrand et Régis Girard, ainsi qu'aux membres du jury.

Je remercie toute l'équipe du laboratoire pour leur contribution au déroulement de la thèse dans de bonnes conditions. Je remercie en particulier, Xavier Nicolay, Magalie Clain et Gisèle Ulderic.

Ce travail a été réalisé en étroite collaboration avec la société Groupe Austral Assistance qui est à l'origine de la problématique de la thèse. Je remercie de ce fait son président directeur général, en la personne du Docteur Philippe Bellard. Je le remercie, avec une profonde émotion, d'être ce qu'il est chaque jour pour tous ses salariés : un patron sans égal.
Je remercie également Julien Mauras dont les qualités en font un directeur et une personne d'exception, un exemple de générosité, d'écoute et de bienveillance. Cette thèse n'aurait tout simplement pas vu le jour sans lui. Je n'ai fait que me dépasser et m'élever toujours plus haut face aux défis à ses côtés.

Je n'oublie pas les enseignants-chercheurs de l'Université de La Réunion qui ont contribué à mon épanouissement intellectuel dans des thématiques diverses et variées. Leurs enseignements m'ont guidée vers cette thèse et vers une vie professionnelle riche et épanouissante. Je remercie David Grosser, Frédéric Mesnard, Etienne Payet, Philippe Martin, Anil Cassam-Chenai, Olivier Sébastien et bien d'autres encore.

J'ai une pensée particulière pour le Professeur Bernard Fichet et le Professeur Henry Ralambondrainy dont la présence en congrès a égayé mon séjour.

La thèse est un long chemin sur lequel on s'isole parfois de sa famille et où on leur accorde moins de temps. J'ai la chance d'être soutenue par un époux doué de patience et de bienveillance à mon égard. Son soutien a été sans faille et je lui en remercie. Je ne serais pas arrivée jusqu'au bout sans son intelligence et son amour.

Que ce soit là le début de quelque chose d'encore plus grand.

Table des matières

Introduction générale **16**

I État de l'art **24**

1 Traitement automatique de documents **26**

1.1 Structures de documents 28

1.1.1 Structure géométrique 29

1.1.2 Structure logique 30

1.2 Analyse de documents 30

1.2.1 Méthodes hiérarchiques 32

1.2.2 Méthodes non-hiérarchiques 36

1.3 Compréhension de documents 37

1.3.1 Approches fondées sur des grammaires 38

1.3.2 Approches à base de règles 39

1.3.3 Approches fondées sur des techniques d'apprentissage automatique 41

1.3.4 Autres approches 41

1.4 Évaluation des performances des systèmes de traitement automatique de documents 43

1.5 Conclusion 44

2 Extraction d'informations textuelles au sein d'images de documents **46**

2.1 Caractéristiques inhérentes au texte contenu au sein d'images 47

2.2 Approches d'extraction d'informations textuelles 50

2.2.1 Approches fondées sur des régions 51

2.2.2 Approches fondées sur les contours 52

2.2.3 Approches morphologiques 53

2.2.4 Approches fondées sur les textures . . . 53
2.3 Approches d'extraction d'informations textuelles au sein d'images de factures . . . 54
2.4 Problématiques d'évaluation des systèmes d'extraction d'informations textuelles . . . 59
2.4.1 Problématiques générales de création de jeux de données . . . 60
2.4.2 Avantages et inconvénients des données réelles et des données synthétiques . . . 62
2.4.3 Mesures pour l'évaluation de performance . . . 62
2.5 Conclusion . . . 63

3 Classification **64**
3.1 Classification supervisée . . . 64
3.1.1 Arbres de décision . . . 66
3.1.2 Réseaux bayésiens . . . 69
3.1.3 Réseaux de neurones artificiels . . . 72
3.1.4 Discussion . . . 77
3.2 Classification non supervisée . . . 79
3.2.1 Méthodes hiérarchiques . . . 80
3.2.2 Méthodes de partitionnement . . . 84
3.3 Conclusion . . . 88

4 Analyse formelle de concepts **90**
4.1 Notions de base . . . 91
4.2 AFC et fouille de règles d'association . . . 94
4.3 AFC et recherche d'informations . . . 96
4.4 Diverses autres applications de l'AFC . . . 97
4.5 Conclusion . . . 100

II Contributions **102**

5 Extraction d'informations textuelles au sein d'images de factures basée sur la décomposition quadtree **104**
5.1 La décomposition quadtree . . . 104

5.2 Notre approche de décomposition quadtree pour l'extraction d'informations textuelles au sein d'images de factures . . . 106
5.3 Évaluation expérimentale . . . 109
5.4 Conclusion . . . 113

6 Système d'extraction d'informations textuelles au sein d'images de factures 116

6.1 Création du jeu de données synthétiques . . . 117
6.2 Classification des données synthétiques et détermination des régions prototypes . . . 123
6.2.1 Classification des données synthétiques . . . 124
6.2.2 Détermination des régions prototypes . . . 127
6.3 Détermination de chemins pour naviguer au sein des régions prototypes à partir d'un treillis de concepts . . . 128
6.3.1 Construction du treillis de concepts . . . 128
6.3.2 Détermination de chemins à partir du treillis de concepts . . . 129
6.4 Traitement d'une image de facture inconnue fondé sur des modèles incrémentaux . . . 135
6.4.1 Mise à jour de l'ensemble de régions prototypes . . . 136
6.4.2 Mise à jour des chemins pour naviguer au sein de l'ensemble des régions prototypes . . . 139
6.5 Conclusion . . . 142

7 Extraction d'informations textuelles au sein d'images de factures à l'aide de notre système 146

7.1 Processus d'extraction d'informations textuelles . . . 147
7.2 Évaluation expérimentale de notre système pour l'extraction d'informations textuelles au sein d'images de factures . . . 149
7.2.1 Évaluation expérimentale de notre système pour la localisation et l'extraction d'informations textuelles . . . 150
7.2.2 Évaluation expérimentale de notre stratégie de parcours des régions prototypes guidée par les chemins . . . 153
7.2.3 Évaluation des modèles de mise à jour incrémentale des régions prototypes et des chemins pour naviguer dans l'ensemble des régions prototypes . . . 153
7.3 Conclusion . . . 155

Bilan et perspectives 158

Annexes 162

A Exemple de facture émise par une société de location 162

B Exemple de facture émise par une société de remorquage 163

C Exemple de facture émise par une société de dépannage 164

D Exemple de facture émise par une société de taxi 165

Bibliographie 166

Table des figures

1.1 Schéma du modèle de base de traitement de documents proposé par (Tang *et al.*, 1996). 27

1.2 Structure géométrique et structure logique d'une image de facture. 31

1.3 Dendrogramme produit par l'algorithme DTMSER. Les nœuds feuilles correspondent aux composants connectés de l'image, alors que les regroupements de nœuds dépendent uniquement de la distance entre régions, ce qui donne lieu à des groupes sémantiquement pertinents. 35

1.4 Exemples d'images de documents ayant une structure géométrique complexe. 37

2.1 Exemples d'images contenant du texte. 48

2.2 Processus d'extraction d'informations textuelles au sein d'images. 51

3.1 Processus général d'apprentissage supervisé. 65

3.2 Arbre de décision obtenu à partir des données d'entrainement de la Figure 3.1. 67

3.3 Représentation d'un neurone formel. 73

3.4 Exemple de perceptron multicouche élémentaire avec une couche cachée et une couche de sortie. 75

3.5 Exemple de réseau de neurones pour la classification d'un ensemble de documents. 77

3.6 Dendrogramme de la hiérarchie indicée (H_E, v) où H_E est la hiérarchie de l'exemple 1 et v l'indice de niveau de l'exemple 2. 80

3.7 Arbre hiérarchique indicé associé à H_E. 81

4.1 Diagramme de Hasse du treillis de Galois correspondant au contexte formel du Tableau 6.8 92

5.1 Un quadtree. 106

5.2 Principe de décomposition en quatre régions adopté dans l'algorithme proposé. 109

5.3 Exemple de quadtree obtenu en appliquant notre algorithme sur une image de facture. . . . 110

5.4 Exemple de fichier XML obtenu en sortie de notre algorithme de décomposition. . . . 111

6.1 Vue d'ensemble du système d'extraction d'informations textuelles au sein d'images de documents. . . . 118

6.2 Interface graphique du programme JAVA de création d'images de factures synthétiques. . . . 120

6.3 Zone 1 de l'interface graphique de la Figure 6.2. . . . 121

6.4 Zone 2 de l'interface graphique de la Figure 6.2. . . . 121

6.5 Zone 3 de l'interface graphique de la Figure 6.2. . . . 121

6.6 A droite une image de facture réelle. A gauche une image de facture synthétique générée par notre programme. . . . 122

6.7 Représentation des 3 régions prototypes relatives à l'information I4 dans un repère orthonormé. . . . 128

6.8 Treillis de concepts du contexte formel d'images de factures. . . . 130

6.9 Sous-partie du treillis de concepts de la Figure 6.8. Les arrêtes apparaissant en bleu représentent deux règles d'association approximatives de la base de Luxenburger. . . . 135

7.1 Exemple de parcours d'un chemin (une séquence de nœuds) du treillis de concepts du contexte formel d'images de factures (Tableau 6.8). . . . 149

7.2 Rappel et précision obtenus pour chaque type d'extraction réalisé sur les 4 ensembles de 200, 400, 800 et 1000 échantillons d'images de factures réelles respectivement. . . . 152

Liste des tableaux

3.1 Données d'entrainement pour la construction de l'arbre de décision de la Figure 3.2. 66

3.2 Paramètres d'apprentissage du réseau. 70

3.3 Tableau Comparatif d'algorithmes de classification supervisée extrait de l'étude de (Kotsiantis *et al.*, 2007) : **** 4 étoiles représentent la meilleure performance et * une étoile représente la moins bonne performance. . . . 78

3.4 Exemples de distance utilisées en classification de données quantitatives. 82

3.5 Exemples de distance utilisées en classification de données qualitatives. . 83

4.1 Exemple de contexte formel. 92

5.1 Liste des expressions régulières correspondant à chacune des cinq informations recherchées. 112

5.2 Distribution originale du corpus d'images de factures de test. 113

5.3 Résultats 113

6.1 Distribution originale du corpus d'images de factures réelles. 117

6.2 Exemples de données du jeu de données D1 relatif à l'information I1 (a) et du jeu de données D5 relatif à l'information I5 (b). 124

6.3 Valeurs de k optimales relevées dans le cas de la classification avec CAH, k-means et k-medoids selon une vue globale. 125

6.4 Valeurs de k optimales relevées dans le cas de la classification selon cinq vues indépendantes des jeux de données D1, D2, D3, D4 et D5 en appliquant les trois méthodes de classification : k-means, k-medoids et CAH. . 126

6.5 Valeurs de Silhouette et de Calinski-Harabasz relevées pour une classification du jeu de données selon une vue globale en k=10 et k=13 classes. 126

6.6 Nombre de classes retenu pour le partitionnement des jeux de données D1, D2, D3, D4 et D5. 127

6.7 Valeurs relevées pour les indices de qualité Silhouette, Calinski-Harabasz et PBM, des partitions obtenues avec k-means dans le cas de la classification selon cinq vues indépendantes. Les valeurs de k utilisées pour la classification de chaque jeu de données Di sont présentées dans le Tableau 6.6. 127

6.8 Extrait du contexte formel de l'ensemble des images de factures du jeu de données synthétiques. 129

6.9 Ensemble des règles d'association approximatives de la base de Luxenburger du treillis de concepts dérivé du contexte formel d'images de factures synthétiques (1ère partie). 131

6.10 Ensemble des règles d'association approximatives de la base de Luxenburger du treillis de concepts dérivé du contexte formel d'images de factures synthétiques (2ème partie). 132

6.11 Ensemble des chemins déterminés à partir de la base de Luxenburger du treillis de concepts dérivé du contexte formel d'images de factures synthétiques (1ère partie). 133

6.12 Ensemble des chemins déterminés à partir de la base de Luxenburger du treillis de concepts dérivé du contexte formel d'images de factures synthétiques (2e partie). 134

7.1 Distribution originale du corpus d'images de factures réelles. 150

7.2 Résultats obtenus pour l'extraction des informations textuelles I1 à I5 au sein de l'ensemble de 1000 échantillons d'images de factures réelles. . . . 151

7.3 Résultats obtenus pour l'extraction des informations textuelles I1 à I5 au sein d'un ensemble de 100 échantillons d'images de factures réelles. 154

Introduction générale

Contexte

Cette thèse s'inscrit dans le cadre d'une convention de collaboration de recherche entre l'Université de La Réunion et la société Groupe Austral Assistance [1] (GAA). GAA a été créé il y a 20 ans à l'île de La Réunion. La société exerce son activité dans le domaine de la prestation de services et de la gestion déléguée pour compte de tiers. C'est "un plateau de gestion de dernière génération, basé sur l'utilisation étendue des T.I.C (Technologie de l'Information et de la Communication), avec centres d'appels intégrés". GAA est spécialisé dans le traitement des "Services Clients Spécifiques" pour des partenaires professionnels. En tant que prestataire de services, GAA intervient dans le domaine des Services Immédiats, des Services Clients, et de la Gestion Déléguée. En particulier, le concept du service immédiat consiste à apporter en permanence une écoute, une information, une recommandation, ou une aide rapide, efficace et adaptée, à toute personne qui se trouve confrontée brutalement à un événement inhabituel. Ses services sont principalement destinés aux constructeurs automobiles, aux assureurs, aux établissements bancaires, aux sociétés d'assistance et plus généralement à toutes les sociétés désireuses d'apporter des services spécifiques à une partie significative de leur clientèle.
L'entreprise a débuté son activité dans la Zone Océan Indien qui regroupe les pays Francophones de la région, à savoir : La Réunion, Maurice, Madagascar, Mayotte, les Comores et les Seychelles. A la suite des demandes de ses principaux partenaires, et pour répondre à l'évolution croissante des besoins et des technologies en Outre Mer, GAA a créé de nouvelles implantations dans les Régions Ultra Périphériques de la C.E.E.. Ce développement s'est largement appuyé sur le savoir-faire et les compétences acquises à La Réunion. Ainsi, GAA a ouvert, en 2003, un premier bureau aux Antilles avec l'installation d'une plate-forme opérationnelle en Martinique. Celle-ci a été complétée un an plus tard, fin 2004, par la plate-forme de Guadeloupe. En 2006, GAA s'est implanté dans le Pacifique avec la création d'une plate-forme opérationnelle à Nouméa. Grâce à ses 4 implantations, l'entreprise dispose d'une couverture complète sur l'ensemble de l'Outre Mer Français. Plus récemment, en 2015, GAA s'est implanté en Afrique sub-saharienne avec la création de deux plates-formes, au Cameroun et en Côte d'Ivoire. L'utilisation des implantations de GAA permet à ses partenaires de sous-traiter totalement les phases de gestion opérationnelle, administrative et comptable des prestations d'assistance, externalisant ainsi un service aux contraintes particulièrement exigeantes et lourdes. Le déploiement des nouveaux services s'accompagne non seulement d'une amélioration de la qualité, mais également d'un transfert de savoir-faire avec formation des personnels recrutés localement.

1. www.gaa.fr

Présentation de Groupe Austral Assistance

Ses compétences

GAA met à la disposition de ses partenaires un panel complet de prestations de services dans les domaines de l'automobile, de la santé, de l'habitation, du médical, de la gestion, de l'e-business, de l'informatique et de la téléphonie.
GAA intervient depuis plus de 15 ans, sous le nom d'Océan Indien Assistance dans le cadre de prestations d'assistance médicale. L'activité consiste à opérer dans toutes prestations de service effectuées ou mise en place à la demande d'un *Client-Donneur d'Ordres* et pouvant être utiles à un usager victime d'un incident de santé lors d'un déplacement. Les prestations de service sont de plusieurs types :

- accompagnement médicalisé et non médicalisé,
- transfert vers La Réunion, l'Europe ou tout autre pays,
- billetterie aérienne,
- affrètement d'avions privés,
- régulation médicale,
- conseil sur les équipements et les potentiels des établissements hospitaliers de la zone,
- visite régulière des établissements hospitaliers de la zone,
- médicalisation de sites à la demande,
- avance de fonds,
- réservation et prise en charge de frais hôteliers,
- mise à disposition de matériel médical.

Dans le cadre de l'assistance automobile GAA opère dans des prestations de service telles que :

- dépannage et remorquage,
- location de véhicule,
- taxi,
- suivi de réparations automobiles.

Ce vaste panel d'activités a conduit GAA à se doter d'un réseau de milliers de prestataires référencés à La Réunion, aux Antilles, en Guyane, en Nouvelle Calédonie et en Afrique.

Son système d'information

GAA dispose de baies informatiques sur chaque zone géographique où la société est présente. Tous les plateaux sont inter-connectés par des liaisons spécialisées pour les échanges de données (data) mais également pour les communications téléphoniques (VoIP). Le système d'informations est composé de :

- plusieurs baies informatiques virtualisées,
- serveurs répartis sur chaque site,
- standards téléphoniques inter-connectés en VoIP,
- plus de 70 postes de travail (informatique et téléphonique),
- dispositifs de réplication en temps réel des données,
- dispositifs de supervision et de monitoring H24,

— dispositifs de backup et de bascule d'activité entre sites.

Pour garantir un service fiable, sécurisé et performant, l'entreprise a développé des outils informatiques basés sur les technologies les plus récentes et a recours aux dernières innovations de la communication. Ainsi le développement d'un applicatif informatique commun à tous les sites, offre l'intégralité des outils nécessaires au suivi opérationnel, à la gestion et à la comptabilité. L'applicatif informatique développé par GAA a été baptisé LiSA, pour Logiciel interactif de Suivi d'Assistance. Les caractéristiques fonctionnelles de LiSA sont les suivantes :

— intégration des données clients (EDI),
— gestion de la relation client de ses différents partenaires avec croisement de données,
— gestion opérationnelle des dossiers d'assistance,
— conseil et diagnostic en ligne,
— gestion électronique des documents (GED),
— création de rapports statistiques et de tableaux de bord d'aide au pilotage des activités,
— mise à disposition d'un extranet pour le suivi d'activité,
— extraction de données comptables et opérationnelles (afin d'être injecté en fin de chaîne dans les systèmes d'informations des donneurs d'ordres),
— gestion comptable avec intégration des factures prestataires.

Problématique de la thèse

Bien que le numérique se soit considérablement développé ces dernières décennies, dans un contexte d'entreprise, la réception de documents papiers est encore très généralisée (factures, courriers de réclamation, contrats clients, demande d'indemnisation client, etc.). Dans le cadre de sa gestion comptable avec intégration des factures prestataires (à l'aide du logiciel LiSA), la problématique principale de GAA est le traitement des factures. Actuellement, le traitement des factures générées par les activités de l'entreprise fait partie d'un workflow semi-automatisé. A la réception des factures par courriers, des opérateurs numérisent ces factures à l'aide d'un scanner et enregistrent manuellement les données présentes sur les factures (date de facture, numéro de facture, montants, etc.) dans une base de données, via le logiciel LiSA, qui s'occupe également de les traiter (gestion comptable, règlement, remboursement, etc.). GAA a traité pour l'année 2016, plus de 100 000 factures par ce procédé semi-automatisé. Ces documents sont divers et variés de par leur mise en page et leur présentation. Les factures sont des documents particuliers car très utilisés mais non standards. Elles contiennent des informations obligatoires (numéro de facture, date de facture, identifiant siret de la société émettrice, montants, etc.) qui selon l'émetteur peuvent se trouver à des endroits différents dans le document. L'hétérogénéité des mises en page adoptées pour ces documents constitue une difficulté majeure pour les systèmes de traitement automatique de ce type de documents. Des solutions commerciales existent sur le marché. Ces solutions sont connues pour être performantes, mais sont couteuses et demandent des paramétrages plus ou moins lourds pour être intégrées au sein d'une entreprise. L'attente de GAA est de disposer de sa propre solution pour automatiser l'ensemble du processus de traitement des factures, et plus particulièrement la tâche qui concerne la collecte d'informations depuis une facture

papier. L'objectif est de supprimer la tâche de saisie manuelle des informations présentes dans les factures, cette tâche étant peu valorisante, chronophage et sans valeur ajoutée. L'automatisation a également pour but de soutenir la société dans la gestion de son volume croissant de factures.
Dans le contexte posé par l'entreprise, l'objectif de la thèse est de proposer et de développer une méthode incrémentale de classification de documents numérisés, en vue, notamment, d'améliorer la performance d'outils d'extraction d'informations pertinentes à partir de ces documents.

Contributions de la thèse

En réponse à la problématique industrielle posée par GAA, les contributions de cette thèse sont les suivantes :

1. Conception d'une méthode, pour l'extraction d'informations textuelles au sein de documents, fondée sur la décomposition quadtree

Nous avons élaboré une méthode de décomposition d'images de factures fondée sur la décomposition quadtree et permettant d'extraire des informations textuelles ciblées à l'aide d'un moteur de reconnaissance optique de caractères (OCR).
Notre approche décompose une image de factures, de manière récursive, en quatre sous-régions rectangulaires identiques, afin d'y extraire une information textuelle ciblée, à l'aide d'un moteur d'OCR. A partir du texte brut extrait par le moteur d'OCR, des expressions régulières particulières sont utilisées afin d'identifier de manière précise l'information ciblée à extraire.

2. Conception d'un système de traitement de factures dans lequel les interventions humaines sont limitées

Le cycle de vie du système que nous avons conçu est constitué de 5 processus :

1. Le 1er processus construit le jeu de données initial à partir duquel le système va travailler. En effet via une interface graphique d'acquisition de données, que nous avons développée, un utilisateur peut renseigner les coordonnées spatiales des rectangles contenant des informations cibles au sein d'images de documents.
2. Le 2e processus réalise un partitionnement du jeu de données par l'algorithme *k-means* et construit un ensemble de *régions prototypes* en calculant, pour chaque classe de la partition obtenue, le plus petit rectangle couvrant les rectangles de cette classe.
3. Le 3e processus détermine une liste de chemins pour parcourir efficacement les régions prototypes obtenues à l'étape précédente, en s'appuyant sur l'ensemble des règles approximatives de la base de Luxenburger du treillis de concepts dérivé d'un contexte formel convenablement construit.

4. Le 4e processus extrait au sein d'images de documents candidates les informations textuelles cibles à l'aide d'un moteur de reconnaissance optique de caractère.
5. En cas d'échec du 4e processus, le 5e processus du système requiert l'intervention d'un utilisateur, afin d'enrichir le jeu de données initial avec les positions des rectangles contenant les informations non extraites. Se déclenche alors une mise à jour incrémentale des différents modèles mis en jeu dans la chaine de processus.

Ainsi, dans le cas d'une facture notre système est capable à la fin de son exécution d'extraire, dans un format compréhensible par une machine, des informations telles que le montant de la facture, le numéro de la facture et la date d'émission de la facture, pour en faciliter le traitement (règlements, gestion comptable, etc.).

Organisation de la thèse

Ce mémoire est constitué de deux parties.

La première partie constitue un état de l'art des domaines scientifiques en lien avec la problématique de cette thèse et ses contributions. Elle est constituée de quatre chapitres.

Le premier chapitre présente tout d'abord les notions de base relatives au traitement automatique de documents. Ensuite, une revue de la littérature concernant les approches existantes pour le traitement automatique de documents est présentée. Le chapitre se poursuit par une discussion autour des difficultés liées à l'évaluation des systèmes de traitement automatique de documents. Une conclusion termine le chapitre.

Le deuxième chapitre détaille la tâche particulière d'extraction d'informations textuelles au sein d'images de documents. Tout d'abord, les étapes qui constituent cette tâche sont décrites, puis une revue des approches mentionnées dans la littérature pour cette tâche est présentée. Ensuite, une attention est portée sur l'extraction d'informations textuelles au sein d'images de factures en particulier. Le chapitre se poursuit avec la mise en lumière des problématiques d'évaluation des systèmes d'extraction d'informations textuelles. Enfin, le chapitre se termine par une conclusion.

Le troisième chapitre donne une vue d'ensemble des techniques de classification supervisée et non supervisée. Pour chaque catégorie de classification, les techniques les plus connues et ayant un intérêt particulier dans le cadre du mémoire sont présentées. Une conclusion termine ce chapitre.

Le quatrième chapitre aborde les notions relatives à l'analyse formelle de concepts. Les notions, particulièrement utiles dans ce mémoire, de concepts d'un contexte formel, de treillis de concepts et de règles d'association sont présentées. Le chapitre se termine par une conclusion.

La deuxième partie est dédiée à la présentation de nos contributions. Trois chapitres y sont consacrés.

Le premier chapitre présente notre approche d'extraction d'informations textuelles fondée sur la décomposition quadtree. Tout d'abord, notre approche de décomposition d'images de factures est présentée. Ensuite, quelques résultats d'expérimentation de notre approche sont présentés, avant de conclure le chapitre.

Le deuxième chapitre présente notre système d'extraction d'informations textuelles. Tout d'abord notre programme informatique de génération automatique d'images de factures synthétiques est présenté. Ensuite, les modèles sur lesquels est fondé notre système sont présentés : (i) détermination de *régions prototypes* à partir d'un jeu de données initial ; (ii) détermination de chemins pour naviguer efficacement au sein des *régions prototypes*, à partir du treillis de concepts d'un contexte formel convenablement construit ; (iii) mise à jour incrémental du système afin de l'enrichir avec des données supplémentaires.

Le troisième chapitre présente le processus d'extraction d'informations textuelles mis en œuvre par notre système. Une évaluation de notre système pour l'extraction d'informations textuelles au sein d'images de factures est également présentée dans ce chapitre.

Finalement, nous dressons un bilan de notre travail et dégageons quelques perspectives.

Première partie

État de l'art

Chapitre 1

Traitement automatique de documents

Sommaire

1.1 Structures de documents 28
1.1.1 Structure géométrique 29
1.1.2 Structure logique 30
1.2 Analyse de documents 30
1.2.1 Méthodes hiérarchiques 32
1.2.2 Méthodes non-hiérarchiques 36
1.3 Compréhension de documents 37
1.3.1 Approches fondées sur des grammaires 38
1.3.2 Approches à base de règles 39
1.3.3 Approches fondées sur des techniques d'apprentissage automatique 41
1.3.4 Autres approches 41
1.4 Évaluation des performances des systèmes de traitement automatique de documents 43
1.5 Conclusion 44

Les images de documents contiennent des informations utiles à leurs destinataires. En effet, ce sont des médias permettant de transférer des connaissances. Des documents tels que des rapports techniques, des journaux, des livres, des chèques de banque et des factures contiennent des connaissances spécifiques qu'un humain peut acquérir. Par exemple, sur un chèque de banque, la banque qui traite le chèque est intéressée par le montant renseigné en chiffre et en lettre, par l'émetteur du chèque, le destinataire du chèque, le compte bancaire concerné, etc.. Le montant, l'émetteur, le destinataire et le numéro de compte bancaire font partie des connaissances à acquérir. L'acquisition de ces connaissances par un système informatique constitue un défi important. En effet, les volumes de documents numériques en circulation au sein des entreprises et organisations sont de plus en plus importants. Leur traitement par des agents humains devient chronophage et présente de moins en moins de valeur ajoutée. Le traitement automatique d'images de documents (ou plus simplement, le traitement de documents) consiste en la transformation dans un format compréhensible par un système informatique, de données présentes au sein de documents et compréhensibles par l'Homme. Dans (Tang *et al.*,

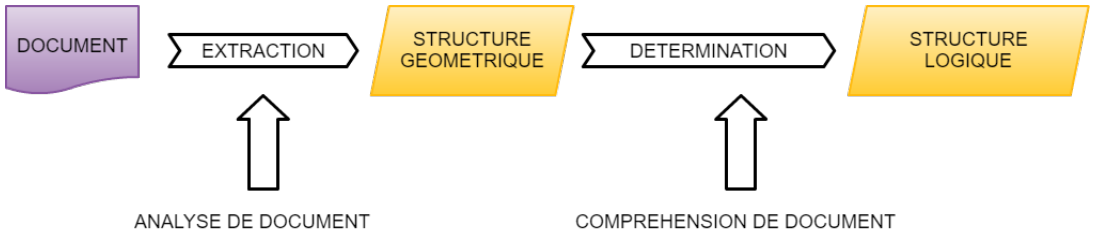


FIGURE 1.1 – Schéma du modèle de base de traitement de documents proposé par (Tang *et al.*, 1996).

1996), un modèle de base pour le traitement de documents est présenté. Ce modèle est composé des concepts suivants :

- un document possède deux structures : une structure géométrique (ou de mise en page) et une structure logique ;
- le traitement de documents est constitué de deux phases : l'analyse de documents et la compréhension de documents ;
- l'analyse de documents consiste à extraire la structure géométrique d'un document ;
- la compréhension de documents consiste à transformer la structure géométrique d'un document en sa structure logique ; une fois la structure logique établie, sa signification peut être décodée par une intelligence artificielle ou toutes autres techniques.

La Figure 1.1 met en évidence les relations entre la structure géométrique, la structure logique, l'analyse de documents et la compréhension de documents.

Le modèle de base pour le traitement de documents est formellement décrit dans ce qui suit.

Définition 1. *Un document Ω est défini par un quintuplet*

$$\Omega = (\mathcal{T}, \Phi, \delta, \alpha, \beta) \tag{1.1}$$

tel que

$$\mathcal{T} = \{\Theta^1, \Theta^2, ..., \Theta^i, ..., \Theta^m\}, \tag{1.2}$$

où

$$\Theta^i = \{\Theta^i_j\}^*$$

et

$$\begin{aligned} \Phi &= \{\varphi_l, \varphi_r\} \\ \delta &= \mathcal{T} \times \Phi \rightarrow 2^{\mathcal{T}} \\ \alpha &= \{\alpha^1, \alpha^2, ..., \alpha^p\} \subseteq \mathcal{T} \\ \beta &= \{\beta^1, \beta^2, ..., \beta^q\} \subseteq \mathcal{T}, \end{aligned} \tag{1.3}$$

où $\mathcal{T}$ est un ensemble fini d'objets du document. Ces objets sont des ensembles de blocs $\Theta^i (i = 1, 2, ..., m)$; $\{\Theta^i_j\}^*$ dénote des répétitions de subdivisions ; Φ est un ensemble fini de facteurs de liaisons ; φ_l et φ_r représentent respectivement les liaisons et les répétitions de liaisons ; δ est un ensemble fini de fonctions de liaisons logiques indiquant un lien logique entre les objets du document ; α est un ensemble fini d'objets racines ; et β est un ensemble fini d'objets finaux.

Définition 2. *Le traitement de documents consiste à construire le quintuplet représenté par les équations 1.1 à 1.3.*
L'analyse de documents consiste à extraire les éléments $\mathcal{T}$, Θ^i et Θ^i_j de l'équation 1.2, autrement dit extraire la structure géométrique de Ω.
La compréhension de documents consiste à déterminer Φ, δ, α et β de l'équation 1.3 en considérant la structure logique de Ω.

Pour un document spécifique, telle que la facture de la Figure 1.2(a), sa division en plusieurs blocs est illustrée dans la Figure 1.2(b) et la structure géométrique qui en découle peut-être représentée graphiquement par la Figure 1.2(c). Dans cette figure, la facture est décomposée en plusieurs objets dits *composites*) : les zones textuelles (Z_i), les zones graphiques (G_j) et les zones de tableaux (T_k). Ces objets *composites* sont eux mêmes composés de blocs de texte, blocs d'objets graphiques ou blocs d'objets de type tableau.

Dans ce qui suit, la Section 1.1 décrit les deux types de structures constituant un document : structure géométrique et structure logique. Les Sections 1.2 et 1.3 décrivent les deux phases du processus de traitement de documents que sont l'analyse et la compréhension de documents. Différentes approches de la littérature consacrées à ces deux phases sont également évoquées.

1.1 Structures de documents

Selon le standard international ISO/CEI 8613-1 :1994 [1] concernant les technologies de l'information, une structure de documents est la division et la subdivision successive du contenu d'un document au sein de zones de plus en plus petites nommées *objets*. Un objet qui ne peut être subdivisé en sous-objets est appelé un *objet de base*. Un mot est exemple d'objet de base. Tous les autres objets sont appelés *objets composites*. Un paragraphe constitué de plusieurs mots est un exemple d'objet composite. Une structure de documents est la combinaison d'une structure géométrique et d'une structure logique. D'une part, la structure géométrique concerne les caractéristiques géométriques (de mise en page) d'un document. D'autre part, la structure logique concerne les propriétés sémantiques de ce document.
En traitement de documents le découpage d'images de documents en régions ou blocs homogènes (texte, graphiques, tableaux, etc) puis en lignes de texte et en caractères à l'intérieur de ceux-ci est désignée par segmentation. Ce processus permet d'extraire la structure géométrique du document qui peut être représentée de diverses manières, indépendamment de, ou conjointement à, la structure logique du document. Des paramètres de présentation du document ont été utilisés dans (Kreich *et al.*, 1991), (Niyogi & Srihari, 1995) et (Ishitani, 1999). Ces paramètres correspondent aux dimensions et aux distances entre les objets contenus dans le document tels que les caractères, les mots, les lignes ou les régions. Bien que cette méthode de représentation fournisse une information utile, elle ne reflète pas complètement les relations spatiales entre les composantes géométriques du document. La structure géométrique du document peut être mieux représentée par un arbre hiérarchique dérivé d'un ensemble de règles (Tsujimoto & Asada, 1990) et (Yamashita *et al.*, 1991). Une telle représentation décrit les relations spatiales

1. https ://www.iso.org/obp/ui/iso :std :iso-iec :8613 :-1 :ed-1 :v1 :fr

entre les composantes géométriques du document de manière hiérarchique. L'inconvénient des représentations à base de règles réside dans le fait que les règles peuvent devenir arbitraires. D'autre représentations basées sur des grammaires formelles offrent l'avantage de limiter les types de règles de production qui peuvent être utilisés. Dans ces représentations, un document est considéré comme une séquence ou une chaîne de caractéristiques des composantes géométriques.

1.1.1 Structure géométrique

La structure géométrique d'un document, également appelée structure de mise en page ou structure physique, représente les objets de celui-ci. Selon le standard international ISO/CEI 8613-1 :1994, la structure géométrique d'un document est définie de la manière suivante.

Définition 3. *La structure de mise en page est le résultat de la division et de la subdivision du contenu d'un document en parties de plus en plus petites sur la base de la présentation, par exemple en pages, en blocs. Elle est constituée de tous les objets de mise en page et les portions de contenu associées formant la hiérarchie de mise en page d'un document.*

S'ensuivent les définitions ci-dessous :

Définition 4. *Un bloc est un composant de mise en page de base qui correspond à une zone rectangulaire dans un cadre ou dans une page.*

Définition 5. *Une page est une composante de mise en page qui correspond à une zone rectangulaire utilisée pour la présentation du contenu du document.*

Définition 6. *Un cadre est un type de composant de mise en page composite qui correspond à une zone rectangulaire à l'intérieur d'une page ou d'un autre cadre.*

La définition suivante précise de manière formelle la structure géométrique d'un document selon le modèle de base donné par les équations 1.1 et 1.2.

Définition 7. *La structure géométrique est décrite par l'élément $\mathcal{T}$ dans l'espace du document $\Omega = (\mathcal{T}, \Phi, \delta, \alpha, \beta)$ des équations 1.1 et 1.2 et β_U un ensemble d'opérations effectuées sur $\mathcal{T}$ telles que :*

$$\begin{aligned} \mathcal{T} &= \{\mathcal{T}_B, \mathcal{T}_C\}, \\ \beta_U &= \{\cup, \cap\}, \\ \forall_{i \neq j}((\mathcal{T}_i \cup \mathcal{T}_j) &\subseteq \Omega), \\ \forall_{i \neq j}((\mathcal{T}_i \cap \mathcal{T}_j) &= \varphi), \end{aligned} \tag{1.4}$$

où $\mathcal{T}_B$ représente un ensemble d'objets de base, et $\mathcal{T}_C$ un ensemble d'objets composites.

$$\begin{aligned} \mathcal{T}_C &= \{\Theta^1, \Theta^2, ..., \Theta^m\}, \\ \mathcal{T}_B &= \{\Theta^i_j | \Theta^i \in \mathcal{T}_C\}. \end{aligned} \tag{1.5}$$

Pour un document spécifique, telle que la facture de la Figure 1.2(a), sa division en plusieurs blocs est illustrée dans la Figure 1.2(b) et la structure géométrique qui en découle peut-être représentée graphiquement par la Figure 1.2(c). Dans cette figure, la facture est décomposée en plusieurs objets composites : les zones textuelles (Z_i), les zones graphiques (G_j) et les zones de tableaux (T_k). Ces objets composites sont eux mêmes composés de blocs de texte, blocs d'objets graphiques ou blocs d'objets de type tableau.

1.1.2 Structure logique

La compréhension de documents consiste à trouver des relations logiques entre les objets d'un document. Pour faciliter ce processus, une structure logique est nécessaire. Toujours selon le standard international ISO/CEI 8613-1 :1994, une structure logique est le résultat de la division et de la subdivision du contenu d'un document en parties de plus en plus petites, sur la base de la signification perceptible par l'homme du contenu, par exemple des chapitres, des sections, des alinéas.

Définition 8. *La structure logique est décrite par les éléments Φ, δ, α et β dans l'espace de documents $\Omega = (\mathcal{T}, \Phi, \delta, \alpha, \beta)$ des équations 1.1 et 1.2, telle que :*

$$\begin{bmatrix} \Phi \\ \alpha \\ \beta \\ \delta \end{bmatrix} = \begin{bmatrix} \{\varphi_l, \varphi_r\} \\ \{\alpha^1, \alpha^2, ..., \alpha^p\} \\ \{\beta^1, \beta^2, ..., \beta^q\} \\ \mathcal{T} \times \Phi \rightarrow 2^{\mathcal{T}} \end{bmatrix} \tag{1.6}$$

Pour un document spécifique telle que la facture de la Figure 1.2(a), sa structure logique peut être représentée graphiquement par la Figure 1.2(d).

1.2 Analyse de documents

L'analyse d'une image de documents I constituée de mots, de lignes et d'objets graphiques tels que des logos, consiste à extraire et isoler les mots, les lignes et les objets de I, puis à les regrouper en blocs. Les différents blocs ainsi formés constituent la structure géométrique du document. La Figure 1.2(b) montre un exemple de découpage en blocs d'une image de facture, définissant ainsi sa structure géométrique. Un bloc est représenté par un rectangle contenant du texte ou des objets graphiques. Un bloc peut lui-même être constitué de sous-blocs, eux-même constitués de sous-sous-blocs et ainsi de suite. Cette structure peut être représentée par un arbre comme montré dans la Figure 1.2(c). Plusieurs catégories de méthodes permettant de construire un tel arbre existent :

- les méthodes hiérarchiques, qui considèrent les relations géométriques entre les blocs d'une page de documents découpée en blocs ;
- les méthodes non-hiérarchiques, qui ne tiennent pas compte des relations géométriques entre les blocs d'une page de documents découpée en blocs.

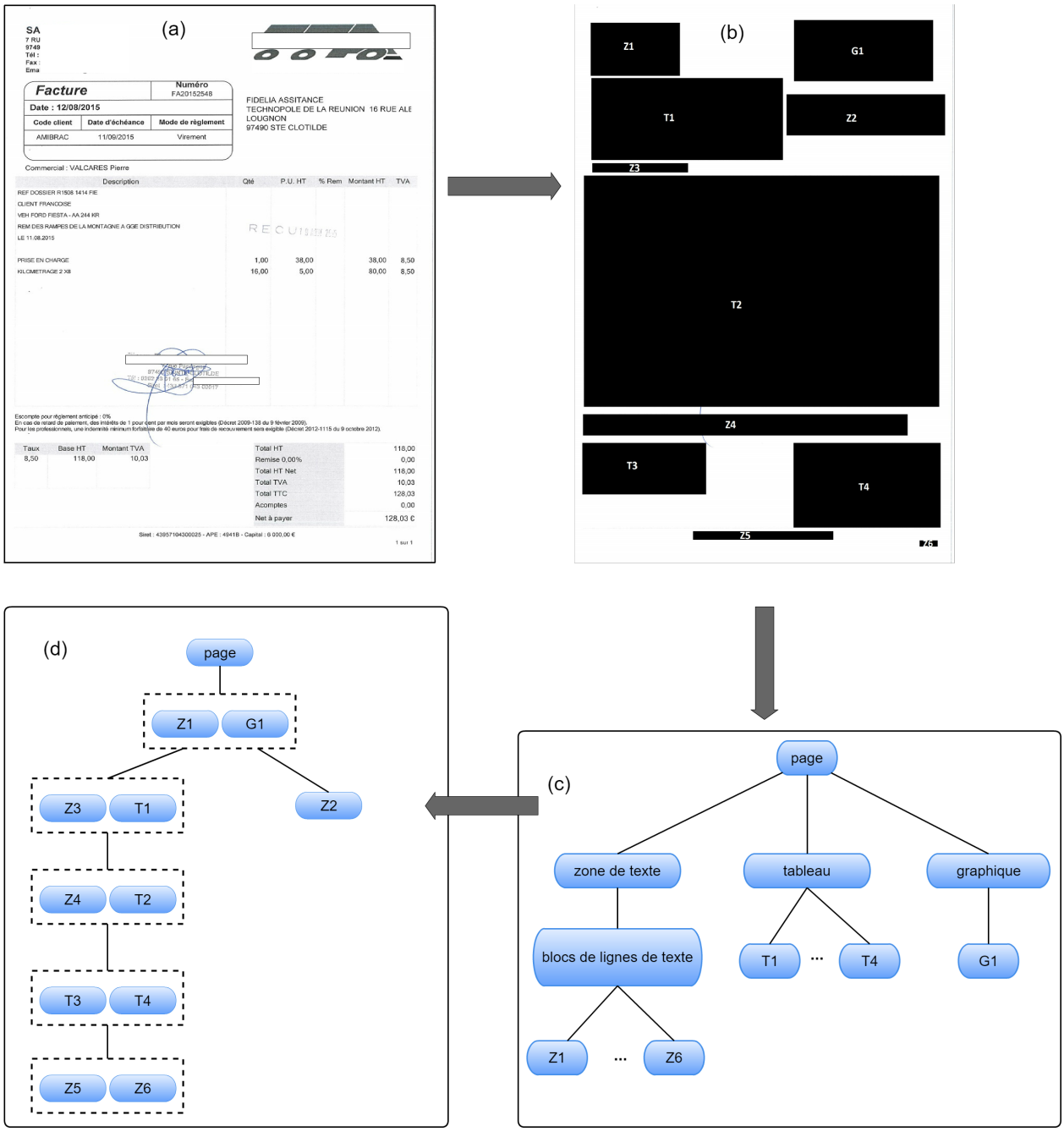


FIGURE 1.2 – Structure géométrique et structure logique d'une image de facture.

1.2.1 Méthodes hiérarchiques

Les méthodes hiérarchiques sont divisées en deux types d'approches :
— approche descendante : des nœuds parents vers les nœuds fils ;
— approche ascendante : des nœuds fils vers les nœuds parents ;

Ces deux types d'approches utilisées en analyse de documents ont chacune des avantages et des inconvénients. D'une part, l'approche descendante est rapide et efficace pour le traitement des documents ayant un format spécifique. D'autre part, l'approche ascendante nécessite des temps de calcul importants. Néanmoins, elle est applicable a une grande variété de documents.

Approches descendante

L'approche descendante consiste à subdiviser successivement un document en régions et en sous-régions. La subdivision commence à partir de l'image du document entier et se termine lorsque un certain critère est vérifié. Les régions finales obtenues constituent la segmentation finale du documents.
L'algorithme de subdivision basé sur une structure d'arbre X-Y pour l'analyse de journaux techniques de (Nagy *et al.*, 1992), l'algorithme de segmentation d'image par recouvrement de (Baird *et al.*, 1990) sont des algorithmes descendant typiques.
La structure d'arbre X-Y de (Nagy *et al.*, 1992) est une structure de représentation spatiale dans laquelle chaque nœud correspond à un bloc rectangulaire. Les fils de chaque nœud représentent les localisations des subdivisions du bloc parent dans une direction (horizontale ou verticale) particulière.
(Baird *et al.*, 1990) proposent une méthode de segmentation d'images fondée sur un algorithme de recouvrement d'espaces blancs. L'algorithme consiste à recouvrir la page d'un document par des rectangles maximaux, c'est à dire des rectangles qui ne peuvent pas être étendus car rencontrant des obstacles ou les bords de la page.
Krishnamoorthy et al. (Krishnamoorthy *et al.*, 1993) décrivent un algorithme de découpage hiérarchique de page de documents qui construit un arbre dans lequel chaque nœud représente un bloc rectangulaire. Un utilisateur peut spécifier des grammaires pour chaque bloc rectangulaire.
Fujisawa (Fujisawa & Nakano, 1992) introduit un langage de définition de formulaire permettant de représenter les structures géométriques de documents en un ensemble de régions rectangulaires imbriquées. Cette méthode permet de définir des modèles de documents génériques auxquels sont comparés des documents candidats afin d'y extraire des éléments spécifiques.

Aiello et al. (Aiello *et al.*, 2003) présentent un système de traitement de documents capable d'assigner des étiquettes logiques et d'extraire l'ordre de lecture au sein d'un vaste ensemble de documents. Toutes les sources d'informations telles que les caractéristiques géométriques, les relations spatiales, ou encore les éléments textuels sont utilisées au cours du traitement. Afin de traiter efficacement ces sources d'informations, les auteurs définissent un modèle de représentation général et suffisamment flexible pour représenter des documents complexes. Le système proposé intègre des modules fondés sur l'intelligence artificielle, les techniques de traitement du langage naturel et la vision

par ordinateur.
Les auteurs distinguent deux classes de connaissances relatives aux documents : des connaissances génériques et des connaissances spécifiques. Les connaissances spécifiques aux documents sont les connaissances qui sont spécifiques à une classe restreinte de documents. Les connaissances génériques sont les connaissances de documents communes à une vaste variété de documents. Un exemple de connaissance spécifique est l'utilisation de la police d'écriture Times Roman et la taille de police de 10 points dans les textes de ACM Transaction [2]. Un exemple de connaissance générique est la taille de police plus grande généralement utilisée pour le titre par rapport à la taille de police utilisée pour les paragraphes.
Le système proposé est capable d'extraire la structure géométrique et la structure logique d'une image de documents. Les détails concernant l'établissement de la structure logique par le système sont abordés en Section 1.3. Pour établir la structure géométrique d'un document, les auteurs distinguent deux grandes classes de relations géométriques :

— l'arrangement global des objets de la page ;
— les relations spatiales entre les objets de la page.

Pour la représentation des relations spatiales entre objets représentés par des régions rectangulaires les en-capsulant, les auteurs utilisent des relations qualitatives. Sur les axes X et Y (en considérant le plan en deux dimensions défini par une page de documents) les relations suivantes sont considérées : *précède, rencontre, chevauche, commence, finit, équivaut, dans* et leurs inverses. Pour l'arrangement global des objets d'une page de documents, une relation de voisinage est utilisée. Deux objets o_1 et o_2 sont *voisins* si ils partagent une arête sur le diagramme de Voronoi (Aurenhammer, 1991). Le diagramme de Voronoi est calculé sur les centres de gravité des régions rectangulaires représentant les objets du document. Cette relation est stockée sur un graphe pondéré où les nœuds sont les objets du document. Une arête représente l'existence de la relation de voisinage entre deux nœuds. Le poids d'une arête est la distance Euclidienne entre les objets de ces deux nœuds. Ainsi, la représentation géométrique proposée capture les informations essentielles d'un document donné, lesquelles sont requises par le système pour d'autres analyses logiques (voir la Section 1.3).

Approches ascendante

L'approche ascendante consiste à rassembler des composants géométriques de base en plusieurs groupes selon leurs caractéristiques, puis ces groupes sont combinés en groupes plus grands et ainsi de suite. Dans la littérature, les premières méthodes d'analyse de documents mettant en œuvre une approche ascendante apparaissent à partir des années 80 avec l'algorithme RLSA introduit par (Wong *et al.*, 1982; Wahl *et al.*, 1982), le système MACSYM de (Inagaki *et al.*, 1984), l'algorithme de (Doster, 1984) et l'algorithme de (Masuda *et al.*, 1985). S'ensuivent l'algorithme de (Iwaki *et al.*, 1987), l'algorithme de séparation de chaines textuelles de (Fletcher & Kasturi, 1988), l'algorithme de (Ciardiello *et al.*, 1988), le Docstrum de (O'Gorman, 1993) et l'algorithme de (Akiyama & Hagita, 1990).

Plus récemment, (Gao *et al.*, 2013), (Hamza *et al.*, 2007), (Cesarini *et al.*, 2003)

2. http ://dl.acm.org/pubs.cfm

s'appuient également sur une approche ascendante afin d'extraire la structure géométrique d'un document.
Gao et al. (Gao *et al.*, 2013) proposent un algorithme nommé DTMSER (Distance Transform based Maximally stable extremal regions) pour la détection de régions d'intérêt au sein d'images de documents. La transformée de distance utilisée associe à chaque pixel d'une image de documents la distance minimum aux autres pixels du document. Dans le domaine de l'analyse de documents, il est intéressant de pouvoir identifier les régions d'intérêt relatives aux éléments de la structure d'un document. Les caractères, les mots, les lignes et les paragraphes sont des éléments structurant dans un document car ils contiennent des informations sémantiques importantes. De plus, l'identification doit se faire de manière efficace, répétable et stable. La notion d'échelle dans le cas des documents est étroitement liée à la distance entre les éléments de la structure d'un document. En effet, les caractères sont placés plus proches les uns des autres que le sont les mots entre eux. Ces derniers sont eux-même placés plus proche les uns des autres que le sont les paragraphes. L'algorithme MSER (Maximally stable extremal regions) (Matas *et al.*, 2004) fournit un cadre d'analyse multi-niveaux efficace, basé sur la stabilité d'une propriété d'un pixel donné, typiquement sa luminosité. L'idée clé du détecteur proposé par les auteurs est d'améliorer l'efficacité de l'algorithme MSER afin d'identifier les régions stables où la stabilité est définie par une fonction de distance d'une région aux régions avoisinantes. Ainsi, les régions les plus distantes des régions voisines seraient plus stable que les régions proches les unes des autres. En appliquant l'algorithme MSER à la transformée de distance un dendrogramme des régions maximales est produit (Figure 1.3) [3].

Hamza et al. (Hamza *et al.*, 2007) proposent une méthode nommée CBRDIA (Case-base Reasoning For Document Invoice Analysis) utilisant les principes de raisonnement à partir de cas (RàPC) (Aamodt & Plaza, 1994) pour analyser, reconnaitre et interpréter des factures. Le raisonnement à partir de cas est une stratégie de résolution qui utilise des expériences passées pour traiter de nouveaux problèmes jamais traité auparavant. Dans CBRDIA, deux sortes de cas sont définis : document et structure. Pour un document donné, l'approche est fondée sur trois étapes principales : élaboration du problème, résolution globale du problème et une résolution locale du problème.
L'élaboration du problème consiste à extraire des indices à partir du document. Ces indices sont soit des mots clés et leurs relations spatiales, soit des lignes de tableaux. Ce problème est ensuite résolu en utilisant soit un processus de résolution globale, soit un processus de résolution locale.
La résolution globale consiste à vérifier si un cas de type document similaire existe dans la base de cas. Si oui, le système résout le problème en appliquant la solution de la base de données pour ce problème. Sinon, le problème est décomposé en sous-problèmes et résolu en résolvant ses sous-problèmes. Cette étape de résolution d'un problème par la résolution de ses sous-problèmes correspond au processus de résolution locale.
La méthodologie est similaire concernant l'extraction de la structure géométrique d'un document. Une structure de documents est le rassemblement de tous les éléments de la structure. Un graphe est utilisé pour représenter la structure géométrique d'un document. Dans le but d'avoir une représentation graphique harmonieuse, utile pour de futures comparaisons, les auteurs considèrent tous les sommets visibles à un même niveau. Cela signifie que la différence entre des sommets est caractérisée par des arêtes qui

3. Dendrogramme extrait de (Gao *et al.*, 2013)

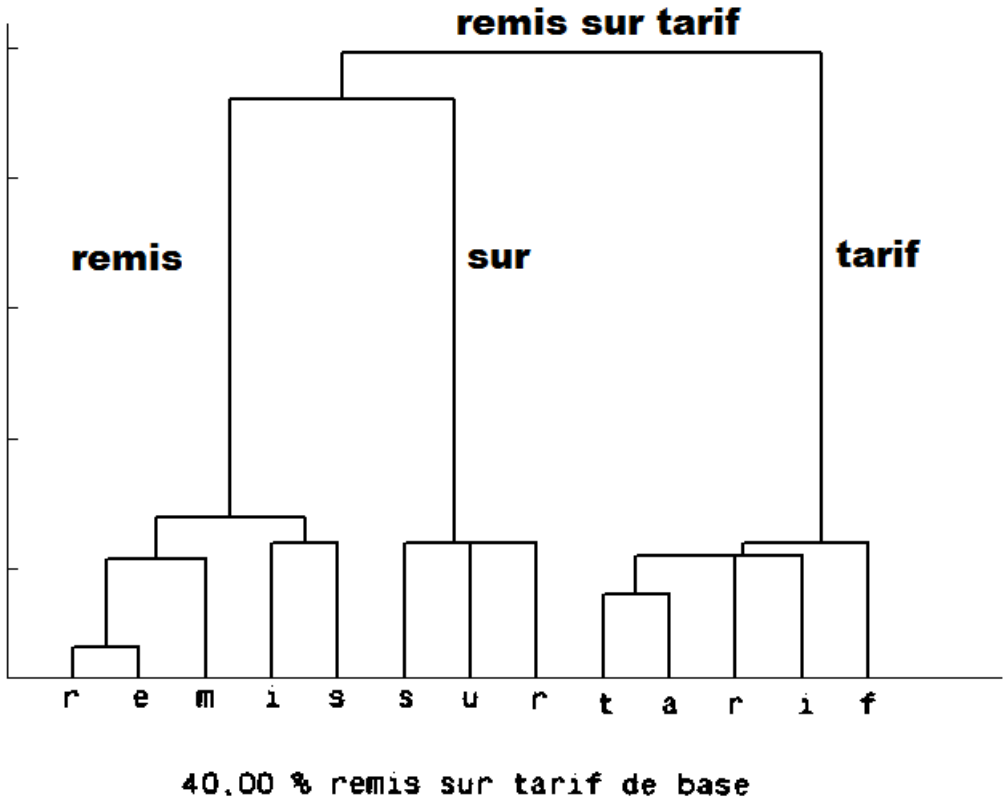


FIGURE 1.3 – Dendrogramme produit par l'algorithme DTMSER. Les nœuds feuilles correspondent aux composants connectés de l'image, alors que les regroupements de nœuds dépendent uniquement de la distance entre régions, ce qui donne lieu à des groupes sémantiquement pertinents.

sont soit "spatial" (gauche, droite, haut, bas) lorsqu'ils désignent des relations spatiales, soit "contenu" lorsqu'ils désignent un composant de la structure. Finalement, le processus de résolution défini précédemment s'applique de la même manière aux cas de type structure : vérifier l'existence d'une structure similaire dans la base de cas ; si oui, appliquer la solution correspondante (résolution globale) ; si non, décomposer le problème en sous-problèmes et résoudre chaque sous problème (résolution locale).

Cesarini et al. (Cesarini *et al.*, 2003) proposent un système utilisant une procédure ascendante consistant à regrouper des pixels en objets physiques. Le système est composé des étapes suivantes :

1. Extraire les lignes horizontales et verticales (si elles existent) à partir d'une image de documents en noir et blanc.
2. Extraire les composants connectés à partir de l'image dépourvue de ligne.
3. Calculer la longueur et la hauteur moyenne des composants connectés. Les composants connectés sont dès lors considérés comme étant de type "caractère".
4. Ordonner ces composants au sein d'une liste par ordre croissant des valeurs de barycentre de leurs abscisses.
5. Regrouper ces composants (caractères) en rectangles de type "mot". Ces rectangles peuvent contenir des mots ou des phrases alignés horizontalement en une seule ligne de texte. Pour cela, le premier composant de type caractère de la liste ordonnée de composants de ce type est affecté à un nouveau rectangle de type mot ; ensuite, le composant de type caractère suivant est affecté à ce rectangle sous certaines conditions. Cette étape de regroupement est répétée à chaque composant de la liste ordonnée non encore affecté à un rectangle de type mot jusqu'à ce que tous les composants soient affectés à un rectangle de type mot.
6. Déterminer quels rectangles de type mot contiennent du texte et lesquels contiennent des objets graphiques en utilisant un module OCR. Si le contenu lu par le module OCR au sein des rectangles de type mot contient un seuil minimum fixé de caractères alors le rectangle est un bloc de texte et le texte contenu est extrait ; sinon le rectangle est considéré comme un objet graphique.
7. Classer les blocs de texte en texte numérique, alphabétique ou alpha-numérique.

Finalement, la structure géométrique obtenue correspond à une collection d'objets physiques tels que :

1. des lignes verticales et horizontales ;
2. des objets graphiques ;
3. des blocs de texte (numérique, alphabétique ou alpha-numérique).

1.2.2 Méthodes non-hiérarchiques

Traditionnellement deux types d'approches non-hiérarchiques sont utilisées en analyse de documents : approche descendante, approche ascendante. Ces deux types d'approches présentent des faiblesses. Selon (Tang *et al.*, 1996) :

— elles ne sont pas efficaces pour le traitement de documents présentant des complexités géométriques importantes, en particulier, l'approche descendante ne peut

FIGURE 1.4 – Exemples d'images de documents ayant une structure géométrique complexe.

traiter que des documents simples ayant un format spécifique ou contenant un ensemble d'informations connues à l'avance ; ce type d'approche échoue lors du traitement de documents dont la structure géométrique est complexe comme montré dans la Figure 1.4 ;

— pour extraire la structure géométrique d'un document, l'approche descendante réalise des opérations itératives afin de découper le document en plusieurs blocs tandis que l'approche ascendante agrège de petits composants en des composants plus grands de manière itérative ; c'est pourquoi les deux approches sont coûteuses en temps de calcul.

Tang et al. (Tang *et al.*, 1995) présentent une approche fondée sur la signature de fractale modifiée (*modified fractal signature*) pour l'analyse de documents. Cette approche ne nécessite ni découpage, ni agrégation itérative et peut séparer un document en blocs en une seule étape. L'approche peut être utilisée pour traiter des types variés de documents y compris ceux dont la structure présente une grande complexité géométrique. Les détails de l'approche étant en dehors du cadre de ce mémoire, les lecteurs peuvent se référer à Tang et al. (Tang *et al.*, 1995) pour plus d'informations.

1.3 Compréhension de documents

Tandis que l'analyse de documents extrait la structure géométrique d'une image de documents, la compréhension de documents fait correspondre à cette structure géométrique une structure logique en considérant des liaisons logiques entre les objets du document. La compréhension de documents se réfère au domaine concerné par l'analyse logique et sémantique des documents afin d'en extraire des informations compréhensibles par l'humain dans un format compréhensible par une machine. Pour cela, les systèmes de compréhension de documents fournissent des technologies permettant de transformer des informations utiles à partir d'une image en une représentation formelle.
Dengel (Dengel, 2003)discute des défis de la compréhension de documents. L'auteur souligne l'importance de l'utilisation des connaissances pour extraire toutes les informations pertinentes au sein d'une image de documents. Pour la tâche de compréhension de documents il se réfère aux diverses sources intuitivement compréhensibles qui sont utilisées pour capturer les principaux morceaux d'informations qui conduisent à des décisions

ou des processus. Une structure géométrique peut être une connaissance, en particulier si elle est significative d'une certaine classe de documents à traiter. Par exemple, en considérant les rectangles (blocs) de l'image de la Figure 1.2(b), selon Dengel nous pouvons facilement décrire la mise en page typique d'une facture. De plus, il est facile d'émettre des hypothèses sur la position spatiale de certains objets logiques tels que le destinataire ou l'émetteur. L'auteur déclare également qu'il est possible de définir de façon plus restrictive les différents concepts pertinents à un rôle et à une tâche dans une entreprise, par exemple «notification de réclamation» et «rapport d'évaluation» dans une opération d'assurance ou «facture» et «bon de livraison» dans un service d'achat. En outre, toutes ces sources ne sont valables que si les caractéristiques et propriétés de la personne assurée, des clients, des fournisseurs et des produits sont connues, ces caractéristiques et propriétés étant obtenues à partir des bases de données de l'entreprise ou d'autres dispositifs. Lorsqu'un humain interprète la sémantique présente derrière un concept, il fait usage de ce type de sources de connaissances acquises par l'expérience ou par l'utilisation d'une expertise particulière. Ainsi, l'un des grands défis de la compréhension de documents est de combiner ces sources de connaissances de sorte que les concepts (ou messages) pertinents capturés au sein des documents puissent être extraits automatiquement par un système informatique. Un autre défi principal est de prendre en considération la diversité des documents à traiter. En effet, il y a :

- les documents très structurés qui ont une mise en page statique. Ils peuvent être décrits par des modèles reprenant leurs structures géométrique et faisant apparaitre les localisations des régions d'intérêt où se trouvent les informations pertinentes à extraire ;
- les documents semi-structurés qui ont quelques caractéristiques géométriques régulières permettant d'identifier la classe de documents ; par un exemple, un document administratif avec une en-tête et un pied de page pré-définis et un corps vide destiné à accueillir du texte libre ;
- les documents non structurés qui ne peuvent pas être caractérisé par un modèle ; ils contiennent des objets logiques disposés dans une mise en page irrégulière.

Dans cette section nous parcourons un panorama des méthodes de la littérature établi par (Mao *et al.*, 2003) et (Trupin, 2005). Dans ce panorama, des méthodes telles que (Ingold & Armangil, 1991), (Krishnamoorthy *et al.*, 1993), (Conway, 1993) et (Tateisi & Itoh, 1994) sont fondées sur des grammaires. D'autres méthodes telles que (Kreich *et al.*, 1991) et (Niyogi & Srihari, 1995) s'appuient sur des bases de connaissances, tandis que les approches telles que (Dengel & Dubiel, 1996), (Ishitani, 1999) et (Malerba *et al.*, 2001) utilisent des techniques d'apprentissage automatique afin d'extraire la structure logique d'un document.

1.3.1 Approches fondées sur des grammaires

Ingold et Armangil (Ingold & Armangil, 1991) proposent une méthode de reconnaissance de la structure logique fondée sur une description formelle de chaque classe de documents à l'aide de règles de composition et de représentation. Les règles de composition définissent la structure logique générique et les règles de représentation définissent les caractéristiques physiques des entités logiques à reconnaitre. Les règles de compositions sont décrites de manière formelle par des grammaires EBNF (Extended

Backus-Naur Form) (Scowen, 1998). La description d'un document permet d'établir un graphe d'analyse dont les arcs sont étiquetés avec les classes des entités à reconnaitre. La structure logique d'un document est ainsi établie en recherchant un chemin dans le graphe d'analyse sous la contrainte que les attributs typographiques d'une entité de ce chemin doivent correspondre à ceux de l'objet correspondant du document.

Krishnamoorthy et al. (Krishnamoorthy *et al.*, 1993) proposent une méthode de reconnaissance de la structure logique d'un document en appliquant récursivement des grammaires aux profils de projection horizontaux et verticaux de la page de documents. Il y a quatre étapes dans le processus d'analyse :

1. établir l'histogramme de projection afin d'extraire des "atomes" ;
2. regrouper les atomes en "molécules" ;
3. assigner des étiquettes logiques aux "molécules" ;
4. fusionner les entités de même type.

Les résultats de la segmentation et du processus d'étiquetage logique sont stockés dans un arbre X-Y étiqueté. Cette approche ne fait pas de distinction entre la structure physique et la structure logique.

Conway (Conway, 1993) utilise des grammaires de page et des techniques d'analyse grammaticale afin de reconnaitre la structure logique de documents à partir de la structure géométrique. La structure géométrique est décrite par un ensemble de règles de grammaires. Chacune des règles est une séquence de composants reliés par une relation de voisinage. Les relations de voisinage identifiées sont du type : "sous", "à gauche de", "sur", "sur la gauche" et "proche de". Une grammaire particulière et indépendante du contexte est utilisée pour décrire la structure logique de documents. Les deux grammaire sont déterministes.

Tateisi et Itoh (Tateisi & Itoh, 1994) considèrent l'extraction de la structure logique d'un document comme un problème d'analyse syntaxique et stochastique. Le document est modélisé comme une chaine de lignes de texte et d'objets graphiques. Une étape de pré-traitement permet de segmenter et de classer les lignes de texte et les objets graphiques, et la chaîne est analysée à l'aide d'une grammaire stochastique régulière avec des attributs. Les caractères contenus dans les lignes sont reconnus et la taille de leur fonte est déterminée. Chaque règle grammaticale est associée à un coût. L'analyse retient les résultats d'analyse possibles en fonction de leur coût total. L'algorithme a été testé sur 70 pages de texte Japonais extraits de livres et de magazines. Les auteurs rapportent une moyenne de 89% de reconnaissance correcte sur des pages de journaux techniques.

1.3.2 Approches à base de règles

Kreich et al. (Kreich *et al.*, 1991) décrivent un environnement expérimental nommé SODA (System for Office Document Analysis). Leur approche ascendante consiste à regrouper des composants connexes au sein de blocs de texte, puis à déterminer les lignes

au sein de chaque bloc et les mots au sein de chaque ligne. Une reconnaissance de texte et d'objets graphiques est réalisée. La connaissance du domaine et celle de la structure géométrique et de la structure logique sont stockées dans une base de connaissances. Les objets de documents sont mis en correspondance avec les informations de la base de connaissances relatives à la structure géométrique et la structure logique. Une distance de Hamming (Hamming, 1950) est utilisée dans le processus de mise en correspondance afin de calculer une mesure de confiance. Une correspondance est trouvée si la mesure de confiance calculée est supérieure à un seuil fixé.

Yamashita et al. (Yamashita *et al.*, 1991) proposent une méthode fondée sur la construction d'un modèle pour l'analyse de la structure logique. Le modèle est un arbre structuré qui définit l'information concernant l'arrangement géométrique des objets du document. Les objets physiques sont des chaînes de caractères, des lignes et des images. Des séparateurs horizontaux et verticaux (longues zones blanches et lignes noires) sont détectés en considérant les objets extraits et des étiquettes sont assignées aux chaînes de caractères grâce à une méthode de relaxation. Les différentes classes d'étiquette sont : en-tête, titre, auteur, affiliation, résumé, corps de texte, numéro de page, colonne, pied de page, bloc et figure. Cette technique a été appliquée à 77 pages de garde de brevets japonais et 59 structures logiques ont été correctement déterminées.

Niyogi et Srihari (Niyogi & Srihari, 1995) présentent le système DeLoS pour la déduction de la structure logique de documents. Dans ce système, un modèle fondé sur une structure de règles de contrôle et sur un schéma de représentation de connaissances hiérarchique multi-niveaux est développé. Dans ce schéma, la connaissance à propos de la structure géométrique et de la structure logique de documents de types variés est encodée au sein d'une base de connaissances. Le système inclut trois niveaux de règles : les règles de connaissance, les règles de contrôle et les règles de stratégie. Les règles de contrôle vérifie l'application des règles de connaissances. Les règles de stratégie détermine l'usage des règles de contrôle. Une image de documents est d'abord segmentée en utilisant un algorithme ascendant. Ensuite, les blocs segmentés sont classés. Enfin, les blocs classés sont traités en entrée du système DeLoS et une structure logique est déduite en sortie du système.

Summers (Summers, 1995) décrit un algorithme de dérivation automatique de structure logique de documents à partir de structure géométrique générique. L'algorithme combine une segmentation de texte en zones et une classification de ces zones en composants logiques. La structure logique de documents est obtenu par le calcul d'une mesure de distance entre un segment géométrique et des prototypes pré-définis. Pour chaque étiquette logique, un ensemble de prototypes est défini. L'algorithme prend en entrée les zones de texte segmentées. Chaque segment de texte est étiqueté comme correct, sur-généralisé, ou incorrect. L'auteur rapporte une efficacité de 85% sur 196 pages de rapports techniques dans le domaine de l'informatique.

1.3.3 Approches fondées sur des techniques d'apprentissage automatique

Dengel et Dubiel (Dengel & Dubiel, 1996) décrivent le système DAVOS capable à la fois d'apprendre et d'extraire une structure logique de documents. DAVOS est un système de formation de concept qui apprend des concepts de documents en détectant des valeurs d'attribut distinctes sur des objets de documents. Un arbre géométrique est utilisé pour représenter le langage de concept. La construction de l'arbre géométrique est fondée sur un apprentissage à partir d'arbres de décision. Le système a été testé sur un ensemble de 40 lettres inconnues à classer.

Ishitani (Ishitani, 1999) propose un système fondé sur le concept de *calcul émergent* (Forrest *et al.*, 1991). Le système comprend cinq modules interactifs : un module d'analyse typographique, un module de reconnaissance d'objet, un module de segmentation d'objets, un module de regroupement d'objets et un module de modification d'objets. Tout d'abord, l'image de documents est segmentée en lignes de texte qui sont ensuite classées en différents types selon des règles spécifiques. Puis les lignes de texte classées sont regroupées et classées en composants logiques en utilisant des heuristiques. Les objets de documents mal segmentés peuvent être modifiés via le module de modification. Les objets modifiés sont transmis aux autres modules et de nouveaux objets sont créés par interaction entre les modules. Le système a été testé sur 150 documents extraits de sources variées. L'auteur rapporte un taux moyen de 96.3% d'extraction correcte d'objets logiques.

Malerba et al. (Malerba *et al.*, 2001) proposent une méthode fondée sur l'application de techniques d'apprentissage de règles pour l'acquisition automatique de modèles. Au sein d'un modèle les composants logiques sont en relation de dépendance les uns avec les autres. Ces dépendances peuvent être reflétées par des relations logiques entre les composants de mise en page (provenant d'une structure géométrique) associés aux composants logiques considérés. Par exemple, des composants logiques "titre" et "auteur" d'un document papier sont généralement liés de la manière suivante : l'auteur suit le titre. Dans le cas des articles publiés dans le journal IEEE *Transactions on Pattern Analysis and Machine Intelligence*, une telle dépendance est matérialisée par la relation logique suivante : le composant de mise en page nommé "titre" est au-dessus du composant de mise en page nommé "auteur". Ainsi, les auteurs mettent en œuvre une méthode d'apprentissage de règles dans le but de capturer une sorte de modèle typographique en générant des clauses logiques du type :
auteur(X) ← au-dessus de(Y,X), titre (Y).
Les modèles ainsi obtenus reflètent les dépendances qui peuvent exister entre composants logiques.

1.3.4 Autres approches

Dans (Anjewierden, 2001) les auteurs présentent le projet AIDAS ayant pour objectif d'extraire la structure logique d'un fichier PDF et d'assigner des index à chaque élément de la structure logique. Les fichiers PDF sont des manuels techniques. Les index

peuvent soit être le contenu de l'élément (ex : "Cette section traite de la partie arrière d'une voiture"), soit indiquer comment l'élément peut être utilisé comme instruction (ex : "Cette section fournit un aperçu des composants d'une voiture"). L'ensemble des index obtenus est stocké en base de données. Un instructeur peut alors retrouver une information technique en interrogeant la base de données d'index. Le système AIDAS comprend les étapes suivantes :

1. Interpréter le document PDF : le format PDF est un format puissant en terme de possibilités de rendu et largement utilisé pour l'échange de documents. L'interprétation du document PDF consiste à extraire la structure géométrique de celui-ci.
2. Découvrir la structure logique du document : les segments que AIDAS doit stocker correspondent à la structure logique du document (sections, tableaux, images, éléments, etc.). Au cours de cette étape la structure logique est analysée de manière incrémentale dans le but de la découvrir.
3. Indexer et segmenter la structure logique : la structure logique produite par l'étape précédente est annotée en utilisant une ontologie de domaine. Durant cette étape AIDAS regarde le contenu, par exemple en comparant le titre d'une section à la liste des concepts de l'ontologie. Si une correspondance est trouvée entre le titre et un concept, la section est indexée en tant que segment. Ce segment est étiqueté par le nom du concept correspondant.
4. Stocker le document dans une base multimédia : prendre les segments annotés et les convertir en base de données. Les segments textuels sont stockés au format ASCII et les segments graphiques au format SVG.

L'approche développée dans AIDAS est fondée sur l'idée que les composants géométriques contiennent des caractéristiques de mise en page explicites et que ces caractéristiques contiennent des indices à propos de la structure logique. Par exemple, un composant texte écrit en gras et grand (la forme) contient l'indication que ce composant pourrait être un titre de section (la fonction). Le système AIDAS utilise cette idée en assignant un ensemble de fonctions possibles à chaque composant géométrique et en les regroupant en composants plus complexes. Ce processus est répété de manière incrémentale jusqu'à ce que la structure logique soit établie. La détermination des fonctions possibles d'un composant géométrique peut être réalisée de manière ascendante, tandis que la fonction correspondante est déterminée de manière descendante à l'aide de grammaires spécifiques.

Pour rappel, dans la Section 1.2 nous avons évoqué comment le système de traitement d'images de documents proposé par (Aiello *et al.*, 2003) est capable d'extraire la structure géométrique de celles-ci. A présent, nous discutons de la méthode mise en œuvre par les auteurs pour extraire la structure logique associée. Aiello et al. (Aiello *et al.*, 2003) considèrent une relation d'ordre partielle nommée "*PrécèdeDansLaLecture*", originalement nommée "*BeforeInReading*", qui détermine pour deux objets si l'un est à lire avant l'autre. Cet ordre partiel peut être étendu à un ordre total parmi les objets d'un document. Cet ordre total est *l'ordre de lecture*. Les auteurs émettent l'hypothèse qu'une page de documents ne possède qu'un unique ordre de lecture. C'est une limitation du système car des exemples de pages de journaux contenant plusieurs articles indépendants et pouvant être lu indépendamment dans n'importe quel ordre ont été

identifiées par les auteurs. Dans la phase de compréhension de documents, deux étapes sont considérées : le regroupement et la classification des objets géométriques en objets logiques, puis la détermination des relations logiques entre l'ensemble des objets logiques. L'ordre de lecture est considéré comme relation logique. Ainsi, ces deux étapes combinées à l'utilisation de connaissances de documents (générique, spécifique (voir la Section 1.2)) aboutissent à l'établissement de la structure logique d'un document donnée. Les détails concernant les deux étapes évoquées peuvent être trouvés dans (Aiello *et al.*, 2003).

1.4 Évaluation des performances des systèmes de traitement automatique de documents

D'après (Mao *et al.*, 2003) et (Trupin, 2005) il est nécessaire de discuter de la difficulté d'évaluer les systèmes présentés dans la section précédente. (Haralick, 1994) indiquait dès 1994, qu'il est clair que beaucoup des systèmes présentés dans le paragraphe précédent donnent des résultats propres à la technique employée et sur des documents dédiés à leur utilisation. Dès que la diversité des documents à traiter est élargie, ces systèmes doivent avoir un niveau de performance élevé. Ceci implique de valider ces systèmes sur des bases de plusieurs milliers de documents et de disposer pour cela de mesures de performance adaptées tant pour l'extraction de la structure géométrique que pour la construction de la structure logique. Il est alors nécessaire de disposer de bases conséquentes de données d'images de documents avec des structures géométrique et logique correctement étiquetées pour chaque image. D'autre part, bien que des niveaux de performance importants soient nécessaires, très peu de méthodes présentées cherchent à optimiser leur performance sur une base de données. Mao et al. (Mao *et al.*, 2003) présentent un état de l'art dans lequel ils discutent de l'évaluation des performances des algorithmes développés sur la base de différents critères : métrique de performance, données expérimentales, spécification de "la vérité terrain", analyse des erreurs et évaluation comparative. Une métrique est nécessaire pour évaluer les performances d'un algorithme donné. Elle est fonction de la base de données, de "la vérité terrain" et des paramètres de l'algorithme. Elle n'est pas obligatoirement unique afin de permettre de sélectionner la métrique adaptée à une analyse de performance particulière. Krishnamoorthy et al. (Krishnamoorthy *et al.*, 1993) proposent une métrique fondée sur le pourcentage de zones correctement étiquetées. Niyogi et Srihari (Niyogi & Srihari, 1995) font appel à trois métriques pour la classification des blocs, la fusion des blocs et la précision de l'ordre de lecture trouvé. Le point commun à toutes ces métriques est le manque de définition formelle. Elles correspondent plutôt à des évaluations intuitives. Yamashita et al. (Yamashita *et al.*, 1991) décrivent une métrique basée sur une fonction de coût pour sélectionner le résultat de moindre coût. Kreich et al. (Kreich *et al.*, 1991) utilisent une métrique fondée sur la distance de Hamming (Cover & Thomas, 2006) pour calculer une mesure de confiance des correspondances entre une base de structures géométrique et logique de documents et un document donné. Summers (Summers, 1995) définit des métriques de précision pour évaluer la performance de son algorithme. Ces métriques sont déjà plus formelles et présentent ainsi moins d'ambiguïté dans leur interprétation. Une évaluation fondée sur des bases de grande taille est nécessaire pour évaluer objectivement les performances des algorithmes et une "vérité terrain" est indispensable pour

quantifier les résultats expérimentaux. Ainsi quelques auteurs ont utilisé des bases de test relativement importantes allant jusqu'à une centaine d'images. Une évaluation comparative de performances est nécessaire pour pouvoir comparer deux algorithmes sur le même jeu de données. Pourtant, pour la plupart des algorithmes, il n'existe pas d'évaluation comparative, sauf pour (Dengel & Dubiel, 1996) qui a effectué une évaluation comparative de son algorithme selon le choix d'une approche ascendante ou descendante. Mao et al. (Mao *et al.*, 2003) résument ainsi les limitations et les besoins des travaux sur l'analyse des structures de documents :

1. Une majorité de travaux présentés ne sont pas fondés sur des modèles formels pour les pages de documents. Pourtant un tel usage présenterait plusieurs avantages.
   — un modèle avec un niveau de complexité approprié pour une classe donnée de documents pourrait être utilisé ;
   — dès qu'un modèle a été choisi pour une classe de documents donnée, des exemples de cette classe pourraient être utilisés pour estimer les paramètres du modèle ;
   — des modèles formels pourraient à la fois être utilisés pour l'analyse et la synthèse de documents ; en effet, un modèle pourrait être validé en étant utilisé pour créer des images synthétiques de documents qui pourraient être comparées à des images réelles pour une classe donnée.
2. Une majorité de travaux sur la structure logique de documents supposent que la structure géométrique a déjà été extraite.
3. Une majorité de travaux utilisent des modèles déterministes qui sont très sensibles au bruit (photocopies, fax, etc), ce qui peut mener à des résultats ambigus ou faux.
4. Parfois, une évaluation quantitative a complètement été négligée.

Bien que les algorithmes présentés soient fondés explicitement ou implicitement sur des modèles de documents, peu d'entre eux fournissent une définition formelle de ces modèles. C'est ce qui rend difficile la caractérisation de la relation entre les modèles et la performance des algorithmes. De plus, les paramètres des algorithmes sont généralement choisis empiriquement. La génération d'images synthétiques pourrait permettre de simuler le fonctionnement du système et d'effectuer des expérimentations pour évaluer l'algorithme et notamment pour détecter ses faiblesses. De même, l'extraction de la structure géométrique peut produire des résultats erronés. Aussi, des modèles stochastiques, représentés par des grammaires stochastiques, pourraient être utilisés pour pallier ce problème en associant des probabilités aux structures géométriques extraites et au bruit apparaissant sur les documents. Finalement, (Trupin, 2005) soulève un questionnement : "Peut-on vraiment concevoir des modèles formels de documents ?"

1.5 Conclusion

Dans ce Chapitre nous présenté le cadre général du traitement automatique de documents. Le traitement d'images de documents consiste en la transformation dans un format compréhensible par un système informatique, de données présentes au sein de documents et compréhensibles par l'Homme. L'analyse et la compréhension de documents sont les deux phases du processus de traitement de documents. L'analyse de

documents consiste à extraire et isoler au sein de blocs, les mots, les lignes et les objets graphiques (tels que des logos) contenus dans un document. La compréhension de documents consiste à transformer la structure géométrique d'un document en sa structure logique. Une fois la structure logique établie, sa signification peut être décodée par une intelligence artificielle ou toutes autres techniques.
Dans ce mémoire nous sommes particulièrement intéressés par la phase d'analyse de documents. Pour rappel, l'objectif de la thèse est de proposer une méthode de traitement de documents numérisés, en vue, notamment, d'améliorer la performance d'outils de reconnaissance d'informations textuelles pertinentes au sein de ces documents. D'une part, notre première contribution (Partie II Chapitre 5) s'inscrit dans le cadre des approches hiérarchique descendante d'analyse de documents. Nous présentons un algorithme de segmentation d'images de factures fondée sur la décomposition Quadtree, pour la localisation et l'extraction d'informations textuelles données. D'autre part, notre deuxième contribution (Partie II Chapitre 6), s'inscrit dans le cadre des approches de localisation et d'extraction d'informations textuelles dans des images de documents, fondées sur les régions les contenant. Le Chapitre suivant présente un état de l'art des approches d'extraction d'informations textuelles au sein d'images de documents.

Chapitre 2

Extraction d'informations textuelles au sein d'images de documents

Sommaire

2.1 Caractéristiques inhérentes au texte contenu au sein d'images 47
2.2 Approches d'extraction d'informations textuelles 50
2.2.1 Approches fondées sur des régions 51
2.2.2 Approches fondées sur les contours 52
2.2.3 Approches morphologiques 53
2.2.4 Approches fondées sur les textures 53
2.3 Approches d'extraction d'informations textuelles au sein d'images de factures 54
2.4 Problématiques d'évaluation des systèmes d'extraction d'informations textuelles 59
2.4.1 Problématiques générales de création de jeux de données . . . 60
2.4.2 Avantages et inconvénients des données réelles et des données synthétiques 62
2.4.3 Mesures pour l'évaluation de performance 62
2.5 Conclusion 63

Dans le Chapitre 1 nous avons vu en quoi consiste le traitement automatique de documents. Une des étapes du traitement de documents est l'analyse de documents. L'objectif final de l'analyse de documents est d'extraire et de reconnaitre les composants textuels et graphiques présents au sein d'images de documents. L'extraction d'informations textuelles (dite aussi de texte) au sein d'images est un problème important dans beaucoup d'applications telles que le traitement automatique de documents (Li *et al.*, 2014; Tang *et al.*, 1996), l'indexation d'images (Juneja *et al.*, 2015), le résumé de documents (Goldstein *et al.*, 2000), la recherche d'images (Dixit & Shirdhonkar, 2015; Dharani & Aroquiaraj, 2013) et bien d'autres encore. Dans ce Chapitre nous présentons tout d'abord dans la Section 2.1 les caractéristiques que comportent les éléments textuels présents au sein d'images. Dans la Section 2.2 nous étudions les approches d'extraction de texte au sein d'images de différents types avant d'aborder l'extraction de texte au sein d'images de factures en particulier dans la Section 2.3. Enfin, nous évoquons dans la Section 2.4 les problématiques liées à l'évaluation des systèmes d'extraction de texte.

2.1 Caractéristiques inhérentes au texte contenu au sein d'images

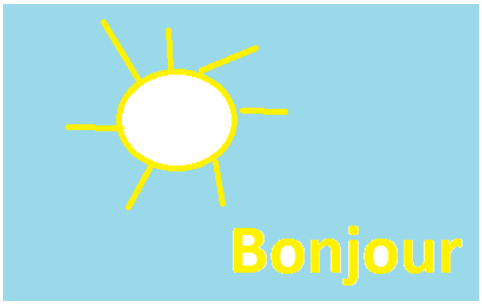
La Figure 2.1 présente différents types d'images. On distingue principalement trois types d'images :

- les images de documents : obtenues par la numérisation de documents tels que des livres, des journaux, des factures, des formulaires, des rapports, etc (Figure 2.1c et d).
- les images de scènes réelles : obtenues grâce à un appareil de capture d'images tels qu'un appareil photo et représentant une scène de la vie réelle, par exemple, la photo d'une voiture de gendarmerie (Figure 2.1e).
- les images graphiques : obtenues à l'aide d'un outil de dessin tel que Paint, Inkscape ou Photoshop (Figure 2.1a et b).

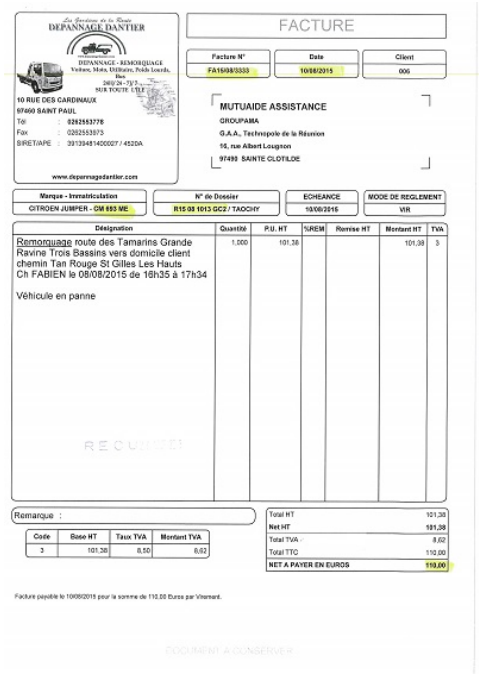
Sumathi et al. (Sumathi *et al.*, 2012b) décrivent les différents défis liés à l'extraction de texte au sein d'images. La détection de texte au sein d'images de scènes réelles apparait comme étant une tâche difficile. En effet, contrairement aux images où du texte a été ajouté artificiellement (Figure 2.1a et b), les images de scènes réelles (Figure 2.1e), qui contiennent du texte naturellement présent, peuvent avoir différentes orientations et peuvent être déformées par la perspective. De plus, elles sont souvent affectées par des variations de paramètres liées à la scène (mouvement, luminosité, etc) ou à l'appareil (caméra, appareil photo) utilisé (zoom, luminosité, contraste, etc). Les images contenant du texte incrusté artificiellement et les images de documents présentent des caractéristiques similaires et sont généralement traitées de la même manière. Par la suite nous ne nous intéresserons exclusivement aux images de documents qui sont au centre de la problématique de recherche traitée dans ce mémoire. Plus de détails concernant le traitement des images de scènes réelles sont disponibles dans (Jung *et al.*, 2004; Lienhart & Wernicke, 2002; Zhang & Kasturi, 2008; Chen *et al.*, 2004; Zhang *et al.*, 2013; Doermann *et al.*, 2003).
Certaines propriétés inhérentes au texte présent dans les images impactent directement le processus d'extraction. En effet, les textes peuvent subir différents changements d'apparence tels que la fonte, la taille, le style, l'orientation, l'alignement, la texture, la couleur, le contraste et l'arrière plan (Jung *et al.*, 2004). Tout ces changements compliquent et rendent plus difficile le problème d'extraction de texte. Plusieurs chercheurs tels que (Yin *et al.*, 2014; Sumathi & Devi, 2014; Raj & Ghosh, 2014; Seeri *et al.*, 2016; Sung *et al.*, 2015; Gorski *et al.*, 2001; Ye *et al.*, 2001) ont proposé différentes méthodes pour pallier les difficultés citées ci-dessus. En effet, le texte présent au sein d'images peut subir des variations vis à vis des propriétés suivantes :

- Géométrie :
  - la taille : les intervalles entre les tailles de fonte disponibles peuvent être grands ;
  - l'alignement : les caractères textuels peuvent présenter des distorsions et différentes orientations ; cette propriété impacte négativement la performance d'un système d'extraction ;
  - la distance entre caractères : il peut y avoir des distances différentes entre les caractères d'une même ligne de texte ;
- Couleur : les caractères d'un texte tendent à avoir des couleurs similaires voir la

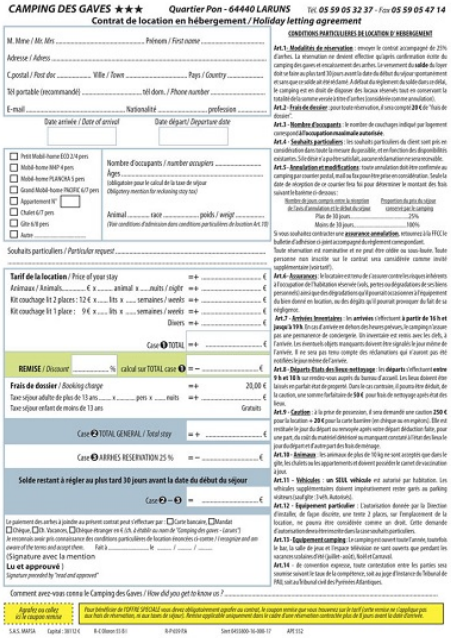
(a) Photographie avec incrustation artificielle de texte.



(b) Image, contenant le texte "Bonjour", réalisée avec Paint.



(c) Image d'une facture numérisée contenant du texte.



(d) Image d'un contrat de location numérisé contenant du texte et un formulaire.

(e) Photographie d'une scène réelle contenant du texte incrusté de manière naturelle sur une plaque d'immatriculation.

FIGURE 2.1 – Exemples d'images contenant du texte.

même couleur ; cette propriété rend possible l'utilisation des approches fondées sur des composants connectés pour la détection de texte (Phan *et al.*, 2009; Ye *et al.*, 2007) ;
— Contour : certains textes présentent des contours épais afin d'en faciliter la lecture ;
— Compression : les images sont souvent compressées pour pouvoir être partagées facilement (email, disque de stockage amovible, etc). Un système d'extraction sera d'autant plus rapide qu'il sera capable d'extraire du texte au sein d'une image sans la décompresser.

Rappelons que les images de documents sont des images numériques d'objets tels que des documents administratifs (constats, factures, déclarations, etc.), des formulaires, des cartes topographiques, des rapports techniques, des chèques, etc. Ces images numériques peuvent être produites par un scanner ou un fax par exemple. Les images de documents ainsi obtenues sont généralement stockées sous forme de fichiers au format jpeg, tiff, bmp, png, etc. Ces images peuvent être en couleur ou en niveaux de gris. Comme détaillé dans le Chapitre 1, l'analyse d'images de documents est une des deux étapes du traitement de documents. L'objectif de l'analyse de documents est d'extraire et de reconnaitre les composants textuels et graphiques contenus dans ces documents. Elle consiste à transformer une image de document dans un format compréhensible par une machine. L'analyse de documents est souvent confondue avec la tâche de recherche d'images de documents, également nommée recherche de documents. Pourtant, l'objectif de la recherche de documents est principalement d'indexer et de retrouver une image de document à partir d'une requête, au sein d'une grande base de données d'images de documents. Bien que l'objectif ultime de l'analyse de documents soit d'abandonner le papier au profit du tout numérique, elle est également utilisée en recherche de documents par exemple pour la vérification de signature de chèques, la détection de fraude au sein des sociétés d'assurance, ou encore l'archivage et l'indexation de banques de documents historiques ou légaux. Selon (O'Gorman & Kasturi, 1995), l'analyse d'images de documents comprend les étapes suivantes :

— Binarisation et pré-traitement : une image de document peut être binarisée afin de séparer les informations de premier plan et celles de l'arrière plan. Dans cette étape l'image de document est convertie en pixels de niveau d'intensité 0 ou 1. Le pré-traitement comprend la réduction du bruit, la segmentation et la conversion de l'image de document dans une forme requise pour les traitements suivants. Du bruit peut être introduit dans l'image de document à cause de différentes sources de dégradation comme le temps, la photocopie de l'image ou lors de son acquisition. Le processus de segmentation consiste à séparer les composants textuels et graphiques de l'image de document. D'une part, la segmentation des informations textuelles permet d'identifier des colonnes, paragraphes, mots et caractères. D'autre part, la segmentation des éléments graphiques consiste à séparer des symboles, logos, signatures et lignes contenus dans l'image de document.
— Analyse des éléments contenus : cette étape consiste à analyser les composants textuels et graphiques de l'étape précédente. Les caractéristiques structurelles et textuelles sont extraites de l'image de document. En particulier, deux types d'analyses sont appliquées au texte contenu dans l'image de document. Le premier est une reconnaissance optique de caractère (OCR) afin d'extraire le sens des caractères et des mots contenus. Le deuxième est une analyse de la mise en page

(voir Chapitre 1) de l'image afin de reconnaitre la mise en forme du texte contenu en différents blocs fonctionnels, en-têtes, pieds-de-page, titres, sous-titre, etc..
— Description du document : le résultat de l'analyse d'une image de document est une description du document. Cette description consiste en la description des composants graphiques et textuels présents dans le document (mise en page, relations spatiales, localisation, contenu textuel résultant de l'OCR).

Dans ce contexte l'extraction de texte au sein d'images peut être vue comme une tâche particulière jouant un rôle dans un système d'analyse de documents. Dans la Section 2.2 différentes approches d'extraction de texte au sein d'images de manière générale sont évoquées avant d'étudier l'extraction de texte au sein d'images de factures en particulier dans la Section 2.3. Les problématiques concernant l'évaluation de la performance des systèmes d'extraction d'informations textuelles sont discutées dans la Section 2.4.

2.2 Approches d'extraction d'informations textuelles

Dans la littérature (Agrawal & Varma, 2012; Sumathi *et al.*, 2012b; Tehsin *et al.*, 2014), l'extraction d'informations textuelles au sein d'images peut être décomposée en cinq tâches :

— *la détection de texte* qui consiste à déterminer la présence de texte dans une une image donnée ;
— *la localisation de texte* qui consiste à déterminer la localisation de contenu textuel au sein d'une image donnée et de générer des régions rectangulaires autour du texte ;
— *le suivi de texte* qui consiste à créer des groupes de régions textuelles, est réalisé afin de réduire le temps de traitement pour la localisation de texte ; bien que la localisation précise du texte contenu dans une image donnée peut être indiquée par des régions rectangulaires, le texte nécessite encore d'être séparé de l'arrière plan de l'image afin de faciliter sa reconnaissance ; cela signifie que l'image de texte localisée doit être convertie en image binaire et renforcée avant d'être traitée par un moteur d'OCR ;
— *la reconnaissance de texte* qui consiste à séparer les composants textuels de l'arrière plan d'une image donnée ; par la suite, les composants peuvent être transformés en texte brute en utilisant un moteur d'OCR ;
— *le renforcement de texte* des composants textuels, qui consiste à améliorer la qualité des composants textuels, peut être réalisé lors de la tâche précédente afin d'améliorer la reconnaissance de ces composants par un moteur d'OCR ; en effet, les régions textuelles identifiées peuvent avoir une faible résolution et contenir du bruit ;

Il est à noter que ce processus peut s'appliquer à tous les types d'images et de vidéos dans lesquelles du texte apparait de manière naturelle ou de manière artificielle. Le schéma de la Figure 2.2 montre l'enchainement des étapes d'extraction d'informations textuelles. Les étapes de suivi et de renforcement de texte sont optionnelles et sont généralement appliquées à l'extraction d'informations textuelles au sein d'images de vidéo.

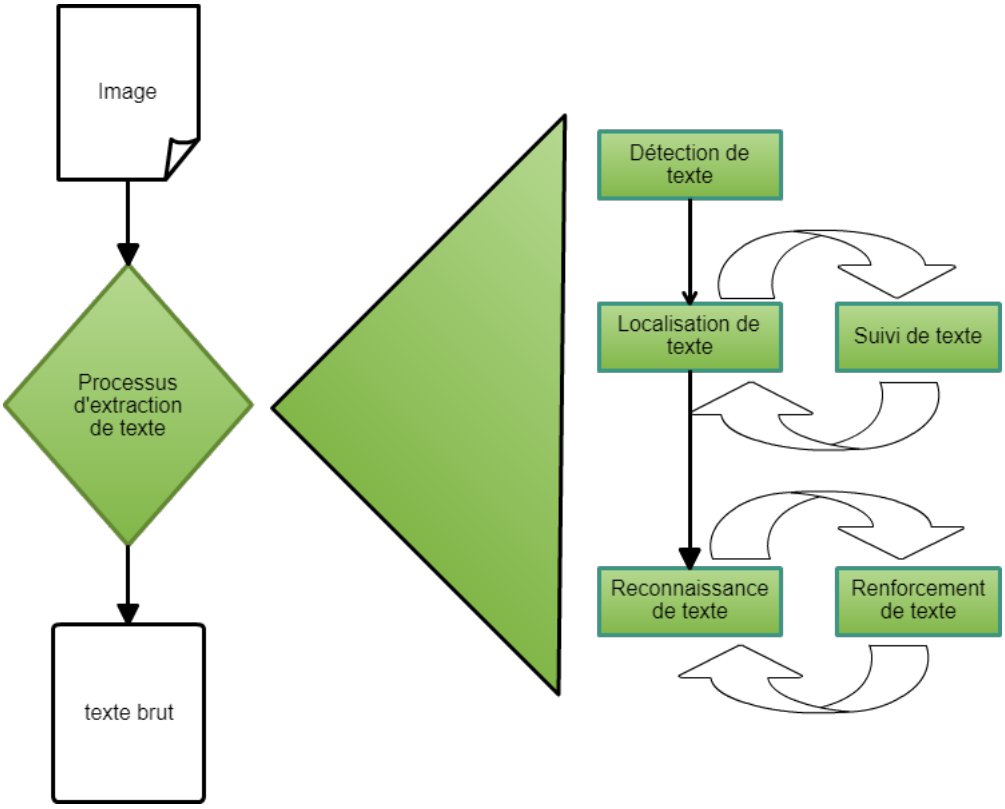


FIGURE 2.2 – Processus d'extraction d'informations textuelles au sein d'images.

Comme évoqué précédemment, l'extraction de texte au sein d'images comportent au moins trois des étapes citées plus haut : détection, localisation, reconnaissance. Les étapes de détection et de localisation de texte sont étroitement liées mais ne sont pas à confondre. L'objectif de ces deux étapes est de générer des régions rectangulaires autour de tous les éléments textuels présents dans des images de documents et d'identifier de manière unique chaque élément. Tandis que l'étape de détection permet d'identifier si un élément est du texte ou non, l'étape de localisation consiste à délimiter et à identifier la position de l'élément textuel détecté (lors de l'étape de détection) en l'encapsulant au sein d'une région rectangulaire. A notre connaissance, la littérature récente est relativement peu fournie en ce qui concerne l'extraction de texte au sein d'images de documents en particulier. Les principales études récentes menées sont celles de (Sumathi *et al.*, 2012b,a) et (Jung *et al.*, 2004) et concernent de manière générale les types d'images évoqués plus haut. On distingue quatre types d'approches dans la littérature :

— les approches fondées sur des régions ;
— les approches fondées sur les contours ;
— les approches morphologiques ;
— les approches fondées sur les textures.

2.2.1 Approches fondées sur des régions

Les approches fondées sur des régions s'appuient sur les propriétés relatives à la colorimétrie d'une région textuelle ou sur les différences de ces propriétés entre une région textuelle et l'arrière-plan. Ces approches sont ascendantes et consistent à regrouper de

petits composants en composants plus grands jusqu'à ce que toutes les régions d'une image de document soient traitées. Une analyse géométrique est nécessaire pour fusionner les composants textuels en s'appuyant sur leur arrangement spatial afin de filtrer les composants qui ne sont pas textuels et de délimiter les régions textuelles.
Par exemple, Sobottka et al. (Sobottka *et al.*, 1999) proposent une approche pour la détection et la localisation automatique de texte au sein de couvertures de livres et de revues en couleur. Afin de réduire le nombre de variations de couleur, un algorithme de classification développé par (Matas & Kittler, 1995) est appliqué dans une phase de prétraitement. Deux méthodes ont été développées pour l'extraction de texte. La première est une analyse descendante mettant en œuvre une division successive des régions d'une image. La deuxième est un algorithme ascendant pour regrouper les régions en régions plus grandes. Les résultats des deux méthodes sont combinées et permettent de distinguer de manière robuste les éléments textuels et non-textuels. Les éléments textuels sont ensuite binarisés et sont utilisées en entrée d'un moteur d'OCR traditionnel. L'approche proposée ne se limite pas aux pages de couverture, mais peut aussi être appliquée à l'extraction de texte au sein d'autres types d'images de documents en couleur.

2.2.2 Approches fondées sur les contours

Les approches fondées sur les contours sont adaptées aux images contenant du texte incrusté aussi bien naturellement qu'artificiellement. Elles permettent de localiser et d'extraire rapidement et efficacement du texte au sein d'images. Les contours possèdent des caractéristiques d'intensité, de densité et de variation d'orientation qui en font des éléments majeures pour la détection de texte.
Par exemple, la méthode proposée par (Liu & Samarabandu, 2006) comporte trois étapes :

1. la détection de régions textuelles candidates ;
2. la localisation des régions textuelles ;
3. la reconnaissance de caractères.

Dans la première étape, l'ordre de grandeur de la dérivée seconde de l'intensité est calculée afin de mesurer l'intensité des contours et permet une meilleure détection de l'intensité d'arêtes qui normalement caractérisent le texte contenu dans des images. La densité des contours est également calculée à partir de l'intensité moyenne de ces contours au sein d'une fenêtre rectangulaire. En considérant l'efficacité et le rendement, quatre orientations (0°, 45°, 90°, 135°) sont utilisées pour évaluer la variation des orientations. 0° indique une direction horizontale, 90° indique une direction verticale, et 45° et 135° représentent les deux directions diagonales. La détection de contours est effectuée en utilisant une stratégie multi-échelles, où des images multi-échelles sont produites par des pyramides Gaussiennes (Adelson *et al.*, 1984) après application successive d'un filtre passe-bas [1] et d'un procédé d'échantillonnage de l'image originale réduisant ainsi l'image verticalement et horizontalement.
Dans la seconde étape, des caractéristiques de regroupement des caractères peuvent être utilisées pour localiser les régions contenant du texte. Les caractères identifiés comme

1. Un filtre passe-bas est un filtre utilisé en traitement d'images qui laisse passer les basses fréquences et qui atténue les hautes fréquences.

étant proches selon une certaine mesure de proximité sont isolés au sein d'une même région, tandis que les caractères éloignés sont laissés de côté.
Dans la troisième étape, un moteur d'OCR existant est utilisé pour effectuer la reconnaissance des régions textuelles détectées et localisées. Cette approche ne traite que les caractères imprimés sur un arrière plan "net" et ne peut pas traiter des caractères incrustés sur des fonds texturés, ombrés ou complexes.

2.2.3 Approches morphologiques

Les approches morphologiques sont fondées sur une approche topologique et géométrique pour l'analyse d'images qui est la morphologie mathématique. Ce type d'approche fournit de puissants outils pour l'extraction de structures géométriques dans beaucoup d'applications. Des techniques d'extraction de caractéristiques morphologiques ont été efficacement appliquées à la reconnaissance de caractères et à l'analyse de documents. A partir d'une image, le traitement consiste à extraire des caractéristiques textuelles relatives au contraste.
Wu et al. (Wu *et al.*, 2008) présentent un algorithme d'extraction de lignes de texte fondé sur la morphologie. Cet algorithme est appliqué à l'extraction de régions textuelles au sein d'images bruitées. Tout d'abord, la méthode définit un nouvel ensemble d'opérations morphologiques pour l'extraction des zones de contraste important qui sont des lignes de texte candidates possibles. Dans le but de détecter des lignes obliques de texte, une méthode fondée sur des moments (Bowman & Shenton, 2004) est ensuite utilisée pour estimer leur orientation. Selon l'orientation, une technique de projection sur l'axe des abscisses peut être appliquée afin d'extraire différentes géométries de texte à partir de segments de texte analogues pour de la vérification textuelle. Cependant, à cause du bruit, une ligne de texte est souvent fragmentée en plusieurs morceaux de segments. Par la suite, un schéma de vérification est proposé pour la vérification de toutes les lignes de texte potentielles extraites selon leur géométrie textuelle.

2.2.4 Approches fondées sur les textures

Les approches fondées sur la texture utilisent le fait que le texte contenu dans des images possèdent des propriétés de texture particulières qui les distinguent de l'arrière plan. Ces approches appliquent des transformations telles que la transformation de Fourier rapide (Baker *et al.*, 2014), la décomposition en ondelettes (Shekar *et al.*, 2014; Seeri *et al.*, 2015) et des filtres de Gabor (Raju *et al.*, 2004), pour l'extraction des propriétés de texture.
Goel et Sharma (Goel & Sharma, 2014) proposent un algorithme pour l'extraction de texte au sein d'images colorées en utilisant la décomposition en ondelettes de Haar 2D (2D-DWT) ainsi que des opérateur morphologiques. L'algorithme a été testé pour l'extraction de texte au sein d'images de plaque d'immatriculation et de documents.
Sumathi et Gayathri (Sumathi & Devi, 2014) proposent une méthode d'extraction de régions textuelles et de suppression de régions non textuelles au sein d'images colorées comportant un arrière plan complexe. La méthode est fondée sur une correction Gamma en déterminant une valeur gamma pour le renforcement des détails du premier plan d'une image. L'approche utilise également des matrices de co-occurrence de niveaux de gris et

des mesures de textures.

Dans la littérature plusieurs chercheurs se sont intéressés en particulier à l'extraction d'informations textuelles au sein de documents administratifs tels que des factures. Les services administratifs tels que les organismes publiques, les services de santé, les banques, les assurances et les entreprises de toute sorte traitent quotidiennement de grandes quantités de documents (formulaire, déclaration, contrat, rapport, factures). Face à cela, des efforts sont faits pour réduire les charges que le traitement de ces documents représentent. Les délais et les erreurs de traitement tendent à être réduits par la mise en place de nouveaux services permettant de partager rapidement ces documents au format électronique sur internet. Une solution est de transformer le flux de documents papiers en un flux de documents électroniques. Par exemple, les déclarations annuelles de ressources se font de plus en plus via un formulaire en ligne plutôt que sur papier. Néanmoins, certains documents tels que les factures sont encore le plus souvent traités au format papier. Les approches d'extraction de texte au sein d'images de factures sont abordées dans la section suivante.

2.3 Approches d'extraction d'informations textuelles au sein d'images de factures

Le traitement de factures, automatiquement ou manuellement, est une tâche quotidienne au sein de beaucoup d'entreprises. Le volume de factures à traiter dépend fortement de l'activité de chaque entreprise. Par exemple, une assurance reçoit un grand nombre de demandes de remboursement chaque jour. Ces factures contiennent des informations similaires, mais ces informations sont distribuées au sein des factures selon un grand nombre de styles de mise en page différents. Les factures contiennent des informations obligatoires (numérotation de la facture, identifiant unique de la société émettrice, montant hors taxe, taux de tva, montant de tva, montant net, la mention "facture", date d'émission de la facture, etc.) qui, selon l'émetteur peuvent se trouver à des endroits différents dans le document. Néanmoins, pour un ensemble d'émetteurs différents des similitudes peuvent apparaitre localement pour une ou plusieurs informations données. Prenons l'exemple des informations concernant le montant d'une prestation de service : montant hors taxte, montant tva, montant net. Dans le système français, ces informations sont généralement positionnées en bas à droite des factures. De la même manière, l'identité de la société émettrice apparait généralement en haut à gauche des factures. A l'inverse, pour d'autres informations telles que la date d'émission et le numéro de la facture, il est plus délicat d'établir une tendance car ces informations ne semblent pas avoir une localisation spécifique au sein des factures. On distingue deux catégories d'informations :

1. Les informations d'en-tête ou de pied de page : les données concernant l'émetteur, le destinataire, la date de facture, le numéro de facture, la date d'échéance de paiement, etc.

2. Les informations de tableaux ou de corps : les détails sur les articles ou services facturés ligne par ligne, les montants (hors taxe, taxe, net).

L'extraction automatique des éléments d'une facture donnée est l'objectif des systèmes de traitement ou de lecture automatique de factures. Au cours des 20 dernières années, plusieurs éditeurs de logiciels ont proposé des systèmes intelligents pour lire automatiquement des documents administratifs. Les éditeurs les plus connus sont EMC-Captiva [2], Kofax [3], Top Image System [4], Itesoft [5], A2IA [6], ABBYY [7], Nuance [8], Mitek [9] et Parascript [10]. Les performances de ces systèmes sont correctes lorsqu'ils sont appliqués à des taches d'entreprise précises et à un seul type de documents tels que des chèques ou des formulaires. Les performances de ces systèmes se dégradent lorsqu'ils doivent faire face à des documents de sources variées et de plus grande complexité. Les difficultés rencontrées sont liées à l'hétérogénéité des informations à extraire :

- les informations peuvent être de haut niveau, par exemple la classe de document (facture, formulaire, déclaration), ou de bas niveau comme par exemple un numéro de facture ou un numéro de sécurité sociale, ou des tableaux (une liste d'articles ou de montants) ;
- les informations peuvent être imprimées à la machine (codes barre, caractères, symboles, logos) ou imprimées à la main (annotations, écritures manuscrites) ;
- les informations peuvent être très structurées au sein d'un formulaire ; dans les formulaires chaque champ est toujours localisé à la même position géographique ;
- les informations peuvent être semi-structurées, c'est le cas des chèques ou des factures ; les documents de ce type semblent structurés mais les positions des champs peuvent varier ; chaque émetteur est libre de produire ces documents comme il le souhaite selon le respect de quelques principes légaux ; par exemple sur une facture le numéro de facture doit obligatoirement être indiqué non loin de la mention "numéro de facture" ou une mention similaire ;
- les informations peuvent être non structurées à la manière d'un texte libre, par exemple un courrier rédigé à la main ;
- les informations peuvent être isolées et faciles à localiser, par exemple un texte imprimé sur un fond blanc, ou présenter des chevauchements et être, par conséquent, difficiles à extraire (texte imprimé sur un fond texturé ou texte surligné à la main) ;
- les informations peuvent être aussi bien en couleur qu'en noir et blanc ;

Tous les systèmes développés par les éditeurs cités précédemment dépendent de deux types de technologies :

- les classifieurs permettant de convertir des pixels en symboles : OCR (reconnaissance optique de caractères) (Liu *et al.*, 2013; Bissacco *et al.*, 2013), ICR (reconnaissance intelligente de caractères) (Kumar & Vijayabhasker, 2016; Hussain *et al.*, 2016), IWR (reconnaissance intelligente de mots) (Bunke, 2003; Grosicki & El Abed, 2009), etc. ;
- les stratégies de segmentation et d'extraction telles que l'appariement de modèles

2. https ://www.dellemc.com/en-us/index.htm
3. http ://www.kofax.com/
4. https ://www.topimagesystems.com/
5. https ://www.itesoft.com/
6. www.a2ia.com/fr
7. https ://www.abbyy.com/
8. http ://www.nuance.fr/index.htm
9. https ://www.miteksystems.com/
10. https ://www.parascript.com/

afin de localiser et d'interpréter des champs à extraire (Ye *et al.*, 2001; Chen, 2015).

Dans les méthodes fondées sur l'appariement de modèles, l'extraction d'informations textuelles est guidée par un modèle (ou masque). Le modèle définit un emplacement physique (une région rectangulaire) pour chaque champ à extraire au sein d'une image. Les systèmes fondés sur ces méthodes interprètent un document candidat en appliquant un OCR ou un ICR sur l'objet délimité par la région rectangulaire. Dans les systèmes industriels, les descriptions de modèles sont généralement obtenues par des paramétrages manuels via une interface graphique. Ces systèmes s'avèrent efficaces pour des documents dont la structure est fixée, mais sont consommateurs de temps et difficiles à maintenir lorsque l'utilisateur doit définir beaucoup de modèles. Plusieurs chercheurs comme (Bartoli *et al.*, 2014), (Medvet *et al.*, 2011) et (Belaïd *et al.*, 2011) ont proposé des méthodes de construction automatique de modèles. Ces méthodes ont en commun la recherche d'un document similaire à un document candidat, au sein d'une base. Au document similaire trouvé est associé un "modèle" listant un certain nombre d'attributs (position, type, mots-clés) permettant de localiser une information donnée au sein d'un document candidat.

Bartoli et al.(Bartoli *et al.*, 2014) proposent une méthode semi-supervisée d'extraction, au sein de documents numérisés, des éléments pré-définis, en se basant sur la sélection d'une *enveloppe* spécifique à ces documents appelée *wrapper*. Un wrapper est un objet contenant des informations sur les propriétés géométriques et textuelles des éléments à extraire. Le système proposé consiste à sélectionner, pour un document en entrée, le wrapper le plus proche selon une mesure de distance. Lorsque celui ci n'existe pas, le système propose à l'utilisateur de sélectionner manuellement depuis une interface graphique les éléments à extraire. Dans le mode manuel, le système génère par la même occasion un nouveau wrapper basé sur les sélections manuelles de l'utilisateur.

Medvet et al. (Medvet *et al.*, 2011) présentent une approche probabiliste pour l'étiquetage logique de factures. Leur approche est fondée sur un modèle, où un modèle représente des documents d'une même classe, c'est à dire des factures d'une même société. Un document D est représenté par un ensemble de blocs $\{b_1, b_2, ...\}$. Un bloc est constitué d'une position, une taille et un contenu. La position est constituée d'un numéro de page p et des coordonnées x et y de l'origine de la page (le point haut gauche). La taille est identifiée par une largeur w et une hauteur h. Le contenu est spécifié par une ligne de texte l exprimée sous la forme d'une séquence de caractères. Un bloc b est alors identifié par un tuple d'attributs $b = \langle p, x, y, w, h, l \rangle$. La construction d'un modèle consiste à générer un ensemble de règles étant donné un ensemble de documents. Chaque règle est constituée d'une cardinalité, une probabilité de correspondance P et une fonction d'extraction. La cardinalité définit la longueur de la séquence de blocs sur laquelle la règle sera appliquée. La probabilité de correspondance désigne la probabilité qu'une séquence B contient une valeur v correspondant au contenu d'une information à extraire. La fonction d'extraction est une fonction permettant de calculer la valeur v de l'information à extraire. Étant donné un modèle et un ensemble de règles, l'extraction d'une information au sein d'un document consiste pour chaque règle du modèle à :

1. trouver la séquence $B*$ qui correspond et qui maximise la probabilité P ;

2. extraire la valeur v à partir de la séquence qui correspond.

Belaïd et al. (Belaïd *et al.*, 2011) proposent une méthode de traitement de factures, fondée sur le principe de raisonnement à partir de cas. Un cas correspond à la co-existence d'un problème (les mots clés d'une adresse ou les lignes d'un tableau indiquant des montants, par exemple) et de sa solution (le contenu de ces éléments). Le problème est comparé à une base (de cas) de documents (dont la solution est connue) à l'aide de graphes. Les auteurs proposent un système dont le but est d'organiser et de retrouver les différents cas à partir d'une base de connaissance de cas. Au sein de la base de connaissance chaque problème est représenté selon deux niveaux : la formulation du problème et la structure du problème. Pour un document candidat, leur approche consiste à extraire le problème puis à sélectionner au sein de la base de connaissance de cas le problème le plus similaire. Deux configurations sont alors possibles :

1. plusieurs problèmes similaires existent dans la base. Dans ce cas la solution du problème le plus proche (selon une mesure de proximité) est appliquée au problème du document candidat ;
2. il n'existe pas de problème similaire dans la base ; dans ce cas on procède à la comparaison des structures des problèmes ; la solution d'un problème dont la structure est la plus proche de celle du problème du document candidat est alors appliquée à ce dernier ; enfin, la solution du nouveau cas est intégrée dans la base de cas ;

La structure du problème d'un document contient la position (haut, gauche, droit, bas) et l'étiquette (alphabétique, numérique, alphanumérique) des éléments du document tels que : des mots, des champs (par exemple une date suivie de sa valeur), des champs alignés horizontalement, des champs alignés verticalement. La recherche d'un problème similaire est basée sur une comparaison des graphes représentant des documents. Le graphe du document candidat est comparé à tous les graphes des documents de la base. L'originalité du système proposé provient de la phase d'apprentissage qui y est ajoutée. La résolution d'un nouveau cas est ajoutée à la base de cas et une classification des cas est effectuée afin de regrouper les cas similaires ensemble, et d'améliorer la précision de la comparaison entre un futur cas et la base de cas. La méthode d'apprentissage adoptée consiste à améliorer un réseau de neurones incrémental nommé *Incremental Growing Neural Gas* qui réalise déjà un apprentissage incrémental.

Néanmoins, lorsque le flux de documents contient des documents hétérogènes qui ne peuvent pas être décrit par un seul modèle, une étape de classification est requise pour la sélection des modèles adaptés. Beaucoup d'algorithmes de classification ont été explorés comme par exemple les arbres de décision, les réseaux de neurones et les machines à vecteur de support.
Cesarini et al. (Cesarini *et al.*, 2003) proposent un système pour traiter les documents qui peuvent être regroupés en classes. Les factures sont utilisées comme cas d'étude. Le système comprend trois phases :

1. analyse de documents ;
2. classification de documents ;
3. compréhension de documents.

La phase d'analyse de documents consiste à extraire la structure géométrique d'une facture. Les objets géométriques résultants sont : les lignes verticales et horizontales, les objets graphiques (par exemple un logo), des régions rectangulaires contenant du texte. Les régions textuelles, en particulier, sont étiquetées numérique, alphabétique ou alphanumérique.
La phase de classification consiste à appliquer des arbres de décision à la structure physique d'une facture afin de déterminer sa classe. Une classe correspond à une société émettrice.
La phase de compréhension consiste à extraire la structure logique d'une facture et de l'associer à la structure géométrique correspondante. Si la classe de la facture est connue, le système tente de localiser un objet déterminé (le numéro de facture par exemple) à partir d'une étiquette précisant : la position absolue de l'élément dans la facture, et une liste de mots clés permettant de réaliser un test de correspondance de chaines de caractères avec le contenu textuel extrait par OCR à la position indiquée. Si la classe de la facture est inconnue, un autre type d'étiquette est utilisé pour localiser l'objet considéré. Ce type d'étiquette est dit indépendant du domaine de connaissance. Il contient des informations telles que :

- le nom de l'objet,
- le type de donnée qu'il contient,
- une liste de positions absolues et l'occurrence à laquelle l'objet a été observé à chaque position au sein d'un ensemble de factures,
- une liste de positions relatives et les occurrences associées,
- une liste complète de mots clés.

Les détails concernant les stratégies d'utilisation des contenus des étiquettes sont disponibles dans (Cesarini *et al.*, 2003).

Bartoli et al. (Bartoli *et al.*, 2010) proposent une méthode de classification de factures numérisées selon la densité de leurs pixels noirs et la densité des contours de texte. Deux classifieurs sont considérés : machine à vecteur de support et un classifieur fondé sur une mesure de distance.

Un système commercial nommé "smart FIX" est présenté dans (Klein & Dengel, 2003) pour l'analyse de factures provenant d'une variété de sources. Le système implémente un procédé de compréhension de document en plusieurs phases :

1. numérisation et de pré-traitement des factures ;
2. classification des images de factures ;
3. vérification des éléments extraits.

Depuis que le système a été déployé dans plusieurs compagnies d'assurance en santé d'Allemagne, il couvre une variété complète d'étiquettes logiques afin de supporter les différents processus de travail des compagnies.
Un panorama des techniques de classification utilisées pour la classification d'images de documents est disponible dans (Chen & Blostein, 2007) et (Chen, 2015).

La construction automatique de modèles est une approche intéressante mais est limitée à une ou peu de classes de documents similaires. En effet les caractéristiques et

les structures de documents sont propres à chaque domaine d'application : factures, chèques, formulaires ou tableaux. C'est pourquoi même si la méthode permet une bonne flexibilité pour un seul domaine d'application, elle peut difficilement traiter des flux de documents hétérogènes.
Une alternative à l'approche fondée sur l'appariement de modèles est l'approche fondée sur l'intégralité du texte. Dans cette approche l'extraction de texte est guidée par les données elles-mêmes et un ensemble de règles afin de déterminer des étiquettes logiques (Kim *et al.*, 2001). Les règles peuvent être paramétrées par un utilisateur ou entrainées sur des documents étiquetés comme dans (Cesarini *et al.*, 2002) où un arbre M-X-Y (Cesarini *et al.*, 1999) est entrainé pour l'apprentissage de structures de tableaux. D'autres méthodes comme celles de (Coüasnon, 2006) et de (Lee *et al.*, 2014) sont fondées sur une grammaire formelle. Coüasnon présente le système DMOS, une méthode générique de reconnaissance pour des documents structurés. La méthode s'appuie sur un formalisme grammatical nommé *Enhanced Position Formalism* et un parser associé qui permet de modifier la structure analysée durant l'analyse afin de pallier à des problèmes de segmentation. La généricité de la méthode a été validée sur plusieurs types de documents tels que des formules mathématiques, des structures de tableaux imbriquées et des partitions musicales. Lee et al. (Lee *et al.*, 2014) présentent un algorithme de similarité fondé sur un corpus sémantique et grammatical pour le traitement de phrases du langage naturel. L'approche proposée tire avantage d'un corpus de règles grammaticales et d'ontologies pour comparer des phrases à la syntaxe et à la grammaire complexes.

Toutefois, tous les systèmes actuels font face aux mêmes problèmes :
— Les stratégies de segmentation et d'extraction nécessitent l'intervention d'un expert afin de décrire les documents et entrainer le système. Ces paramétrages et entrainement sont couteux en temps et pas toujours faciles à manipuler.
— Le traitement du périmètre complet d'hétérogénéité d'informations reste difficile.
— Les solutions élaborées dépendent intrinsèquement des performances des outils de reconnaissance utilisés.

2.4 Problématiques d'évaluation des systèmes d'extraction d'informations textuelles

Comme évoqué dans le Chapitre 1.4 l'évaluation des systèmes de traitement automatique de documents présente des difficultés à cause de bases de données d'images de documents difficiles d'accès ou peu fournies, de l'existence de peu d'études comparatives ou de l'utilisation fréquente de métriques non formelles. L'utilisation de jeux de données standards est une pratique utile en traitement automatique de documents. C'est la seule manière de réaliser une comparaison correcte des algorithmes existants. Cependant la création de tels jeux de données standards peut être une tâche coûteuse en terme de sélection, d'acquisition et d'annotation des données. Les premiers travaux concernant des jeux de données standards relatifs au traitement de document date des années 1990 et se situent principalement dans le domaine de l'OCR. Ces travaux ont été motivés par le besoin "de jeux de données communs sur lesquels développer et comparer la performance des algorithmes" (Phillips *et al.*, 1993). Ces travaux ont été rapidement étendus à d'autres applications du traitement de documents telles que la binarisation, la vecto-

risation, la reconnaissance de symboles, la reconnaissance de tableaux et la vérification de signature, menant à un large nombre de jeux de données publiques (Lee & Kanungo, 2003; Paredes *et al.*, 2010; Pratikakis *et al.*, 2013). Toutefois, plusieurs problématiques sont à prendre en compte pour la conception et la création d'un jeu de données quel que soit le domaine d'application. Ces problématiques sont principalement liées à la sélection, l'acquisition et l'annotation de données et également aux différentes manières de stocker et d'organiser le jeu de données. La sélection des données, l'acquisition et l'annotation des données sont probablement les tâches les plus coûteuses en terme de temps et de ressources humaines. C'est pourquoi, un certain nombre de protocoles, de cadres, et d'outils ont été proposés pour atténuer l'effort que représentent ces tâches. Il existe par exemple des moyens efficaces d'étiqueter manuellement ou semi-automatiquement de grandes quantités de données réelles. Une alternative est la génération de données synthétiques. Dans ce cas il est nécessaire que des modèles soient définis afin de permettre de générer des données artificielles aussi proches que possible des données réelles.

2.4.1 Problématiques générales de création de jeux de données

Sélection des données

Un pré-requis pour tout jeux de données destiné à être utilisé pour l'évaluation de performance est d'être réaliste et représentatif d'un jeu d'images rencontrées dans des domaines d'applications réels.
Être réaliste implique que le jeu d'images inclus dans le jeu de données doit être le plus similaire possible aux images réelles. Évidemment il est plus facile d'atteindre cet objectif si le jeu de données contient des images directement tirées des domaines d'applications réels. Cependant, le coût d'acquisition de données réelles oblige parfois à utiliser des données synthétiques.
Être représentatif signifie que le jeu de données doit contenir un mélange équilibré de toutes les classes de documents ou d'entités qui existent dans un domaine d'application donné. La manière d'atteindre cet équilibre dépend fortement du domaine d'application. Par exemple, dans des jeux de données pour la segmentation de pages de document, l'accent doit être mis sur la collecte de pages à partir de différents types de documents incluant différentes combinaisons de mises en page, de formats de texte, de graphiques, de figures et de tableaux. Cependant, dans un domaine de recherche différent tel que la reconnaissance de mots, l'important est d'avoir un équilibre réel de tous les mots de vocabulaire ou incluant de multiples styles d'écritures manuscrites. Dans ce contexte, pour chaque domaine un ensemble de propriétés peuvent être définies et une description des données des jeux de données doit être fournie. Cela a pour conséquence l'existence de multiples jeux de données pour un domaine donné. Ainsi, dans la plupart des cas, ces multiples jeux de données peuvent être vus comme étant complémentaires afin d'obtenir un ensemble complet d'images représentatives.
La présence de bruit et de déformation est intrinsèque au traitement de documents. Ainsi, la création de jeux de données doit garantir l'inclusion d'images dégradées représentatives et réalistes, c'est à dire, des images avec un mélange équilibré de types et de niveaux de dégradation similaires à ceux rencontrés dans le monde réel. Cependant, parfois il peut être intéressant de générer des jeux de documents avec des niveaux de bruit extrêmes, plus grands que ceux des images réelles, afin de dépasser les limites de

méthodes existantes.

Acquisition/génération des données

Une fois que les données à inclure dans un jeu de données sont identifiées, l'étape suivante consiste à collecter un nombre suffisant de données. Le nombre de données suffisant est difficile à déterminer. Il va aussi dépendre du domaine d'application et des propriétés relatives aux aspects réaliste et représentatif des données. Collecter de larges ensembles de données est une tâche extrêmement chronophage. C'est pourquoi, dans la plupart des cas, les créateurs de jeux de données décident de générer des données de manière synthétique. Bien que cette pratique rende la collecte et l'annotation de données plus facile, il va à l'encontre du caractère réaliste des données. Les avantages et les inconvénients des données réelles et des données synthétiques sont discutés ci-après.

Annotation des données

Un jeu de données pour l'évaluation de performance n'est pas juste un ensemble d'images stockées dans un répertoire. Chaque image doit être étiquetée avec suffisamment d'informations pour permettre la comparaison de la sortie résultant d'un algorithme au résultat attendu pour l'image traitée. Ces informations sont communément appelées "vérité-terrain". C'est pourquoi, la conception d'un jeu de données doit inclure la définition des informations associées à chaque image et la manière dont ces informations sont représentées. La vérité-terrain est très spécifique à chaque domaine d'application, par exemple, les coordonnées d'une région rectangulaire pour la segmentation ou une chaine de caractères pour la reconnaissance de texte. Des formats de représentations de la vérité-terrain sont proposés par (Kanai *et al.*, 1995; Lee *et al.*, 1993).
L'annotation manuelle de chaque image du jeu de données peut être très fastidieuse et peut être prohibée si la vérité-terrain est complexe. Pour cette raison, des stratégies pour réduire ce coût peuvent être utilisées. La première est l'utilisation de données générées de manière synthétique. Dans ce cas, la vérité-terrain peut être obtenue automatiquement. Pour les images réelles, des stratégies existent, comme l'utilisation d'outils de développement interactif et/ou collaboratif afin d'aider la réalisation de cette tâche en proposant à l'utilisateur de confirmer ou de corriger la vérité-terrain proposée (Antonacopoulos *et al.*, 2006; Héroux *et al.*, 2007).

Choix du format de fichier

Une autre question importante concerne le format de fichiers utilisé pour stocker des images et représenter la vérité-terrain. Ces formats doivent être connus des chercheurs et faciles à manipuler. Dans le cas d'images d'entrée, généralement des formats tels que TIFF ou JPEG sont utilisés. Dans le cas de la vérité-terrain, il existe plus d'options et le format XML, par exemple, est devenu un pseudo-standard puissant (Kanai *et al.*, 1995; Margner & El Abed, 2011).

2.4.2 Avantages et inconvénients des données réelles et des données synthétiques

Le principal avantage de l'utilisation de données réelles est qu'elle permet d'évaluer des algorithmes avec le même type d'images rencontrées dans des situations réelles. Ainsi, l'évaluation peut être une très bonne estimation des performances en situation réelle. Cependant, la collecte manuelle de larges ensembles d'images réelles représente un effort conséquent qui peut être défavorable dans certains cas. De plus, l'annotation de ces images avec leur vérité-terrain correspondante est aussi très couteuse en terme de temps et de ressources humaines et des erreurs peuvent facilement s'introduire lors d'une annotation manuelle. Un autre inconvénient, dans plusieurs domaines, peut être la difficulté d'accès à un nombre suffisant d'images réelles. Parfois, des problématiques de confidentialité rendent difficile la mise à disposition de collections de données réelles en accès libre. De plus, il est difficile de quantifier le degré de bruit présent dans une image réelle. Dès lors, il n'est pas possible de définir un classement des difficultés des images selon le degré de bruit.
L'alternative aux données réelles est de développer des méthodes automatiques de génération de données synthétiques. Les principaux avantages de cette approche sont qu'elle permet de générer autant d'images que nécessaire et l'annotation des images avec la vérité-terrain est automatique. Ensuite, les efforts manuels sont réduits. De plus, les images générées en utilisant ces méthodes peuvent facilement être classées selon le type et le degré de bruit ou de dégradation appliqués, permettant ainsi d'évaluer la baisse de performance lorsque les degrés de bruit et de dégradation des images augmentent. Cependant, la difficulté principale est de réussir le développement des modèles qui permettent la génération de données aussi similaires que possible aux données réelles en prenant en compte tous les types de bruit et de déformation. Dans certains domaines, ces modèles peuvent être développés facilement, mais dans d'autres domaines, c'est une tâche vraiment difficile.

2.4.3 Mesures pour l'évaluation de performance

Les mesures les plus importantes et les plus utilisées dans le domaine de l'extraction d'informations sont le rappel (R) et la précision (P). Le rappel est défini comme la proportion d'objets correctement retrouvés parmi l'ensemble de tous les objets du jeu de données. La précision est définie comme la proportion d'objets correctement retrouvés parmi l'ensemble des objets retrouvés. Afin d'évaluer la qualité générale d'un système, une autre métrique est définie en tenant compte à la fois du rappel et de la précision. L'idée de base est de combiner les deux mesures et de calculer la moyenne de l'ensemble. La meilleure moyenne à utiliser est la moyenne harmonique appelée F-mesure et définie par :

$$FM = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P+R}. \tag{2.1}$$

D'autres mesures de performance sont décrites dans (Chen, 2015) de manière spécifique à différents domaines d'applications du traitement de documents.

2.5 Conclusion

Dans ce Chapitre, nous avons présenté les trois étapes principales qui constituent le processus d'extraction d'informations textuelles au sein d'images de documents : détection, localisation et reconnaissance. Les étapes de détection et de localisation de texte sont étroitement liées mais ne sont pas à confondre. L'objectif de ces deux étapes est de générer des régions rectangulaires autour de tous les éléments textuels présents dans des images de documents et d'identifier de manière unique chaque élément. Tandis que l'étape de détection permet d'identifier si un élément est du texte ou non, l'étape de localisation consiste à délimiter et à identifier la position de l'élément textuel détecté (lors de l'étape de détection) en l'encapsulant au sein d'une région rectangulaire. On distingue quatre types d'approches dans la littérature :

- les approches fondées sur des régions ;
- les approches fondées sur les contours ;
- les approches morphologiques ;
- les approches fondées sur les textures.

Nous avons également évoqué les caractéristiques que comportent les éléments textuels présents au sein d'images de documents : taille, géométrie, couleur, texture, etc.. Certaines de ces caractéristiques inhérentes au texte présent dans des images impactent directement le processus d'extraction. En effet, les textes peuvent subir différents changements d'apparence tels que la fonte, la taille, le style, l'orientation, l'alignement, la texture, la couleur, le contraste et l'arrière plan (Jung *et al.*, 2004). Une difficulté supplémentaire s'ajoute en ce qui concerne les documents administratifs tels que des factures. En effet, selon la mise en page adoptée, la position géographique d'une information donnée peut varier d'un émetteur à un autre. Tout ces changements compliquent et rendent plus difficile le problème d'extraction de texte.
Dans ce mémoire, nous nous intéressons particulièrement à la tâche de localisation d'informations textuelles au sein d'images de documents, tels que des factures, pour l'extraction de ces informations à l'aide d'outils de reconnaissance existants. Nous présentons dans la Partie II Chapitre 6, notre approche pour la localisation et l'extraction d'informations textuelles dans des images de documents, fondées sur les régions les contenant.
Dans les deux derniers Chapitres, nous avons pu constater l'apport important des approches de classification pour le traitement automatique d'images de documents, ainsi que pour l'extraction d'informations textuelles. Nous avons mentionnés l'utilisation des méthodes de classification telles que, les arbres de décision, les machines à vecteur support, ou encore les réseaux de neurones, dans un certain nombre d'approches. Le Chapitre suivant présente plusieurs méthodes de classification utilisées dans la littérature.

Chapitre 3

Classification

Sommaire

3.1 Classification supervisée 64
3.1.1 Arbres de décision 66
3.1.2 Réseaux bayésiens 69
3.1.3 Réseaux de neurones artificiels 72
3.1.4 Discussion 77
3.2 Classification non supervisée 79
3.2.1 Méthodes hiérarchiques 80
3.2.2 Méthodes de partitionnement 84
3.3 Conclusion 88

3.1 Classification supervisée

L'objectif de la classification supervisée est principalement de définir un modèle permettant de classer des objets (appelés aussi instances ou échantillons) dans des classes à partir d'attributs (appelés aussi variables) qualitatifs ou quantitatifs caractérisant ces objets. La classification supervisée s'appuie sur un ensemble d'entrainement à partir duquel un ensemble d'inférences est produit. Ces inférences permettent via une fonction mathématique (classifieur) de prédire la classe (valeur de sortie) d'un objet fournit en entrée du classifieur. Les données d'entrainement ou ensemble d'apprentissage sont constituées de paires entrées-sorties. Les entrées sont les objets en tant que tel et les sorties sont les classes auxquelles appartiennent les objets. Ainsi, une paire est constituée d'un objet et de sa classe.
Le processus général de classification supervisée est schématisé dans la Figure 3.1. La première étape consiste à collecter un jeu de données initial. Lors de cette étape, un expert peut être sollicité afin d'établir les caractéristiques les plus informatives. Néanmoins, des techniques d'extraction de caractéristiques et de sélection d'attributs peuvent être utilisées afin de réduire le nombre des caractéristiques. Les approches d'extraction de caractéristiques établissent une projection des caractéristiques dans un nouvel espace de plus petite dimension. Les nouvelles caractéristiques ainsi obtenues sont souvent des combinaisons des caractéristiques initiales. D'une part, les techniques d'extraction d'attributs les plus connues comprennent l'Analyse en Composantes Principales (Wold *et al.*,

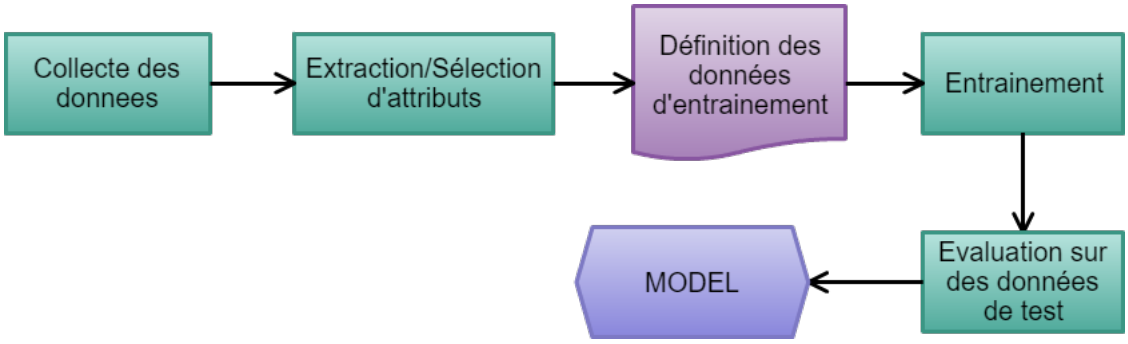


FIGURE 3.1 – Processus général d'apprentissage supervisé.

1987), l'Analyse Discriminante Linéaire (Izenman, 2013) et l'Analyse Canonique des Corrélations (Levine, 1977). D'autre part, les approches de sélection d'attributs visent à sélectionner un petit sous-ensemble de caractéristiques qui minimise la redondance et maximise la pertinence des étiquettes de classe au sein d'un ensemble de données d'entrainement. Les techniques de sélection d'attributs comprennent la mesure du gain d'information (Föllmer, 1973), l'algorithme Relief (Kira & Rendell, 1992), (Robnik-Šikonja & Kononenko, 2003), le score de Fischer (PEHRO & Stork, 2001) et Lasso (*Least absolute shrinkage and selection operator*) (Tibshirani, 1996). La mise en œuvre d'une extraction de caractéristiques et/ou d'une sélection d'attributs permet d'améliorer les performances d'apprentissage. L'étape suivante consiste à appliquer un algorithme d'apprentissage supervisé sur le jeu de données d'entrainement. Il en résulte un modèle (ou classifieur) permettant de prédire les étiquettes de classe d'échantillons de données inconnus.

La classification supervisée est une des tâches les plus fréquemment utilisées par les systèmes dits intelligents. Un nombre important de techniques a été développé à partir de techniques d'intelligence artificielle (perceptron), de techniques statistiques (réseaux bayésiens, k-plus proches voisins) et de techniques fondées sur des noyaux (plus proches voisins, machines à vecteurs supports). Les études de (Wang, 2010), (Mitra *et al.*, 2002), (Goebel & Gruenwald, 1999) et (Kotsiantis *et al.*, 2007) fournissent une vue d'ensemble de la littérature concernant les techniques de classification supervisés utilisées pour la fouille de données. Des détails sur la méthode des plus proches voisins sont disponibles dans (Weinberger *et al.*, 2006; Bhatia *et al.*, 2010). Une revue complète des machines à vecteurs supports (SVM) peut être trouvée dans (Burges, 1998). Des ouvrages plus récents sur cette technique sont celui de Cristianini et Shawe-Taylor (Cristianini & Shawe-Taylor, 2000) et celui de Scholkopf et Smola (Scholkopf & Smola, 2001). L'article de (Hsu *et al.*, 2003) peut aussi être consulté.

Dans cette section nous décrivons trois approches de classification supervisée populaires dans la littérature pour la classification de documents, la reconnaissance de motifs ou encore la fouille de données Les arbres de décision sont abordés dans la Section 3.1.1. La Section 3.1.2 présente les réseaux bayésiens et les réseaux de neurones artificiels sont présentés dans la Section 3.1.3.Une discussion autour des méthodes de classification présentées termine cette section.

3.1.1 Arbres de décision

Rokach et Maimon (Rokach & Maimon, 2014) fournissent une vue d'ensemble des premiers travaux sur les arbres de décision et de leur intérêt en classification. Les arbres de décision permettent de classer des observations en les arrangeant selon les valeurs prises par un ensemble d'attributs. Chaque nœud d'un arbre de décision représente un attribut d'une observation à classer et chaque branche partant d'un nœud représente une valeur que peut prendre l'attribut représenté par ce nœud. La Figure 3.2 est un exemple d'arbre de décision obtenu à partir des données d'entrainement du Tableau 3.1.

Soit $\mathcal{A} = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8\}$ un ensemble de données constitué de documents textuels que l'on décrit en notant l'absence (FAUX) ou présence (VRAI) de mots identifiés dans un dictionnaire $\mathcal{D} = \{m_1, m_2, m_3, m_4\}$ de quatre mots. Deux classes distinctes c_1 et c_2 sont identifiées au sein de l'ensemble de documents.

En utilisant l'arbre de décision de la Figure 3.2, une observation $<m_2$=VRAI, m_3=FAUX, m_4=VRAI> est étiquetée "c_1" en suivant successivement les valeurs prises par les nœuds représentant les attributs m_3, m_4 et enfin m_2 sur l'arbre.
Le problème de construction d'arbres de décision optimaux est un problème NP-complet et beaucoup de chercheurs se sont penchés sur la recherche d'heuristiques efficaces pour la construction de tels arbres de décision.

	m_1	m_2	m_3	m_4	classe
d_1	VRAI	FAUX	VRAI	FAUX	c_1
d_2	FAUX	VRAI	FAUX	VRAI	c_1
d_3	VRAI	VRAI	VRAI	FAUX	c_1
d_4	VRAI	FAUX	VRAI	VRAI	c_1
d_5	FAUX	VRAI	VRAI	VRAI	c_2
d_6	FAUX	VRAI	FAUX	FAUX	c_2
d_7	VRAI	FAUX	FAUX	VRAI	c_2
d_8	VRAI	VRAI	FAUX	FAUX	c_2

TABLE 3.1 – Données d'entrainement pour la construction de l'arbre de décision de la Figure 3.2.

Le nœud racine d'un arbre de décision représente l'attribut le plus discriminant, c'est à dire celui qui sépare le mieux l'ensemble de données d'entrainement. Il existe plusieurs mesures, dites d'impureté, pour déterminer l'attribut le plus discriminant. Le gain d'information (Föllmer, 1973) et l'indice de Gini (Breiman *et al.*, 1984) sont deux mesures largement utilisées dans la littérature. Tandis que certaines méthodes estiment chaque attribut indépendamment, l'algorithme Relief (Robnik-Šikonja & Kononenko, 2003) les estime dans un contexte où d'autres attributs sont aussi considérés. Néanmoins, une majorité d'études s'accordent sur le fait qu'il n'y a pas une unique bonne méthode (Murthy, 1998). Une comparaison individuelle de chaque méthode est importante dans le choix de la mesure à utiliser pour un jeu de données particulier. Une fois que le nœud racine est déterminé, la même procédure est répétée sur chaque partition obtenue par la division du jeu de données à l'étape précédente, créant ainsi des sous-arbres jusqu'à ce qu'une condition d'arrêt soit satisfaite. Une telle condition d'arrêt peut être la séparation du

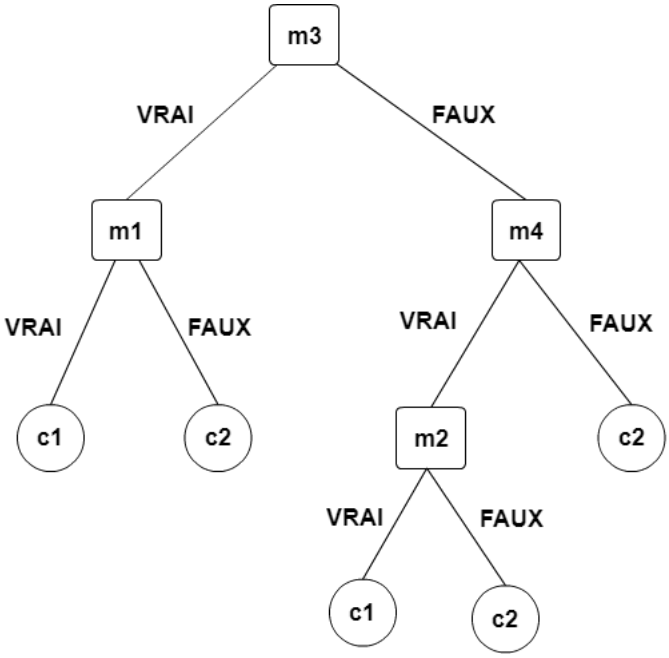


FIGURE 3.2 – Arbre de décision obtenu à partir des données d'entrainement de la Figure 3.1.

jeu de données d'apprentissage en sous-ensembles de même classe. Noter toutefois que la séparation du jeu de données d'apprentissage en sous-ensembles mono-classes expose au sur-apprentissage.
Un arbre de décision est en sur-apprentissage si son erreur de généralisation (mesurée par exemple sur un échantillon test) est "significativement" plus grande que son erreur d"apprentissage (mesurée sur l"échantillon d"apprentissage). Deux approches communes peuvent être utilisées par les algorithmes de construction d'arbres de décision pour éviter le phénomène de sur-apprentissage :

1. arrêter l'algorithme d'apprentissage avant d'atteindre le point à partir duquel celui-ci s'accorde parfaitement avec les données d'apprentissage ;
2. élaguer l'arbre de décision induit, en utilisant le principe du rasoir d'Occam. Si deux arbres utilisent le même type de tests et ont le même taux de prédiction, celui avec le moins de feuilles est préféré. Breslow et Aha (Breslow & Aha, 1997) ont réalisé une étude des méthodes de simplification d'arbres permettant d'améliorer leur compréhensibilité.

Le moyen le plus simple d'éviter le sur-apprentissage est de pré-élaguer l'arbre de décision en ne le laissant pas croître jusqu'à sa taille maximale. Un moyen est de tester la qualité d'un attribut via un critère non trivial de terminaison (un seuil fixé par exemple). Les classifieurs à base d'arbres de décision utilisent généralement des techniques de post-élagage évaluant la performance des arbres élagués sur un ensemble de données d'évaluation. Une étude comparative des méthodes d'élagage les plus connues est présentée dans (Elomaa, 1999). Elomaa conclut qu'il n'existe pas une méthode d'élagage meilleure que toutes les autres. Plus d'informations sur les techniques de pré-élagage et de post-élagage des arbres de décision peuvent être trouvées dans (Bruha, 2000) et (Rokach & Maimon, 2014).
Les arbres de décision sont généralement uni-variés dès lors qu'ils utilisent des sépara-

tions des données fondées sur un seul attribut au niveau de chaque nœud. Beaucoup d'algorithmes d'arbre de décision peuvent ne pas fonctionner correctement sur des problèmes nécessitant un partitionnement sur plusieurs attributs. Néanmoins, il existe quelques méthodes pour la construction d'arbres multi-variés. Par exemple, Zheng (Zheng, 1998) propose un algorithme qui améliore la précision de classification des arbres de décision en construisant de nouveaux attributs binaires à l'aide d'opérateurs logique tels que la conjonction, la négation et la disjonction. De plus, Zheng (Zheng, 2000) créa ce qu'il appelle les attributs X-de-N (originalement nommés X-*of*-N). Pour une observation donnée, la valeur d'un attribut X-de-N est "vrai" si au moins X parmi N conditions sont vérifiées pour cette observation, " "faux" sinon. Gama et Bradzil (Gama & Brazdil, 1999) ont combiné un arbre de décision et un discriminant linéaire pour la création d'arbres multi-variés. Dans leur modèle, les nouveaux attributs sont le résultat de combinaisons linéaires des attributs précédents.
L'algorithme le plus connu de la littérature pour la construction d'arbres de décision est l'algorithme C4.5 (Quinlan, 1993). L'algorithme C4.5 est une extension du premier algorithme de Quinlan, l'algorithme ID3 (Quinlan, 1979). A notre connaissance, une des dernières études comparant les arbres de décision et d'autres algorithmes d'apprentissage a été réalisée par (Lim *et al.*, 2000). L'étude montre que l'algorithme C4.5 a de très bons résultats de taux d'erreur et de temps de calcul. Ruggieri (Ruggieri, 2002) présente une étude analytique du temps d'exécution de l'algorithme C4.5 qui met en lumière plusieurs améliorations possibles. A partir de son étude, il propose une implémentation plus efficace de l'algorithme C4.5 : l'algorithme EC4.5. Il argumente que son algorithme est capable de calculer les mêmes arbres de décision que l'algorithme C4.5 avec cinq fois plus de performance.
L'algorithme C4.5 suppose que les données d'entrainement tiennent dans la mémoire disponible. En partant de ce constat Gehrke et al. (Gehrke *et al.*, 2000) proposent *Rainforest*, un cadre pour le développement d'algorithmes rapides et évolutifs pour la construction d'arbres de décision et qui s'adaptent bien à la taille de la mémoire disponible. Par ailleurs, Olcay et Onur (Yıldız & Dikmen, 2007) présentent trois manières de paralléliser l'algorithme C4.5 :

1. à partir des attributs ;

2. à partir des nœuds ;

3. à partir des données.

Baik et Bala (Baik & Bala, 2004) ont quant à eux présenté un travail préliminaire sur une approche fondée sur des agents pour la construction distribuée d'arbres de décision.
En résumé, une des caractéristiques les plus intéressantes des arbres de décision est leur facilité de compréhension. Les utilisateurs peuvent facilement comprendre pourquoi un arbre de décision classe une certaine observation dans une certaine classe. En effet, un arbre de décision consiste en une hiérarchie de tests par lesquels passe une instance inconnue afin d'en déterminer la classe. L'hypothèse faite par les arbres de décision est que les instances appartenant à des classes différentes possèdent au moins une valeur d'attribut différente. Les arbres de décision sont meilleurs pour le traitement d'attributs à valeurs discrètes ou catégorielles.

3.1.2 Réseaux bayésiens

Les réseaux bayésiens ont été initiés par Judea Pearl dans les années 1980 (Pearl, 1982, 1984, 1986, 1988). Ce sont des outils très pratiques pour la représentations de connaissances incertaines et le raisonnement à partir d'informations incomplètes. Un réseau bayésien $\mathcal{B} = (\mathcal{G}, \Theta)$ est défini par :

- $\mathcal{G} = (X, E)$, un graphe dirigé sans circuit (ou graphe orienté acyclique) dont les sommets sont associés à un ensemble de variables aléatoires $X = \{X_1, ..., X_n\}$,
- $\Theta = \{P(X_i|Pa(X_i))\}$, ensemble des probabilités de chaque nœud X_i conditionnellement à l'état de ses parents $Pa(X_i)$ dans $\mathcal{G}$.

La représentation graphique du réseau bayésien indique les dépendances (ou indépendances) entre les variables et donne un outil visuel de représentation des connaissances facilement appréhendable par des utilisateurs. La représentation graphique d'un réseau bayésien est un graphe orienté acyclique dont les nœuds sont un à un en correspondance avec l'ensemble X de variables. Chaque nœud est conditionnellement dépendant (ou indépendant) de ses non-descendants, étant donné l'état de ses parents immédiats. Pearl et Verma (Pearl & Verma, 1995) ont aussi montré que les réseaux bayésiens permettaient de représenter de manière compacte la distribution de probabilité jointe sur l'ensemble des variables :

$$P(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P(X_i|Pa(X_i)). \tag{3.1}$$

Cette décomposition d'une fonction globale en un produit de termes locaux dépendant uniquement du nœud considéré et de ses parents dans le graphe, est une propriété fondamentale des réseaux bayésiens. Elle est à la base des premiers travaux portant sur le développement d'algorithmes d'inférence, qui calculent la probabilité de n'importe quelle variable du modèle à partir de l'observation partielle des autres variables. Ce problème a été prouvé NP-complet, mais a abouti à différents algorithmes qui peuvent être assimilés à des méthodes de propagation d'information dans un graphe. Ces méthodes utilisent la notion de probabilité conditionnelle, c'est à dire quelle est la probabilité de X_i sachant que X_j a été observé, mais aussi le théorème de Bayes (Théorème 1), qui permet de calculer la probabilité de X_j sachant X_i, lorsque $P(X_i|X_j)$ est connu. Une introduction à ces méthodes est disponible dans (Bishop, 2006) et (Bielza & Larrañaga, 2014).

Théorème 1. *Considérons des évènements A_1,...,A_n tels qu'ils forment une partition de l'ensemble fondamental E. Par définition, les A_i s'excluent mutuellement et leur union est E.*

$$\forall(i), (A_i \cap A_j = \phi);$$
$$\bigcup_{i=1}^{n} A_i = E.$$

Soit B un évènement quelconque. On tire $B = B \cap (A_1 \cup A_2 \cup ... \cup A_n)$. En remarquant que les $B \cup A_i$ sont exclusifs, puisque les A_i le sont, on obtient la formule dite des probabilités totales :

$$P(B) = P(B \cap A_1) + P(B \cap A_2) + ... + P(B \cap A_n).$$

En appliquant le théorème des probabilités conditionnelles on a :

$$P(B \cap A_i) = P(A).P(A_i|B) = P(A_i).P(B|A_i)$$

donc : $P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B|A_1)P(A_1)+P(B|A_2)P(A_2)+...+P(B|A_n)P(A_n)}$.

Les classifieurs fondés sur des réseaux bayésiens sont des types particuliers de réseaux construits pour des problèmes de classification. Pour rappel, en classification supervisée le modèle utilisé pour la classification doit être appris à partir d'un ensemble de données étiquetées. Considérons l'exemple suivant.

Soient trois classes de documents $\mathcal{C} = c_1$,c_2,c_3, un ensemble d'apprentissage $\mathcal{A} = \{d_1, ..., d_i, ..., d_n\}$ de documents textuels et un dictionnaire de mots $\mathcal{M} = \{m_1 m_2$,m_3,$m_4\}$ où les m_i sont supposés indépendants.
Soit $d_i = (1, 1, 0, 1)$ un document à classer. La représentation d'un document d_i utilise le modèle vectoriel caractérisé par l'absence (0) ou la présence (1) des mots du dictionnaire au sein du document. L'apprentissage sur $\mathcal{A}$ donne les paramètres présentés dans le Tableau 3.2.

c_1	c_2	c_3
0,300	0,200	0,150

(a) P(c_i)

	c_1	c_2	c_3
m_1	0,200	0,200	0,101
m_2	0,003	0,005	0,673
m_3	0,003	0,005	0,215
m_4	0,538	0,522	0,005

(b) P(m_i=1|c_j)

TABLE 3.2 – Paramètres d'apprentissage du réseau.

De $P(m_i = 1|c_j)$ est déduit $P(m_i = 0|c_j)$. La classe de d_i est donnée par :

$$\underset{c}{^{argmax}} \left[P(c_i|d_i) = \frac{P(m_i|c_i) \cdot P(c_i)}{P(m_i)}\right] \tag{3.2}$$

Il n'est pas nécessaire de calculer $P(m_i)$ qui est constant d'après l'hypothèse d'indépendance des m_i. Calculons $P(m_i|c_i) \cdot P(c_i)$:

$$\begin{aligned} P(m_i|c_1) \cdot P(c_1) &= \prod_{j=1}^{4}\Big(P(m_{ij}|c_1)\Big) \cdot P(c_1) \\ &= P(m_{i1}|c_1) \cdot P(m_{i2}|c_1) \cdot P(m_{i3}|c_1) \cdot P(m_{i4}|c_1) \cdot P(c_1) \\ &= 0,200 \cdot 0,003 \cdot (1 - 0,003) \cdot 0,538 \cdot 0,300 \\ &= 9,65.10^{-4} \end{aligned} \tag{3.3}$$

$$\begin{aligned} P(m_i|c_2) \cdot P(c_2) &= 1,038.10^{-3} \\ P(m_i|c_3) \cdot P(c_3) &= 4,04.10^{-4} \end{aligned}$$

On en déduit que d_i est de classe c_2.

La tâche d'apprentissage d'un réseau bayésien peut être divisée en deux sous-tâches :
— l'apprentissage de la structure de graphe orienté acyclique du réseau ;
— l'apprentissage de ses paramètres.

Le problème d'apprentissage des paramètres d'un réseau bayésien consiste à estimer les distributions des probabilités conditionnelles à partir d'un ensemble de données

d'apprentissage. Deux types d'approches existent selon la nature des données d'apprentissage : l'apprentissage de paramètres à partir d'un jeu de données complet et l'apprentissage de paramètres à partir d'un jeu de données incomplet.
Dans le premier type d'approches, toutes les variables ont été observées pour chacune des données d'apprentissage. Dans ce cas, l'apprentissage des paramètres est simplifié. En effet, l'apprentissage consiste alors à (a) estimer les probabilités conditionnelles en fonction de la fréquence d'apparition des évènements dans le jeu de données ; ou (b) à considérer une distribution *a priori* des paramètres.
Dans le deuxième type d'approches l'apprentissage est réalisé à partir d'un jeu de données incomplet. Dans ce cas, la démarche d'apprentissage des paramètres dépend de la nature plus ou moins aléatoires des données manquantes. Elles peuvent être totalement aléatoires (ne dépendent pas de la base de données), pseudo-aléatoires (dépendent des données observées) ou non-aléatoires. Dans les deux premiers cas, l'avantage est de pouvoir estimer une distribution des données manquantes ; dans le dernier cas, il faut disposer d'informations supplémentaires.
Plusieurs approches ont été proposées pour cet apprentissage (Jaakkola & Jordan, 2000; Gelman *et al.*, 2014; Zhou *et al.*, 2014), la plus connue étant basée sur l'algorithme EM (Espérance-Maximisation) (Dempster *et al.*, 1977; Friedman, 1998; Moon, 1996) permettant l'estimation de la log-vraisemblance des valeurs manquantes. L'algorithme EM est basé sur la répétition de deux étapes jusqu'à convergence du maximum de vraisemblance, après avoir initialisé les valeurs manquantes :

1. étape E, "espérance" : estimation des valeurs manquantes en calculant leur espérance selon les paramètres du modèle (ces valeurs sont initialisées aléatoirement lors de la première itération) ;
2. étape M, "maximisation" : estimation des paramètres par maximum de vraisemblance (de la même façon qu'avec des données complètes).

L'apprentissage de la structure d'un réseau bayésien s'intéresse à trouver la structure qui représentera le mieux un problème donné. Il existe deux grandes catégories de méthodes d'apprentissage de la structure d'un réseau : les approches dites sous contrainte basées sur des tests d'indépendance et les approches basées sur le calcul d'un score.
Les approches sous contrainte consiste à tester l'indépendance entre toutes les paires de variables du jeu de données. A partir d'un graphe complet, une arête non-dirigée entre deux variables testées est supprimée si le test les déclare significativement indépendantes, puis des tests d'indépendances entre deux variables, conditionnés à un ensemble de variables tierces sont réalisés avant de modifier le graphe. L'indépendance conditionnelle est testée en appliquant le principe de la *d-separation* (Geiger *et al.*, 2013). Les tests d'indépendances non-conditionnés utilisés sont généralement basés sur la statistique du χ^2. Plus d'informations sur ce type d'approches sont disponibles dans (Feelders & Van der Gaag, 2006; Niculescu *et al.*, 2006; Campos & Ji, 2011).
Les approches basées sur le calcul d'un score évalue un réseau en lui associant un score (une constante) généralement basé sur la vraisemblance du graphe face aux observations. Une fonction de calcul de score est donc nécessaire. Le calcul du score peut être soit basé sur le calcul d'une vraisemblance, soit c'est une probabilité a posteriori dans le cadre bayésien. Les scores AIC (Akaike, 1992), BIC (Bayesian Information Criteria) (Schwarz *et al.*, 1978), BD (Bayesian Dirichlet) et MDL (Minimum Description Length) (Rissanen, 1978; Schwarz *et al.*, 1978; Rissanen, 1998) sont des exemples de fonctions de score. Plus d'informations sur les fonctions de score sont disponibles dans (Naïm *et al.*, 2011).

Pour apprendre la structure d'un réseau à partir d'un ensemble de données, le score ne suffit pas. En effet, le calcul du score se fait à partir d'un graphe et il faudrait donc calculer les scores de tous les graphes possibles afin de sélectionner le graphe ayant le score le plus élevé. Dans la pratique le calcul de tous ces scores n'est pas possible. Une solution est de parcourir l'espace de toutes les structures de graphes possibles avec une stratégie de recherche performante. Dans la littérature il existe plusieurs approches telles que la stratégie de recherche gloutonne (Friedman *et al.*, 1997; Chickering, 2002) et la stratégie de recuit simulé (Brown & Huntley, 1992).
Une comparaison des méthodes fondées sur des scores et des méthodes fondées sur des indépendances conditionnelles sont présentées dans (Heckerman *et al.*, 2006). Les deux types d'approches ont chacune leurs avantages et leurs inconvénients. De manière générale, l'approche consistant à analyser des dépendances est plus efficace que l'approche fondée sur des scores pour des réseaux fortement connectés. Il est alors possible de déduire la structure correcte du réseau lorsque la distribution des probabilités des données satisfont certaines hypothèses. Cependant, beaucoup de ces algorithmes requièrent un nombre exponentiel de tests. Mais bien que l'approche de recherche de réseaux fondée sur des scores peut ne pas trouver la meilleure structure en raison de sa nature heuristique, il fonctionne avec une plus large gamme de modèles de probabilités que l'approche fondée sur l'analyse de dépendance. Madden (Madden, 2003) présente une étude comparative de la performance d'un certain nombre de réseaux Bayésien. Ses expérimentations montrent que des performances de classification très similaires peuvent être obtenues par des classifieurs construits en utilisant les différentes approches décrites ci-dessus.
Les classifieurs bayésiens ont plusieurs avantages :

- Ils offrent une représentation graphique explicite et interprétable par des utilisateurs.
- Comme ils produisent un modèle probabiliste, ce modèle peut être utilisé dans un cadre décisionnel dans des problèmes pour lesquels des probabilités conditionnelles peuvent être utilisées comme mesure de confiance d'une classe prédite.
- Leurs algorithmes sont faciles à implémenter.
- Ils peuvent être construits avec des algorithmes efficaces en temps de calcul dont la complexité de temps d'apprentissage est linéaire sur le nombre d'instances et linéaire, quadratique ou cubique sur le nombre de variables.
- Ils sont applicables à beaucoup de situations réelles (Khakzad, 2015; Alonso-Montesinos *et al.*, 2016; Constantinou *et al.*, 2016).

Néanmoins, un problème concernant les classifieurs fondés sur les réseaux bayésiens sont qu'ils ne sont pas adaptés pour les jeux de données avec beaucoup de variables (Cheng *et al.*, 2001). Cela est principalement dû à l'impossibilité de construire un réseau très large en terme de temps et d'espace. Un autre problème est que dans la plupart des cas, les variables numériques doivent être discrétisées avant la production du réseau.

3.1.3 Réseaux de neurones artificiels

Les réseaux de neurones font partie des approches d'Intelligence Artificielle qui ont pour objectif de simuler des comportements du cerveau humain. En 1943 Mc Culloch et Pitts (McCulloch & Pitts, 1943) ont proposé les premières notions de *neurone formel*. Ce concept fut ensuite mis en réseau avec une couche d'entrée et une sortie par Rosen-

blatt en 1957 (Rosenblatt, 1957) pour simuler le fonctionnement rétinien et tacher de reconnaitre des formes. C'est l'origine du *perceptron*.
Nous nous intéressons dans cette section qu'à une structure élémentaire de réseau, celle dite statique, ne présentant pas de boucle de rétroaction et restant dans le contexte d'apprentissage supervisé. Les systèmes dynamiques avec boucle de rétroaction tels que les réseaux dit "cartes de Kohonen" ou carte auto-organisatrices, qui sont en fait des algorithmes de classification non-supervisées, ne sont pas abordés. Le lecteur peut se référer à (Kohonen, 2013) pour plus d'informations sur ces systèmes.
Dans cette section nous définissons et décrivons les caractéristiques des réseaux de neurones ou perceptrons multicouches spécifiques pour la classification supervisée.

Structure d'un réseau de neurones

Un *réseau neuronal* est l'association, en un graphe plus ou moins complexe, d'objets élémentaires, les *neurones formels*. Les principaux réseaux se distinguent par l'organisation du graphe (en couches ou complets par exemple), c'est-à-dire leur architecture, son niveau de complexité (le nombre de neurones, présence ou non de boucles de rétroaction dans le réseau), par le type des neurones (leurs fonctions de transition ou d'activation) et enfin par l'objectif visé : apprentissage supervisé (ou non supervisé), optimisation, systèmes dynamiques.
De façon très réductrice, un neurone biologique est une cellule qui se caractérise par :

- des synapses, les points de connexion avec les autres neurones, fibres nerveuses ou musculaires ;
- des dendrites, les "entrées" du neurones ;
- l'axone, la "sortie" du neurone vers d'autres neurones ou fibres musculaires ;
- le noyau qui active la sortie en fonction des stimulations en entrée. Par analogie, le neurone formel est un modèle qui se caractérise par un état interne $s \in S$, des signaux d'entrée $x_1, ..., x_p$ et une fonction d'activation.

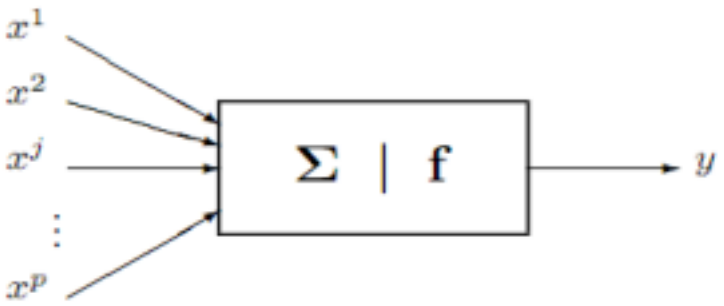


FIGURE 3.3 – Représentation d'un neurone formel.

La fonction d'activation opère une transformation d'une combinaison affine des signaux d'entrée, α_0 étant appelé le biais du neurone. Cette combinaison affine est déterminée par un *vecteur de poids* $[\alpha_0, ...\alpha_p]$ associé à chaque neurone et dont les valeurs sont estimées dans la phase d'apprentissage. Ils constituent "la mémoire" ou "connaissance répartie" du réseau. La Figure 3.3 est une représentation d'un neurone formel.

Les différents types de neurones se distinguent par la nature f de leur fonction d'activation. Les principaux types de fonction sont :

- *linéaire* $f(x) = x$ (la fonction identité),
- *sigmoïde* $f(x) = 1/(1 + e^x)$,
- *seuil* $f(x) = 1_{[0,+\infty[}(x)$,
- *radiale* $f(x) = \sqrt{1/2\pi} exp(-x^2/2)$,
- *stochastiques* $f(x) = 1$ avec la probabilité $1/(1 + e^{-x/H})$, 0 sinon (H intervient comme une température dans un algorithme de recuit simulé (Brown & Huntley, 1992)).

Les modèles linéaires et sigmoïdaux, les plus utilisés, sont bien adaptés aux algorithmes d'apprentissage impliquant une rétro-propagation du gradient car leur fonction d'activation est différentiable. Le modèle à seuil est sans doute plus conforme à la réalité biologique mais pose des problèmes d'apprentissage. Enfin le modèle stochastique est utilisé pour des problèmes d'optimisation globale de fonctions perturbées ou encore pour les analogies avec les systèmes de particules. On ne le rencontre par en fouille de données.

Perceptron multicouche

Le perceptron multicouche (PMC) est un réseau composé de couches successives. Une couche est un ensemble de neurones n'ayant pas de connexion entre eux. Le réseau est constitué :

- d'une couche d'entrée qui lit les signaux entrant et qui est constituée d'un neurone par entrée x_j, une couche en sortie fournit la réponse du système ; selon les auteurs, la couche d'entrée qui n'introduit aucune modification n'est pas comptabilisée ;
- d'une ou plusieurs couches cachées participent au transfert ; un neurone d'une couche cachée est connectée en entrée à chacun des neurones de la couche précédente et en sortie chaque neurone de la couche suivante ;
- d'une fonction de transfert.

Étant donné des entrées $X_1, .., X_p$, des paramètres de poids d'entrée α, β à estimer et une variable de sortie Y à expliquer, un perceptron multicouche réalise une transformation des variables d'entrée :

$$Y = \phi(X_1, ..., X_p; \alpha), \tag{3.4}$$

où α est le vecteur contenant chacun des paramètres α_{jkl} de la j-ème entrée du k-ème neurone de la l-ème couche ; la couche d'entrée ($l = 0$) n'est pas paramétrée, elle ne fait que distribuer les entrées sur tous les neurones de la couche suivante.
Un théorème dit d'*approximation universelle* montre que cette structure élémentaire à une seule couche cachée est bien suffisante pour prendre en compte les problèmes classiques de modélisation ou d'apprentissage statistique. En effet, toute fonction régulière peut être approchée uniformément avec une précision arbitraire et dans un domaine fini de l'espace des variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini possédant tous la même fonction d'activation et un neurone de sortie linéaire. La Figure 3.4 montre un exemple de perceptron multicouche élémentaire avec une couche cachée et une couche de sortie.

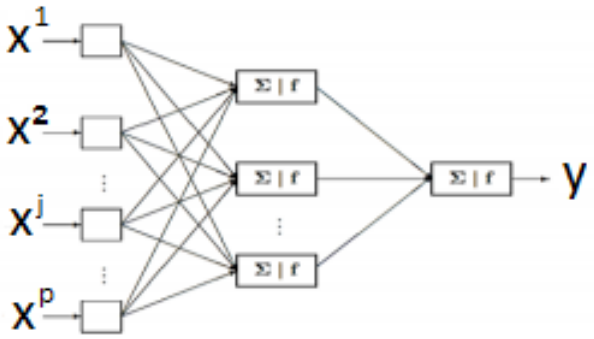


FIGURE 3.4 – Exemple de perceptron multicouche élémentaire avec une couche cachée et une couche de sortie.

De façon usuelle et en régression, la dernière couche est constituée d'un seul neurone muni de la fonction d'activation identité tandis que les autres neurones (couche cachée) sont munis de la fonction sigmoïde. En classification binaire, le neurone de sortie est muni également de la fonction sigmoïde tandis que dans le cas d'une discrimination à m classes, ce sont m neurones avec fonctions sigmoïde, un par classe, qui sont considérés en sortie.
Soient une base d'apprentissage constituée de n observations $(x_i^1, ..., x_i^p; y_i)$, des variables explicatives $X^1, ..., X^p$ et une variable Y à prévoir. Considérons le cas le plus simple de la régression avec un réseau constitué d'un neurone de sortie linéaire et d'une couche à q neurones dont les paramètres sont optimisés par moindres carrés. L'apprentissage est l'estimation des paramètres $\alpha_{j=0;p;k=1}$ et $\beta_{k=0,q}$ par minimisation de la fonction perte quadratique (ou d'une fonction d'entropie en classification) :

$$\mathcal{Q}(\alpha, \beta) = \sum_{i=1}^{n} \mathcal{Q}_i = \sum_{i=1}^{n} [y_i - \phi(x; \alpha, \beta)]^2. \tag{3.5}$$

Différents algorithmes d'optimisation sont proposés, ils sont généralement basés sur une évaluation du gradient par rétro-propagation (Rumelhart *et al.*, 1985; Leonard & Kramer, 1990; Specht, 1991). L'apprentissage consiste donc à évaluer la dérivée de la fonction coût en une observation et par rapport aux différents paramètres. Soit $z_{ki} = f(\alpha_{k0} + \alpha_{k'x_i})$ et $z_i = \{z_{1i}, ..., z_{qi}\}$. Les dérivées partielles de la fonction perte quadratique s'écrivent :

$$\begin{aligned} \frac{\partial \mathcal{Q}_i}{\partial \beta_k} &= -2(y_i - \phi(x_i))(\beta' z_i) z_{ki} = \delta_i z_{ki} \\ \frac{\partial \mathcal{Q}_i}{\partial \alpha_{kj}} &= -2(y_i - \phi(x_i))(\beta' z_i)\beta_k f'(\alpha_k' x_i) x_{ip} = s_{ki} x_{ip}. \end{aligned} \tag{3.6}$$

Les termes δ_i et s_{ki} sont respectivement les termes d'erreur du modèle courant à la sortie et sur chaque neurone caché. Ces termes d'erreur vérifient les équations dites de rétro-propagation :

$$s_{ki} = f'(\alpha_k' x_i)\beta_k \delta_i, \tag{3.7}$$

dont les termes sont évalués en deux passes. Une passe "avant", avec les valeurs courantes des poids, l'application des différentes entrées x_i au réseau permet de déterminer les valeurs ajustées $\hat{\phi}(x_i)$. Ensuite une passe "retour" permet de déterminer les δ_i qui sont

"rétro-propagés" afin de calculer les s_{ki} et ainsi obtenir les évaluations des gradients.
Sachant évaluer les gradients, différents algorithmes, plus ou moins sophistiqués, ont été implémentés (Narendra & Parthasarathy, 1991; Masters, 1995; Ren *et al.*, 2014). Le plus élémentaire est une utilisation itérative du gradient : en tout point de l'espace de paramètres, le vecteur gradient de $\mathcal{Q}$ pointe dans la direction de l'erreur croissante. Pour faire décroître $\mathcal{Q}$ il suffit donc de se déplacer en sens contraire. Cet algorithme est un algorithme itératif modifiant les poids de chaque neurone selon :

$$\begin{aligned} \beta_k^{r+1} &= \beta_k^{(r)} - \tau \sum_{i=1}^{n} \frac{\partial \mathcal{Q}_i}{\partial \beta_k^{(r)}} \\ \alpha_{kp}^{r+1} &= \alpha_{kp}^{(r)} - \tau \sum_{i=1}^{n} \frac{\partial \mathcal{Q}_i}{\partial \alpha_{kp}^{(r)}}. \end{aligned} \tag{3.8}$$

Le coefficient de proportionnalité τ est appelé le *taux d'apprentissage*. Il peut être fixe, à déterminer par l'utilisateur, ou encore varier en cours d'exécution selon certaines heuristiques. Il paraît en effet intuitivement raisonnable que ce taux soit grand au début pour aller plus vite, puis que ce taux décroisse pour aboutir à un réglage plus fin au fur et à mesure que le système s'approche d'une solution.
D'autres méthodes d'optimisation ont été adaptées à l'apprentissage d'un réseau : méthodes du gradient avec second ordre utilisant une approximation itérative de la matrice hessienne (algorithmes BFGS (Battiti & Masulli, 1990) et de Levenberg-Marquardt (Hagan & Menhaj, 1994)) ou encore une évaluation implicite de cette matrice par la méthode dite du gradient conjugué (Johansson *et al.*, 1991; Caruana *et al.*, 2001). Plus de références sont disponibles dans (Haykin, 2001) et (Demuth *et al.*, 2014).

Considérons l'exemple suivant. Soit une base d'apprentissage $\mathcal{A}$ de documents textuels que l'on décrit en notant l'absence (-1) ou présence (1) de mots identifiés dans un dictionnaire $\mathcal{D}$ de 4 mots. La base d'apprentissage est composée de deux classes distinctes, c_1 et c_2, de documents.
Un réseau de neurones de type perceptron mono-couche est utilisé. La valeur du seuil est fixée à 3.0. La fonction de combinaison est une fonction qui calcule la somme pondérée des entrées, et la fonction d'activation est une fonction de comparaison au seuil (si la somme pondérée est supérieur au seuil alors la sortie vaut 1 et l'élément appartient à la classe c_1, à c_2 sinon). On souhaite, à présent, déterminer la classe d'un document $d = (1, -1, 1, -1, -1)$. Pour ce faire, il faut exécuter le réseau de neurones avec en entrée les valeurs de d pondérées par le vecteur de poids $w = (1.5, -3.0, 0.7, 1.3, -0.6)$; On a donc que d est de classe c_1.
De manière générale, lorsque le nombre de classes est supérieur à 2, le problème de classification est ramené à une classification binaire en adoptant par exemple, la stratégie un contre tous. Néanmoins, les réseaux de neurones suffisamment complexe peuvent en sortie gérer les classes multiples.
Les d'applications des perceptrons multicouche sont très nombreux : discrimination, prévision d'une série temporelle, reconnaissance de forme, etc. Les principales difficultés rencontrées avec les perceptrons multicouche sont liées à l'apprentissage (temps de calcul, taille de la base d'apprentissage, localité de l'optimum obtenu) ainsi que son statut de boîte noir. En effet contrairement à un modèle de discrimination ou un arbre, il est a priori impossible de connaître l'influence effective d'une entrée sur le système dès qu'une couche cachée intervient. En revanche, ils possèdent d'indéniables qualités

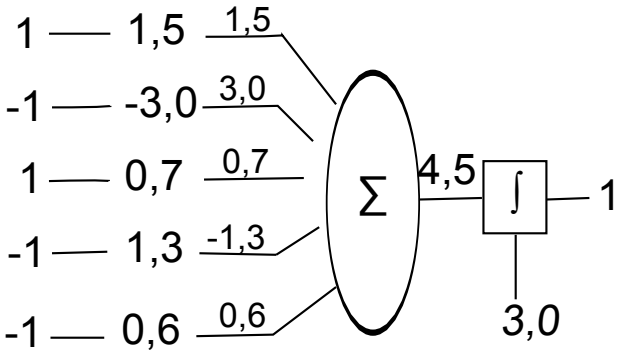


FIGURE 3.5 – Exemple de réseau de neurones pour la classification d'un ensemble de documents.

lorsque l'absence de linéarité et/ou le nombre de variables explicatives rendent les modèles statistiques traditionnels inutilisables.

3.1.4 Discussion

Le Tableau 3.3, synthétise les forces et les faiblesses des techniques de classification évoquées dans cette section. Il est extrait de l'article de (Kotsiantis *et al.*, 2007) sur les techniques majeures de classification (arbres de décision, réseaux de neurones, réseaux bayésiens, kNN). Ces techniques sont comparées selon plusieurs propriétés, d'après des études empiriques et théoriques existantes :

- leur performance générale ;
- leur rapidité d'apprentissage par rapport au nombre d'attributs et au nombre d'instances ;
- leur rapidité de classification ;
- leur tolérance aux valeurs manquantes ;
- leur tolérance aux attributs non pertinents ;
- leur tolérance aux attributs fortement dépendant ;
- leur capacité à traiter des attributs à valeurs discrètes, binaires ou continues ;
- leur tolérance au bruit ;
- leur capacité à prendre en compte le risque de sur-apprentissage ;
- leur capacité à réaliser un apprentissage incrémental ;
- leur facilité d'interprétation du résultat de la classification ;
- la facilité de manipulation des paramètres du modèle.

De manière générale, les SVMs et les réseaux de neurones artificiels obtiennent les meilleurs performances et donnent de meilleurs résultats pour le traitement de nombreux attributs à valeurs continues. Par ailleurs, les arbres de décision tendent à donner de meilleurs résultats pour le traitement de variables discrètes ou catégorielles. Les modèles fondés sur des réseaux de neurones ou des SVMs sont les moins rapides en phase d'apprentissage car de grandes tailles d'échantillons sont requises pour atteindre une bonne précision de précision. A l'inverse, les modèles bayésiens nécessitent un jeu de données d'entrainement relativement petit et sont donc plus rapides en phase d'apprentissage. De plus, l'approche bayésienne nécessite peu d'espace de stockage pendant les

	Arbres de décision	Réseaux de neurones	Bayes naïf	kNN	SVM
Performance générale	**	***	*	**	****
Rapidité d'apprentissage par rapport au nombre d'attributs et au nombre d'instances	***	*	****	****	*
Rapidité de classification	****	****	****	*	****
Tolérance aux valeurs manquantes	***	*	****	*	**
Tolérance aux attributs non pertinents	***	*	**	**	****
Tolérance aux attributs redondants	**	**	*	**	***
Tolérance aux attributs fortement interdépendant	**	***	*	*	***
Traitement d'attributs à valeurs discrètes/binaires/continues	****	***	***	***	**
Tolérance au bruit	**	**	***	*	**
Prise en compte du risque de sur-apprentissage	**	*	***	***	**
Capacité à réaliser un apprentissage incrémental	**	***	****	****	**
Facilité d'interprétation de la classification	****	*	****	**	*
Manipulation des paramètres du modèle	***	*	****	***	*

TABLE 3.3 – Tableau Comparatif d'algorithmes de classification supervisée extrait de l'étude de (Kotsiantis *et al.*, 2007) : **** 4 étoiles représentent la meilleure performance et * une étoile représente la moins bonne performance.

étapes d'entrainement et de classification : la mémoire minimum nécessaire est utilisée pour le stockage des probabilités conditionnelles. Par ailleurs, les modèles bayésiens sont naturellement robuste aux valeurs manquantes car celles-ci sont tout simplement ignorées dans le calcul des probabilités et par conséquent n'ont pas d'impact sur la décision finale. A l'inverse, les réseaux de neurones complètent les données manquantes pour fonctionner, ce qui peut apporter un biais lors de la phase d'entrainement. Aussi, la présence d'attributs non pertinents peut rendre l'entrainement des réseaux de neurones inefficace, voir inexploitable. De la même manière, la littérature s'accorde sur le fait que l'approche k-plus proches voisins (kNN) est très sensible aux attributs non pertinents. Cette particularité peut être expliquée par le principe de fonctionnement même de cette approche. Le nombre de paramètres de modèle ou d'exécution à déterminer par l'utilisateur est un indicateur de la facilité d'usage d'un algorithme. Sans surprise, les réseaux de neurones et les SVMs ont plus de paramètres que les autres techniques. En revanche, pour le modèle kNN le seul paramètre à déterminer est le paramètre k, lequel

est relativement facile à déterminer.
Finalement, aucun algorithme d'apprentissage ne peut à lui seul surpasser tous les autres sur tout jeux de données. Pour déterminer quel algorithme d'apprentissage est le plus efficace pour un problème donné, l'approche la plus simple est d'estimer les performances d'algorithmes candidats sur ce problème, puis de sélectionner celui qui semble le plus performant. Combiner des classifieurs est une méthode de plus en plus utilisée pour améliorer les performances de chaque classifieur individuel. De nombreuses méthodes sont proposées pour la création d'ensemble de classifieurs, notamment par (Dietterich, 2000) pour la construction d'ensembles d'arbres de décision par exemple. L'étude de méthodes pour la construction d'ensembles de classifieurs satisfaisants est ainsi un axe de recherche très actif dans le domaine de l'apprentissage supervisé. Les techniques de classification supervisées d'apprentissage sont applicables dans de nombreux domaines. Saitta et Neri (Saitta & Neri, 1998) et Witten et Frank (Witten *et al.*, 2016) présentent un certain nombre d'articles sur leurs applications.

3.2 Classification non supervisée

La classification numérique (non supervisée) ou, en anglais, *clustering*, est une technique de fouille de données dont l'objet est de révéler, dans un ensemble d'objets, des regroupements homogènes et isolés en ce sens que les objets sont fortement similaires au sein d'un regroupement et faiblement similaires entre deux regroupements. Cela nécessite de disposer du degré de similarité mutuel de chaque paire d'objets ou de pouvoir l'évaluer. La donnée d'entrée d'une méthode de classification est alors soit un ensemble $E = \{x_1, \ldots, x_n\}$ de n objets à étudier sur chacun desquels est observé p variables communes, soit une matrice de degrés de (dis)similarités mutuelles entre les objets de l'ensemble E. A partir des données des observations des p variables sur les n objets peuvent être calculés divers indices de (dis)similarité. Le calcul de ces indices tient compte de la nature des variables, selon qu'elles sont (toutes) quantitatives, (toutes) qualitatives, ou hétérogènes (certaines quantitatives et d'autres qualitatives), voire complexes (type graphe, multimedia, etc). Dans la littérature, on distingue deux familles de méthodes de classification non supervisée : les méthodes hiérarchiques et les méthodes non hiérarchiques.
Les méthodes hiérarchiques les plus populaires sont celles qui construisent une hiérarchie sur l'ensemble des objets étudiés. Les résultats de ce type de classification sont généralement représentés sous la forme d'un dendrogramme (arbre de la classification hiérarchique). On distingue deux catégories d'approches de construction de hiérarchies : les approches divisives et les approches agglomératives, celles-ci, les plus populaires, utilisent des liens d'agrégation. Dans la littérature plusieurs ouvrages ont été consacrés à la classification non supervisée (Ghosh, 2003; Hennig *et al.*, 2015; Brucker & Barthélemy, 2007). L'ouvrage de Brucker et Barthélemy (Brucker & Barthélemy, 2007) peut être consulté pour plus de détails sur les méthodes hiérarchiques et de partitionnement. La Section 3.2.1 présente la classification ascendante hiérarchique connue comme étant la méthode de classification hiérarchique agglomérative la plus connue et utilisée.
Les méthodes non hiérarchiques les plus usuelles sont les méthodes de partitionnement, i.e., des méthodes qui révèlent des classes formant une partition de l'ensemble des objets étudié. Les méthodes k-means et k-medoids sont les méthodes de partitionnement les plus connues. Elles sont évoquées dans la Section 3.2.2. Une conclusion termine la

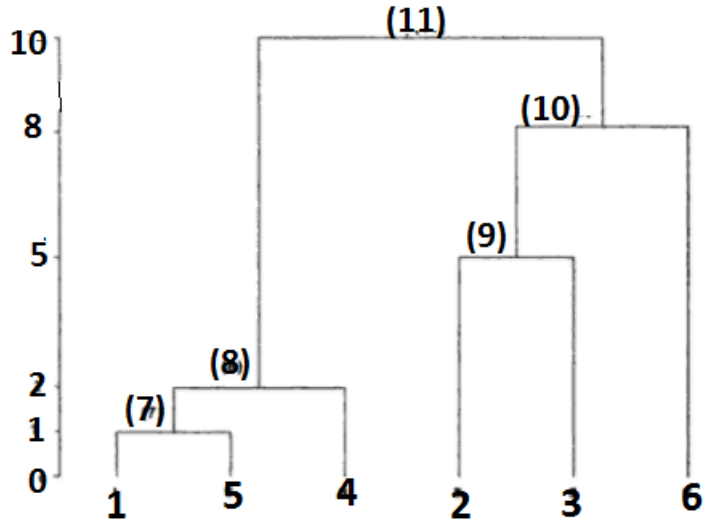


FIGURE 3.6 – Dendrogramme de la hiérarchie indicée (H_E, v) où H_E est la hiérarchie de l'exemple 1 et v l'indice de niveau de l'exemple 2.

section.

3.2.1 Méthodes hiérarchiques

L'objectif des méthodes hiérarchiques est la recherche d'une famille de classes qui forment une hiérarchie.

Définition 9. *Une hiérarchie totale de parties H_E d'un ensemble E est un sous-ensemble de l'ensemble $\mathfrak{P}(E)$ des parties de E satisfaisant les propriétés suivantes :*

(i) $E \in H_E$,
(ii) $\{i\} \in H_E$ *pour tout i élément E,*
(iii) $h \cap h' \in \{\emptyset, h, h'\}$, *pour h et h' deux éléments quelconques de* H_E, *autrement dit h et h' sont soit disjointes, soit incluses l'une dans l'autre.*

Exemple 1. *Pour* $E = \{1,2,3,4,5,6\}$, *l'ensemble* H_E = *{{1},{2},{3},{4},{5},{6},{1,5},{2,3},{1,5,4},{2,3,6},E} est une hiérarchie totale sur E.*

Définition 10. *Une hiérarchie indicée sur E est un couple (H_E, v) où H_E est une hiérarchie sur E et v est un indice de niveau sur H_E, i.e. une application strictement croissante définie sur H_E, à valeurs réelles non négatives et nulles sur les singletons. En d'autres termes :*
v : $H_E \mapsto R_+$ satisfait les propriétés suivantes :
(i) $\forall\ x \in E,\ v(\{x\}) = 0$
(ii) $h \subset h' \Rightarrow v(h) < v(h')$.

Exemple 2. *L'application v définie ci-dessous est un indice de niveau sur la hiérarchie H_E de l'exemple précédent.*

$$\begin{aligned} v(\{i\}) = 0 \text{ pour } i = 1,2,...,6 \\ v(\{1,5\}) = 1; v(\{2,3\}) = 5; v(\{1,4,5\}) = 2; v(\{2,3,6\}) = 8 \\ v(E) = 10 \end{aligned} \tag{3.9}$$

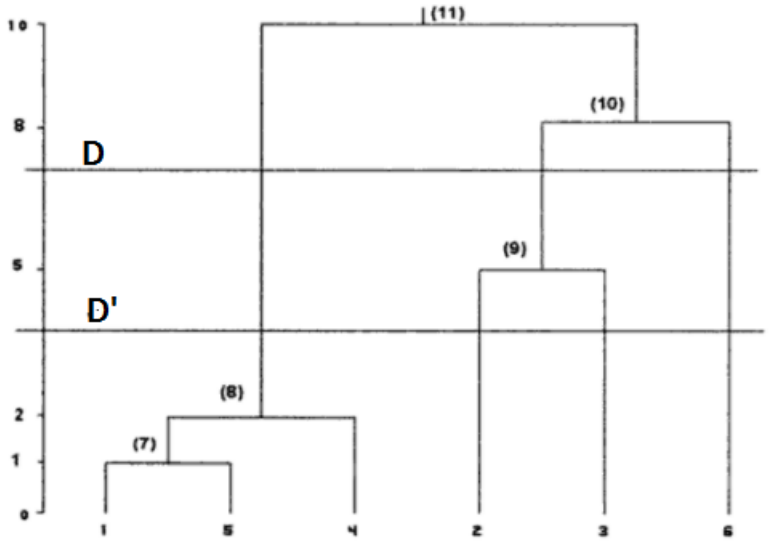


FIGURE 3.7 – Arbre hiérarchique indicé associé à H_E.

Une hiérarchie indicée (H, v) peut être visualisée par un dendrogramme qui est une représentation du diagramme de Hasse de H, gradué par v et dans laquelle les sommets sont matérialisés par des paliers horizontaux. La Figure 3.6 présente un dendrogramme de la hiérarchie indicée (H_E, v) où H_E est la hiérarchie de l'exemple 1et v l'indice de niveau de l'exemple 2.

Toute coupure du dendrogramme par une droite horizontale fournit une partition de l'ensemble E. Ainsi, dans la Figure 3.7, la droite D coupe l'arbre en trois classes composées des éléments {{1,4,5},{2,3},{6}} et la droite D' en quatre classes {{1,4,5},{2},{3},{6}}. Dans la littérature on distingue deux familles de méthodes de classification hiérarchiques : les méthodes ascendantes et les méthodes descendantes (ou divisives).
Les méthodes descendantes construisent une hiérarchie en choisissant à chaque étape une classe que l'on subdivise en deux sous-classes. Certaines de ces méthodes nécessitent d'étudier toutes les subdivisions binaires possibles. Dans ce cas, la méthode n'est pas polynomiale. D'autres minimisent des critères en utilisant des algorithmes polynomiaux.
Les méthodes ascendantes construisent une hiérarchie en fusionnant à chaque étape les deux classes les plus proches en se basant sur un lien d'agrégation. Dans ce qui suit nous présentons en particulier la méthode de classification ascendante hiérarchique (CAH).

L'algorithme de base pour la construction ascendante d'une hiérarchie comprend les étapes suivantes :

1. Munir l'ensemble E des éléments à classer d'une mesure de ressemblance d (distance ou indice de dissimilarité). Construire la matrice $M_d^{(n)}$ contenant les ressemblances des n éléments de E deux à deux selon la mesure d.
2. Choisir une "distance" D entre parties disjointes de E (critère d'agrégation) telle que $D(\{i\}, \{i'\}) = d(i, i')$ et utiliser l'algorithme général suivant :

(a) Agréger en un nouvel élément les deux éléments de E les plus proches au sens de D (donc au sens de d) ;

(b) Mettre à jour la matrice des distances en calculant les distances entre l'élément nouvellement formé et les (n-2) éléments restants ; obtention de la matrice $M_D^{(n-1)}$ d'ordre (n-1) ;

(c) Rechercher à nouveau des éléments les plus proches que l'on agrège ;

(d) Ainsi de suite, jusqu'à agréger tous les éléments de E en une seule classe.

Le choix de la mesure de ressemblance (ou indice de dissimilarité) d est une étape importante. Il est possible à cette étape pour l'utilisateur d'utiliser au mieux l'information a priori dont il dispose, afin de proposer une mesure pertinente de ressemblance entre observations.

Définition 11. *Une similarité sur E est une fonction s définie sur E^2 à valeurs dans $\mathbb{R}^+$, telle que pour tous $x, y \in E$:*
(i) $s(x,y) \leq s(x,x)$;
(ii) $s(x,y) = s(y,x)$.

Définition 12. *Une dissimilarité sur E est une fonction d définie sur E^2 à valeurs dans $\mathbb{R}^+$, telle que pour tous $x, y \in E$:*
(i) $d(x,x) = 0$;
(ii) $d(x,y) = d(y,x)$.

Remarquons qu'une distance est une dissimilarité satisfaisant la séparation ($d(x,y) = 0$ implique $x = y$) et l'inégalité triangulaire ($d(x,y) \leq d(x,z) + d(y,z)$). Plusieurs mesures ont été proposées dans la littérature. Les mesures les plus connues sont présentées dans le Tableau 3.4. Ces mesures peuvent être utilisées pour calculer une matrice de dissemblances entre individus provenant de données quantitatives.
Les données qualitatives peuvent être traitées avec les indices sur signes de présence/absence (Tableau 3.5), basés sur les quantités suivantes définies pour deux individus i et j :

- a_{ij} = nombre de caractères communs à i et j,
- b_{ij} = nombre de caractères possédés par i mais pas par j,
- c_{ij} = nombre de caractères possédés par j mais pas par i,
- d_{ij} = nombre de caractères que ne possèdent ni i ni j,
- a_{ij}+b$_{ij}$ + c_{ij} + d_{ij} = p.

Minkowsky	$D(x,y) = \left(\sum_{i=1}^{m} \lvert x_i - y_i \rvert^r \right)^{1/r}$
Manhattan	$D(x,y) = \sum_{i=1}^{m} \lvert x_i - y_i \rvert$
Chebychev	$D(x,y) = \max_{i=1}^{m} \lvert x_i - y_i \rvert$
Euclidienne	$D(x,y) = \left(\sum_{i=1}^{m} \lvert x_i - y_i \rvert^2 \right)^{1/2}$
Mahalanobis	$D(x,y) = (x-y) \sum^{-1} (x-y)^T$

TABLE 3.4 – Exemples de distance utilisées en classification de données quantitatives.

A partir de l'algorithme de base, deux catégories d'algorithmes ont été proposées pour la construction d'un arbre hiérarchique ascendant : les algorithmes d'agrégation fondés sur un lien métrique et les algorithmes d'agrégation fondés sur la densité. Concernant les algorithmes d'agrégation fondés sur un lien métrique, les critères usuels d'agrégation de la classification hiérarchique sont les quatre suivants :

— le critère du saut minimal (lien simple) (Sibson, 1973) ;
— le critère du diamètre (lien complet) (Sørensen, 1948) ;
— le critère de la moyenne (lien moyen) (Sokal, 1958) ;
— le critère de Ward (perte d'inertie minimale) (Ward Jr, 1963).

Ce dernier critère, le critère de Ward, est l'un des plus utilisés, précisément car il permet de minimiser à chaque étape l'inertie intra-classes des partitions obtenues. Il est défini de la manière suivante :

$$d(C_1, C_2) = \frac{n_1 \times n_2}{n_1 + n_2} d(G_1, G_2) \ (saut\ de\ Ward), \tag{3.10}$$

avec C_1 et C_2 deux classes d'une partition donnée ; n_1 et n_2 sont les effectifs des deux classes et G_1 et G_2 leurs centres de gravité respectifs.

Concordance	$D(x,y) = \frac{a_{ij}+d_{ij}}{p}$
Jaccard	$D(i,j) = \frac{a_{ij}}{a_{ij}+b_{ij}+c_{ij}}$
Dice	$D(i,j) = \frac{2a_{ij}}{2a_{ij}+b_{ij}+c_{ij}}$

TABLE 3.5 – Exemples de distance utilisées en classification de données qualitatives.

L'algorithme SLINK (Sibson, 1973) est un exemple d'algorithme implémentant le lien simple. La méthode de Voorhees (Voorhees, 1986) implémente le lien moyen, et l'algorithme CLINK (Defays, 1977) est un exemple d'implémentation du lien complet. l'algorithme AGNES (AGglomerative NESting) (Kaufman & Rousseeuw, 1990a) implémente les quatre types de lien.
Les algorithmes d'agrégation fondés sur la densité concernent les méthodes de classification qui utilisent des estimations de densité de probabilité non paramétriques et procèdent en deux étapes :

1. définition d'une mesure de dissimilarité $d*$ entre x_i et x'_i, basée sur les estimations de densité et de voisinage ; on peut également définir une distance D entre classe ;
2. application du critère du saut minimal à $d*$ (ou à D).

Les avantages des algorithmes hiérarchiques sont :

— Leur flexibilité concernant le niveau de finesse de la classification.
— Leur facilité de prise en compte de distances et d'indices de similarité de n'importe quel type.

Néanmoins, ils présentent plusieurs inconvénients :

— Il est difficile de déterminer la coupure significative de l'arbre : il n'y a pas de test statistique permettant de guider la décision à prendre pour couper l'arbre. Des critères externes (silhouette (Rousseeuw, 1987), statistique du gap (Tibshirani *et al.*, 2001)) sont utilisés pour choisir le nombre de classes à retenir (décision de nature subjective).

— La partition retenue à une étape dépend de celle obtenue à l'étape précédente.
— Les algorithmes fournissent toujours des classes à partir de n'importe quelles données. Les classes construites sont convexes. Des précautions sont à prendre pour interpréter les résultats.

Les méthodes hiérarchiques classiques, fondées sur un lien métrique, conduisent le plus souvent à des partitions formées de classes de forme convexe, de taille et densité sensiblement égales, sans tenir compte éventuellement de points atypiques. Or dans bien des analyses comme par exemple l'analyse de données spatiales, les classes révélées sont de forme arbitraire (non convexes, de forme et de taille différentes). Plusieurs méthodes hiérarchiques utilisées conjointement avec différentes méthodes de partitionnement ont été proposées dans la littérature parmi lesquelles CURE (Guha *et al.*, 1998), ROCK (Guha *et al.*, 2000) et CHAMELEON (Karypis *et al.*, 1999). Ces algorithmes conduisent à des classes de forme arbitraire.

3.2.2 Méthodes de partitionnement

L'objectif des méthodes de partitionnement est de construire une partition des éléments en k classes homogènes. Contrairement aux méthodes hiérarchiques, le nombre k de classes est un paramètre de ces méthodes et doit donc être spécifié a priori ou déterminé par la méthode utilisée.

Définition 13. *Soit S un ensemble à n éléments. Une partition de S est un ensemble de parties non vides de S, disjointes et dont l'union est S.*

Dans la littérature on distingue deux catégories de méthodes de partitionnement : les méthodes d'optimisation d'un critère sur l'ensemble des partitions à nombre de classes fixé et les méthodes de ré-allocation dynamique des individus à des centres de classes.
Le principe des méthodes de ré-allocation dynamique est de déterminer un ensemble de centres de classes (centroïdes) et d'affecter les individus d'un jeu de données à classer au centre (et donc à la classe qu'il représente) dont ils sont le plus proche selon une mesure de distance. Ces méthodes comprennent généralement deux étapes :
— La génération de la configuration initiale : un ensemble de centres est choisi pour initialiser les q classes. Ils peuvent être aléatoirement choisis ou construits ; la partition résultante dépend de cette configuration initiale.
— L'affectation itérative des individus et la mise à jour itérative des centres : les itérations s'arrêtent lorsqu'un critère (généralement l'inertie : somme des carrés des écarts au centre) ne décroît plus.

Les algorithmes K-means (Hartigan & Hartigan, 1975; Hartigan & Wong, 1979; Forgy, 1965; MacQueen *et al.*, 1967) et K-medoids (Kaufman & Rousseeuw, 1990b) sont les algorithmes de partitionnement par ré-allocation dynamique les plus connus.
Les méthodes d'optimisation d'un critère cherche à construire une partition à nombre de classes fixé qui optimise un certain critère. Les principaux critères utilisés sont :
— la séparation : une bonne partition présente des classes bien séparées ; l'objectif est de maximiser les distances entre classes ;
— l'homogénéité : l'objectif est d'avoir des classes les plus concises possible en minimisant le maximum des distances intra-classes ;

— la dispersion : l'objectif est de minimiser une fonction d'inertie.

Ces méthodes d'optimisation conduisent à des algorithmes NP-difficiles et le minimum atteint par une méthode de descente n'est pas optimal. Des méthodes d'optimisation stochastiques ont été proposées pour tenter de résoudre ce problème : la méthode de recherche Tabou (Glover & Laguna, 2013), le recuit simulé (Brown & Huntley, 1992) ou les méthodes génétiques (Babu & Murty, 1993).
Dans cette section nous décrivons les méthodes k-means et k-medoids qui présentent un intérêt particulier dans ce mémoire. Plus d'informations sur les méthodes de partitionnement sont disponibles dans (Xu & Wunsch, 2005; Brucker & Barthélemy, 2007; Rai & Singh, 2010; Hennig *et al.*, 2015).

K-means

L'idée de l'algorithme k-means est de classer un ensemble donné d'observations en un nombre k de classes disjointes, où la valeur k est fixée à l'avance. L'algorithme comporte deux phases. La première phase consiste à déterminer k centres de classe, une pour chaque classe. La deuxième phase consiste à prendre observation de l'ensemble de données et de l'affecter à la classe dont le centre de classe est le plus proche. La distance Euclidienne est généralement utilisée pour le calcul des distances deux à deux entre les observations et les centres de classe. Lorsque tous les points sont classés, la première étape est terminée et un premier regroupement est effectué. Il est alors nécessaire de recalculer les nouveaux centroïdes de chaque classe car l'insertion de nouvelles observations au sein des classes peut entrainer un changement de centroïdes. Les nouveaux centres de classe correspondent aux centres de gravité de chacune des classes obtenues. Une fois que les k nouveaux centroïdes sont déterminés, les distances entre les observations et ces centroïdes sont calculées et certaines observations peuvent être affecter à une autre classe dont le centroïde est plus proche. Ce processus est répété de manière itérative jusqu'à ce que les centroïdes ne subissent plus de changement et soient stables. L'atteinte de cette stabilité correspond à un critère de convergence pour l'algorithme. Le pseudo-code de l'algorithme K-means est présenté dans l'Algorithme 1.

Entrées:
$D = \{d_1, d_2, ..., d_n\}$: un ensemble de n observations
k : le nombre de classes désirées
Sorties:
A : un ensemble de k classes
début
    1. Choisir arbitrairement k centres de classes initiaux, a priori quelconques ;
    2. **répéter**
        Affecter chaque observation d_i à la classe dont le centre de classe est le plus proche ;
        Calculer le nouveau centre de classe de chaque classe ;
    jusqu'à *ce qu'un critère de convergence soit satisfait* ;
fin

Algorithm 1: k-means

L'algorithme k-means est un des algorithmes de partitionnement les plus étudiés et utilisés. Il est généralement efficace et produit de bons résultats. L'inconvénient majeure du k-means est qu'il produit des classes différentes pour des ensembles de centres de classes différents. La qualité finale des classes dépend donc fortement des centres de classes initiaux. Des méthodes comme celles de (Bradley & Fayyad, 1998; Babu & Murty, 1993) pour tenter de détermination des centroïdes optimaux ou de minimiser l'effet de l'initialisation des centres de classes ont été proposées. L'algorithme k-means original que nous présentons ici requiert un temps de calcul proportionnel au produit du nombre d'observations, du nombre de classes et du nombre d'itérations. Des versions optimisées de l'algorithme, par exemple l'algorithme ISODATA (Ball & Hall, 1965), ont été proposées (Brown & Huntley, 1992; Mao & Jain, 1996; Zhang, 2001).

K-medoids

Tandis que dans la méthode k-means un centre de classe correspond au centre gravité de la classe qu'il représente, dans la méthode k-medoids une classe est représentée par un de ses individus (médoïde). Une fois que les medoïdes sont choisis, des classes sont construites comme sous-ensembles des individus les plus proches des médoïdes selon une mesure de distance. Il existe différentes implémentations de la méthode des k-medoids :

- PAM (Partitioning Around Medoids) (Kaufman & Rousseeuw, 1990b),
- CLARA (Clustering LARge Applications) (Kaufman & Rousseeuw, 1990b),
- CLARANS (Clustering Large Applications based upon RANdomized Search) (Ng & Han, 2002).

Le principe de l'algorithme PAM est de :

1. Choisir un ensemble de medoïdes,
2. Affecter chaque observation au medoïde le plus proche,
3. Remplacer de manière itérative chaque medoïde par un autre si cela permet de réduire une distance globale.

L'avantage de l'algorithme PAM est sa robustesse par rapport au k-means en présence de bruit. La complexité de l'algorithme PAM est $O(k(n-k)^2)$, ce qui en fait un algorithme coûteux pour le partitionnement d'ensemble de grande taille et de valeur de k assez grande.
L'algorithme CLARA a été proposé pour classer des données de taille moyenne. Cet algorithme effectue une recherche locale des représentants de classes en opérant sur plusieurs échantillons de données de taille s extraits de l'ensemble de données entier. Ensuite, l'algorithme PAM est appliqué à chaque échantillon et la meilleur partition est sélectionnée.
L'algorithme CLARANS consiste à construire un graphe de classification de n individus et de k classes dont chaque nœud est représenté par un ensemble d'individus de taille k. Deux nœuds sont voisins, c'est à dire reliés par une arête, si leurs ensembles diffèrent d'un seul représentant. De manière générale, les classes obtenues avec l'algorithme CLARANS sont de meilleur qualité que celles obtenues par les algorithmes PAM et CLARA. Cependant, l'algorithme CLARANS est de complexité $O(kn^2)$ et implique donc de travailler avec des ensembles de données de petite taille.

Questions algorithmiques générales

Dans une majorité d'algorithmes de partitionnement le nombre k de classes à construire est un paramètre a priori à déterminer par l'utilisateur. En pratique l'utilisation de méthodes de partitionnement pose deux questions importantes : quelle valeur de k choisir et comment évaluer la partition obtenue ?
Beaucoup de critères ont été introduit afin de déterminer une valeur de k optimale (Bock, 1996; Fraley & Raftery, 1998; Oliver *et al.*, 1996). Un critère populaire est le **Silhouette** (Kaufman & Rousseeuw, 1990a). En considérant la distance moyenne entre un point x d'une classe C et les autres points de C et en comparant cette distance avec sa distance moyenne avec une classe G, le plus approprié autre que C, le coefficient Silhouette de x est :

$$\begin{aligned} s(x) &= \frac{b(x) - a(x)}{max\{a(x), b(x)\}}, \\ a(x) &= \frac{1}{|C| - 1} \sum_{y \in C, y \neq x} d(x, y), \\ b(x) &= min_{G \neq C} \frac{1}{|G|} \sum_{y \in G} d(x, y). \end{aligned} \tag{3.11}$$

Les valeurs de Silhouette proches de $+1$ correspondent à un choix approprié de partition et des valeurs proches de 0 correspondent à une partition de mauvaise qualité. La moyenne générale des valeurs de Silhouette individuel $s(x)$ donne une bonne indication de la qualité d'une partition.
Une autre approche d'évaluation de la qualité d'une partition est d'employer un **coefficient de partition** (Bezdek, 1981) égal à la somme des carrés des poids $w(x, C)$ d'une classe C (Equation 3.12).

$$W = \frac{1}{N} \sum_{x \in X} w(x, C(x))^2. \tag{3.12}$$

Chacune des mesures d'évaluation évoquées peut être graphiquement représentée en fonction de k et le graphique obtenu peut être utilisé afin de choisir la meilleure valeur de k. Mufti et al. (Mufti *et al.*, 2012) proposent également une mesure fondée sur l'indice de Rand, permettant de choisir le nombre de classes d'une partition.
Deux approches sont possibles pour l'évaluation d'une partition : la partition peut être évaluée par un expert ou par une procédure automatisée particulière. L'évaluation d'une partition par un expert est liée à deux problématiques : (1) l'interprétabilité des classes et (2) la visualisation des classes. L'interprétabilité des classes dépend de la technique utilisée. Les méthodes k-means et k-medoids génèrent des classes qui peuvent être interprétées comme des zones denses autour de centroïdes ou de médoïdes. Par conséquent leurs résultats sont facile à interpréter et à visualiser. L'étude de (Jain *et al.*, 1999) décrit de manière approfondie l'évaluation de partitions et une discussion sur la visualisation de classes peut être trouvée dans (Kandogan, 2001).
En ce qui concerne les procédures automatiques d'évaluation, un des principes appliqués consiste à construire deux partitions (à partir du même ensemble de données ou de sous-ensembles différents d'un même ensemble de données) et de les comparer. L'indice de Rand, par exemple, peut-être utilisé pour cet objectif. Le calcul de l'indice de Rand implique des pairs qui ont été assignées soit à la même classe soit à des classes différentes au sein des deux partitions. Toutefois sa complexité est en $O(N^2)$ et il n'est pas

toujours calculable. L'entropie conditionnelle $H(S|J)$ (Cover & Thomas, 2006) d'une étiquette de classe s étant donnée une partition est une autre mesure d'évaluation utilisée. D'autres mesures sont également utilisées, par exemple, la mesure-F (Larsen & Aone, 1999). Vendramin et al. (Vendramin *et al.*, 2010) présentent une étude comparative des critères de validité de partition les plus connues. Leur étude consiste à exécuter l'algorithme k-means sur un grand nombre de jeux de données en faisant varier la valeur de k de 2 à une certaine valeur k_{max}, puis la qualité des partitions obtenues sont mesurées avec 14 indices de qualité différents. Enfin, les performances des indices sont évaluées en calculant une certaine valeur de corrélation pour chaque indice et chaque jeu de données utilisés, puis en validant les valeurs de corrélation calculées grâce à un test statistique approprié. Leur étude révèle que de manière générale, les meilleures performances sont associées aux indices Silhouette (Kaufman & Rousseeuw, 1990a; Rousseeuw, 1987), PBM (Pakhira *et al.*, 2004), Calinski-Harabasz (Caliński & Harabasz, 1974) et point-biserial (Milligan & Cooper, 1985; Milligan, 1981). Les détails sur ces indices sont disponibles dans (Vendramin *et al.*, 2010).

3.3 Conclusion

Dans ce Chapitre nous avons présenté plusieurs techniques majeures de classification supervisée et non supervisée. Les techniques de classification supervisée présentées sont : les arbres de décision, les réseaux bayésiens et les réseaux de neurones artificiels. Les techniques de classification non supervisée présentées sont : la classification ascendante hiérarchique (appartenant à la famille des techniques de classification hiérarchique), k-means et k-medoids (appartenant à la famille des techniques de partitionnement). Ces techniques sont largement utilisées dans la littérature pour la recherche d'informations, le traitement et la classification d'images de documents. En particulier, nous avons évoqué dans le Chapitre 2 des approches, telles que celles de (Cesarini *et al.*, 2003) et de (Bartoli *et al.*, 2010), fondées sur des méthodes de classification pour le traitement d'images de factures.
Bartoli et al. (Bartoli *et al.*, 2010) proposent une méthode de classification de factures numérisées selon la densité de leurs pixels noirs et la densité des contours de texte. Deux classifieurs sont considérés : machine à vecteur de support et un classifieur fondé sur une mesure de distance.
Cesarini et al. (Cesarini *et al.*, 2003) proposent un système pour traiter les documents qui peuvent être regroupés en classes. Le système comprend trois phases :

1. analyse de documents ;
2. classification de documents à l'aide d'arbres de décision ;
3. compréhension de documents.

Dans ces deux approches, les méthodes de classification utilisées sont supervisées et les classes de documents définies sont, soit les émetteurs des documents, soit le type de documents. Par ailleurs, dans le cadre d'utilisation des méthodes de classification supervisée, il est nécessaire de disposer d'un jeu de données dont les étiquettes de classes sont connues. Nous avons évoqué dans la Section 2.4.1 du Chapitre 2 quelles sont les problématiques liées à la création de jeux de données dans le cadre de l'élaboration de systèmes d'extraction d'informations textuelles au sein d'images de documents. En effet,

nous avons mentionné que l'entrainement et l'évaluation d'un système d'extraction d'informations textuelles nécessite de disposer d'une base de données suffisamment grande et que la construction d'une telle base peut être une tâche difficile et couteuse en temps et en ressource.
Dans ce mémoire, nous présentons un système d'extraction d'informations textuelles au sein d'images de factures, fondé sur des méthodes de classification non supervisée (CAH et k-means) (Partie II Chapitre 6) combinées à un treillis de concepts. Nous verrons dans la Partie II Chapitre 6 comment d'une part, notre système facilite la création d'un large jeu de données adéquates et d'autre part, réalise la tâche d'extraction d'informations textuelles au sein d'images de documents quelque soit l'émetteur des documents.
Dans le Chapitre suivant nous introduisons les notions propres à l'analyse formelle de concepts et en particulier la notion de treillis de concepts d'un contexte formel.

Chapitre 4

Analyse formelle de concepts

Sommaire

4.1 Notions de base 91

4.2 AFC et fouille de règles d'association 94

4.3 AFC et recherche d'informations 96

4.4 Diverses autres applications de l'AFC 97

4.5 Conclusion 100

L'analyse formelle de concepts (AFC) a été introduite au début des années 80 par Rudolf Wille (Wille, 1982) comme théorie mathématiques prenant ses racines dans les travaux de Birkhoff (Birkhoff, 1940) et ceux de Barbut et Monjardet (Barbut & Monjardet, 1970), pour la formalisation de concepts. L'AFC a été appliquée dans beaucoup de domaines tels que la découverte de connaissances, le génie logiciel et la recherche d'informations, au cours des 15 dernières années. Le fondement mathématique de l'AFC est décrite par (Ganter & Wille, 1999). Une revue de la littérature publiée jusqu'à 2004 sur l'historique mathématique et philosophique de l'AFC et certaines applications de l'AFC en recherche d'informations et découverte de connaissances, ainsi qu'en logique et intelligence artificielle, peut être trouvée dans (Priss, 2006). Une comparaison d'algorithmes pour la génération de treillis de concepts est donnée par (Kuznetsov & Obiedkov, 2002). Une revue des logiciels d'AFC disponibles est fournie par (Tilley, 2004). Carpineto et al. (Carpineto & Romano, 2004) présentent un panorama des applications de l'AFC en recherche d'informations. Dans (Tilley & Eklund, 2007), une revue de 47 articles concernant le génie logiciel fondé sur l'AFC est présentée. Les auteurs ont catégorisé ces articles selon les 10 catégories définies dans le standard ISO 12207 sur le génie logiciel. Dans (Lakhal & Stumme, 2005) une étude des techniques de fouille de règles d'association fondée sur l'AFC est donnée. Poelmans et al. (Poelmans *et al.*, 2010) fournissent une revue compréhensive de la recherche fondée sur l'AFC en découverte de connaissances et en fouille de données. Poelmans et al. (Poelmans *et al.*, 2012) et Carpineto et al. (Carpineto *et al.*, 2004) fournissent une revue de la littérature sur l'AFC appliquée à la recherche d'informations. En raison de leur lien avec le sujet de la thèse, seront considérées les applications en rapport avec la fouille de règles d'association (Section 4.2), la recherche d'informations (Section 4.3) et diverses autres domaines (Section 4.4).

4.1 Notions de base

Afin d'introduire les notions relatives à l'analyse formelle de concepts, nous nous appuyons sur la formalisation de ces notions présentée dans les articles de Benayade et Diatta (Benayade & Diatta, 2008) et de Diatta (Diatta, 2003).
En analyse formelle de concepts, un contexte formel est un triplet $\mathbb{K} = (\mathcal{O}, \mathcal{A}, \mathcal{R})$, où $\mathcal{O}$ et $\mathcal{A}$ sont des ensembles et $\mathcal{R}$ une relation binaire de $\mathcal{O}$ vers $\mathcal{A}$ (Wille, 1982; Ganter & Wille, 1999). Les éléments de $\mathcal{O}$ sont appelés objets et ceux de $\mathcal{A}$ attributs. La relation $\mathcal{R}$ induit une correspondance de Galois entre les ensembles ordonnés $(\mathcal{P}(\mathcal{O}), \subseteq)$ et $(\mathcal{P}(\mathcal{A}), \subseteq)$ par le biais des applications :

$$f : X \mapsto \bigcap_{x \in X} \{a \in \mathcal{A} : (x, a) \in \mathcal{R}\}$$

et

$$g : I \mapsto \bigcap_{a \in I} \{x \in \mathcal{O} : (x, a) \in \mathcal{R}\},$$

pour $X \subseteq \mathcal{O}$ et $I \subseteq \mathcal{A}$, i.e. :

$$\begin{aligned} &(G1)\ \text{X} \subseteq Y \text{ implique } f(Y) \subseteq f(X), \\ &(G2)\ \text{I} \subseteq J \text{ implique } g(J) \subseteq g(I), \\ &(G3)\ \text{X} \subseteq g(f(x)) \text{ et } I \subseteq f(g(I)). \end{aligned} \tag{4.1}$$

Ainsi, les applications $\phi = f \circ g$ et $\psi = g \circ f$ sont, respectivement, des opérateurs de fermeture dans $(\mathcal{P}(\mathcal{A}), \subseteq)$ et $(\mathcal{P}(\mathcal{O}), \subseteq)$.
Étant donné un ensemble E, un opérateur de fermeture dans $(\mathcal{P}(E), \subseteq)$ est un opérateur $\varphi : \mathcal{P}(E) \to \mathcal{P}(E)$ tel que :

$$\begin{aligned} &(C1)\ X \subseteq \varphi(X), \\ &(C2)\ \varphi(\varphi(X)) = \varphi(X), \\ &(C3)\ X \subseteq Y \textit{ implique } \varphi(X) \subseteq \varphi(Y). \end{aligned} \tag{4.2}$$

Une partie X de E est dite φ-fermée (ou tout simplement fermée) si elle est égale à sa fermeture $\varphi(X)$.
Soit $\mathbb{K} = (\mathcal{O}, \mathcal{A}, \mathcal{R})$ un contexte formel. Un *concept formel* de $\mathbb{K}$ est un couple $c = (X, I)$ tel que $f(X) = I$ et $g(I) = X$. Les parties X et I sont respectivement appelées *extension* et *intention* de c.
Soit $G(\mathbb{K})$ l'ensemble des pairs $(X, Y) \in P(\mathcal{A}) \times P(\mathcal{O})$ tel que $\varphi(X) = X$ et $f(X) = Y$. Alors, $G(\mathbb{K})$, doté de l'ordre définit par $(X_1, Y_1) \leq (X_2, Y_2)$ est un treillis complet, si et seulement si $X_1 \subseteq X_2$ (de manière équivalent, $Y_2 \subseteq Y_1$).
$G(\mathbb{K})$ est appelé le *treillis de Galois* de la relation binaire $\mathcal{R}$ (Barbut & Monjardet, 1970). Dans le contexte de l'analyse formelle de concepts, la notion de treillis de concepts est utilisée de manière équivalente à la notion de treillis de Galois.
Le contexte formel $\mathbb{K} = (\mathcal{O}, \mathcal{A}, \mathcal{R})$ peut être représenté dans un tableau constitué de lignes (les objets) et de colonnes (les attributs). Une croix est présente à l'intersection d'une ligne x et d'une colonne y si et seulement si l'objet x possède l'attribut y. Le Tableau 6.8 est un exemple de contexte formel dans lequel les objets sont des articles scientifiques, les attributs sont des termes et la relation binaire montre l'occurrence des termes dans les articles. Un article x possède un terme y si et seulement si le titre ou le résumé de l'article contient ce terme. On peut lire par exemple que l'article 1 possède

	browsing	mining	software	web services	FCA	information retrieval
article 1	X	X	X		X	
article 2			X		X	X
article 3		X		X	X	
article 4	X		X		X	
article 5				X	X	X

TABLE 4.1 – Exemple de contexte formel.

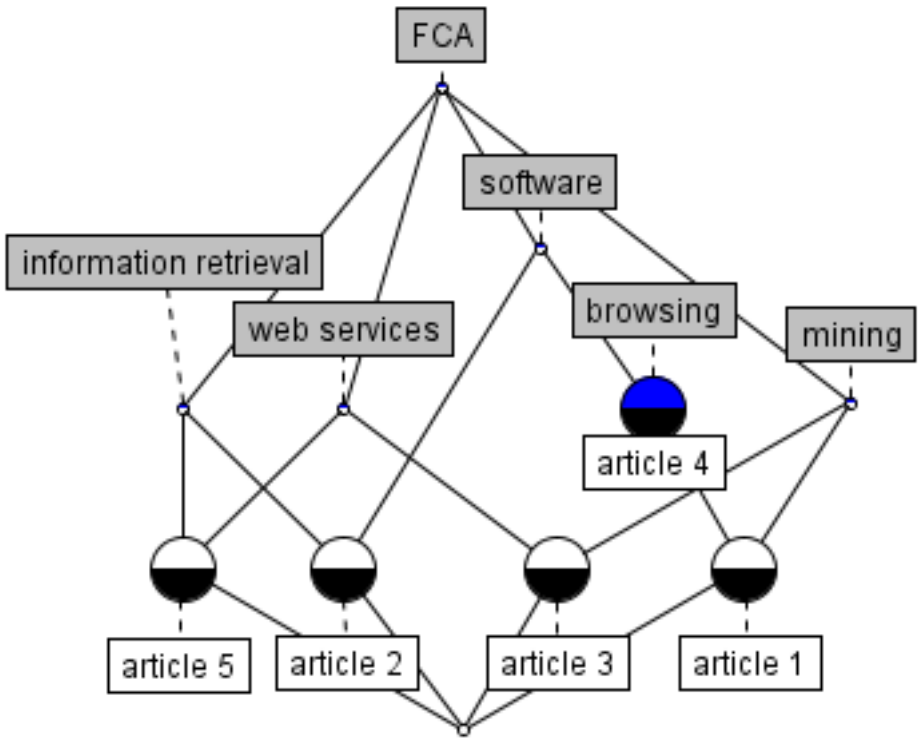


FIGURE 4.1 – Diagramme de Hasse du treillis de Galois correspondant au contexte formel du Tableau 6.8

l'attribut "browsing". Les données du Tableau 6.8 est un extrait des données utilisées par (Poelmans *et al.*, 2013b) dans leurs recherches.

Illustrons la notion de concept formel (ou concept) d'un contexte formel en utilisant les données du Tableau 6.8. Pour un ensemble d'objets $X \subseteq \mathcal{O}$, l'ensemble de leurs attributs communs peut être défini par :

$$I = X' = \{a \in \mathcal{R} | (x, a) \in \mathcal{R} \; pour \; tout \; x \in X\}. \tag{4.3}$$

Par exemple, en considérant les attributs que possède l'article 4 dans le Tableau 6.8 et en collectant tous les articles du contexte formel qui partagent ces attributs, nous obtenons un ensemble $X \subseteq \mathcal{O}$ constitué des articles 1 et 4. L'ensemble X d'objets possède l'ensemble I constitué des attributs "browsing", "software" et "FCA".
Le couple $c = (X, I)$ est un concept du contexte formel du Tableau 6.8. Ce concept a pour extension l'ensemble $X = \{article1, article4\}$ et pour intention l'ensemble $I = \{browsing, software, FCA\}$.
Le treillis de Galois correspondant au contexte formel du Tableau 6.8 peut-être visualisé sur un diagramme de Hasse, où les nœuds représentent des concepts et où les arêtes connectent des pairs de concepts voisins. Le diagramme de Hasse du treillis de Galois du contexte formel du Tableau 6.8 est présenté dans la Figure 4.1. Ce diagramme a été construit avec l'outil *conexp*[1]. Sur ce diagramme de Hasse une représentation succincte est utilisée pour représenter les informations à propos des intentions et des extensions de concepts formels. Dans cette représentation succincte, si une étiquette d'attribut A est attachée à un concept, cela signifie, que cet attribut apparait dans les intentions de tous les concepts atteignables, en descendant dans le treillis, à partir de ce concept jusqu'au "concept bottom" (le concept le plus bas du treillis). Si une étiquette d'objet O est attachée à un concept, cela signifie, que l'objet O figure dans les extensions de tous les concepts atteignables, en remontant dans le treillis, à partir de ce concept jusqu'au "concept top" (le concept le plus haut du treillis). Sur le diagramme de Hasse de la Figure 4.1, un nœud bleu et noir signifie qu'il y a un attribut attaché au concept représenté par ce nœud. Un nœud blanc et noir signifie qu'il y a un objet attaché au concept représenté par ce nœud.

Poelmans et al. (Poelmans *et al.*, 2013b) fournissent une large étude dans laquelle ils analysent la littérature sur l'analyse formelle de concepts. 1072 articles publiés entre 2003 et 2011 mentionnant des termes relatifs à l'AFC dans le titre ont été collectés. Les 1072 articles ont été regroupés selon un certain nombre de caractéristiques dans le cadre des recherches en AFC. Les articles ont été visualisés en utilisant des treillis de concepts, ce qui a facilité l'exploration et l'analyse de la littérature. Il en résulte que la découverte de connaissances est le thème de recherche le plus populaire. En effet ce thème couvre 23% des articles collectés et analysés dans (Poelmans *et al.*, 2013b). Récemment, l'amélioration du passage à l'échelle de l'AFC pour de plus larges et complexes ensembles de données a émergé en tant que nouveau sujet de recherche couvrant 9% des 1072 articles. En particulier, Poelmans et al. remarquent que plus d'un tiers des articles dédiés à ce sujet traite de problématiques dans le domaine de la découverte de connaissances. Un autre sujet de recherche important est la recherche d'informations couvrant 13% des articles. 36 des articles sur la recherche d'informations décrivent une combinaison de la recherche d'informations avec la découverte de connaissances et dans 27 des articles

1. https ://sourceforge.net/projects/conexp/

sur la recherche d'informations les auteurs font usage des ontologies. 15 articles sur la recherche d'informations traitent de la recherche de structures de logiciels dont la recherche de composants de logiciels.

4.2 AFC et fouille de règles d'association

Considérons un contexte formel $\mathbb{K} = (\mathcal{O}, \mathcal{A}, \mathcal{R})$ où $\mathcal{O}$ et $\mathcal{A}$ sont des ensembles finis non vides. Un tel contexte sera appelé contexte de fouille de données. Les éléments de $\mathcal{A}$ seront appelés items et ses parties motifs. Un motif constitué de p items sera appelé p-motif. Le support d'un motif P dans $\mathbb{K}$ est le nombre réel

$$supp(P) = \frac{|(g(P)|}{|\mathcal{O}|},$$

où $|X|$ désigne le nombre d'éléments de X. Le nombre $|(g(P)|$ que nous appellerons le support absolu de P sera également noté $spc(P)$. Étant donné un support minimum seuil $minsupp \in [0, 1]$, un motif P sera dit *minsupp-fréquent* (ou tout simplement fréquent) si $supp(P) \geq minsupp$.
Par ailleurs, comme l'ensemble des règles générées peut être de très grande taille, on s'intéresse plutôt à lui trouver des représentations compactes. Parmi les diverses représentations compactes possibles ce sont les bases qui ont la bonne propriété de n'induire aucune perte d'information. En effet, une base est un ensemble minimal de règles à partir duquel toutes les règles valides peuvent être retrouvées par application d'axiomes d'inférence donnés. Guigues et Duquenne (Guigues & Duquenne, 1986) et, plus tard, Luxenburger (Luxenburger, 1991), ont, respectivement, caractérisé une base pour les règles exactes et approximatives, pour lesquelles aucune condition de support n'est requise. Ces bases ont été adaptées aux règles d'association par Zaki et Ogihara (Zaki & Ogihara, 1998) (uniquement la base de Guigues-Duquenne), Pasquier et al. (Pasquier *et al.*, 1999), Stumme et al. (Stumme *et al.*, 2001), qui ont placé le problème de la fouille de règles d'association dans le cadre théorique latticiel de l'analyse formelle de concepts.

Définition 14. *Une règle d'association est un couple $r := (P, Q)$ de motifs, noté $P \rightarrow Q$, où Q est non vide.*
Les motifs P et Q seront respectivement appelés prémisse et conséquent de r.
Les support et confiance de r sont respectivement définis par :

$$supp(r) = \frac{|(g(P \cup Q)|}{|\mathcal{O}|}$$

$$et\ conf(r) = \frac{supp(P \cup Q)}{supp(P)}.$$

Si $conf(r) = 1$, alors r sera appelé règle d'association exacte ; autrement, r sera appelé règle d'association approximative.

Étant donnés des support et confiance minimums seuils $minsupp, minconf \in [0, 1]$, une règle d'association r sera dite $(minsupp, minconf)$-valide (ou tout simplement valide) dans le contexte $\mathbb{K}$ si $supp(r) \geq minsupp$ et $conf(r) \geq minconf$. L'ensemble de

toutes les règles d'association $(minsupp, minconf)$-valide dans $\mathbb{K}$ sera noté $\sum_{minsupp}^{minconf}$ (ou tout simplement $\sum$).
On notera que lorsque le support minimum seuil *minsupp* est fixé à 0, les règles d'association exactes sont aussi connues sous le nom d'*implications* (Guigues & Duquenne, 1986) ou de *règles d'implication* (Carpineto *et al.*, 1999), alors que les règles d'association approximatives sont connues sous le nom d'*implications partielles* (Luxenburger, 1991). L'ensemble $\sum$ comportant souvent un très grand nombre de règles, il peut être intéressant de lui trouver une base, i.e., un ensemble minimal (au sens de l'ordre d'inclusion) à partir duquel il peut être reconstitué par application d'axiomes d'inférence donnés. Les bases de Guigues-Duquenne et de Luxenburger pour les implications et implications partielles ont été adaptées aux règles d'association exactes et partielles, respectivement (Pasquier *et al.*, 1999; Zaki & Ogihara, 1998; Stumme *et al.*, 2001).

Définition 15. *La base de Guigues-Duquenne pour les règles d'association exactes est l'ensemble $GDB_{minsupp}$ (ou tout simplement GDB) défini par*

$$GDB = \{P \rightarrow \phi(P) \backslash P : P \ est \ \phi - critique \ frequent\}.$$

Un motif P est dit ϕ-critique s'il n'est pas ϕ-fermé et $\phi(Q) \subset P$ pour tout motif ϕ-critique Q strictement contenu dans P (Caspard Monjardet, 2003). Notons que les motifs ϕ-critiques sont aussi connus sous le nom de pseudo-intentions (Wille, 1992).

Définition 16. *La base de Luxenburger pour les règles d'association approximatives est l'ensemble $LB_{minsupp}^{minconf}$ (ou tout simplement LB) défini par*

$$\begin{aligned} LB = \{(r, supp(r), conf(r)) : r = P \rightarrow Q, P = \phi(P), \\ Q = \phi(Q), P \prec Q, supp(Q) \geq minsupp, conf(r) \geq minconf\}. \end{aligned} \tag{4.4}$$

Étant donnés deux motifs fermés P et Q, $P \prec Q$ signifie que $P \subset Q$ et il n'existe pas de motif fermé Q' tel que $P \subset Q' \subset Q$; $P \subset Q$ se lit "P est couvert par Q" ou "Q couvre P".

D'après l'étude de (Poelmans *et al.*, 2010) sur 702 articles publiés entre 2003 et 2009 la fouille de règles d'association couvre 25% des publications utilisant l'AFC pour la découverte de connaissances et la fouille de données. La fouille de règles d'association à partir d'une base de données de transactions requiert la détection de motifs qui apparaissent fréquemment appelés itemsets fréquents. Des approches récentes pour la fouille d'itemsets fréquents utilisent le paradigme de l'itemset fermé afin de limiter l'effort de recherche au sous-ensemble d'itemsets fréquents fermés. Nehmé (Nehmé *et al.*, 2005) propose une méthode pour calculer la famille de générateurs minimaux (i.e. des itemsets non fermés minimaux - au sens de l'inclusion - dans leur classe de fermeture). Tekaya et al. (Tekaya *et al.*, 2005) propose un algorithme nommé GenAll pour construire un treillis de concepts dans lequel chaque concept est étiqueté par ses générateurs minimaux dans le but de dériver des bases génériques de règles d'association. Dans (Hamrouni *et al.*, 2005a), l'extraction de bases génériques de règles d'association de taille réduite est discutée afin de réduire le nombre important de règles d'association résultant de la fouille de règles d'association. Hamrouni et al. (Hamrouni *et al.*, 2005b) proposent un algorithme nommé PRINCE qui construit un treillis à partir duquel la dérivation des règles d'association génériques devient facile. Dong et al. (Dong *et al.*, 2005) introduisent le

système concis de générateurs minimaux (SSMG) comme une représentation minimale des générateurs minimaux de tous les concepts et donnent un algorithme efficace pour la fouille de SSMGs. Les SSMGs sont ainsi utilisés pour réduire sans perte la taille de la représentation de tous les générateurs minimaux. Hamrouni et al. (Hamrouni *et al.*, 2007) présentent une nouvelle mesure de densité pour des contextes formels en utilisant le cadre donné par les SSMGs. Leur mesure est une agrégation de deux mesures complémentaires que sont les mesures de concision et de compacité de chaque classe d'équivalence induite par l'opérateur de fermeture. Ceci est important pour la performance des algorithmes de fouille d'itemsets fréquents fermés. La performance de ces algorithmes est intimement dépendante du type de contexte traité, c'est à dire si il est dense ou éparpillé.
Valtchev et al. (Valtchev *et al.*, 2004) discutent des techniques existantes de fouille de règles d'association fondées sur l'AFC et fournissent un guide pour la conception de nouvelles techniques afin de permettre l'application de l'AFC à un plus large ensemble de situations. Ils proposent ainsi deux méthodes en ligne pour le calcul de générateurs minimaux d'un système de fermés. Gupta et al. (Gupta *et al.*, 2005) discutent de comment des règles de classification fondées sur des règles d'association peuvent être générées en utilisant des treillis de concepts. Valtchev et al. (Valtchev *et al.*, 2008) montrent comment des implications peuvent être fouillées de manière incrémentale et efficacement à chaque fois qu'une transaction est ajoutée à une base de données. Maddouri (Maddouri, 2005) discute de la découverte de règles d'association et propose une approche pour découvrir des itemsets intéressants tels que les concepts optimaux couvrant une table binaire. Maddouri et al. (Maddouri & Kaabi, 2006) résument beaucoup de mesures statistiques introduites pour la sélection de concepts formels pertinents. Maddouri et al. (Meddouri & Maddouri, 2009) présentent une méthode pour la construction de seulement une partie du treillis incluant les meilleurs concepts et utilisé comme classification de règles.
Wollbold et al. (Wollbold *et al.*, 2008) utilisent l'AFC pour construire une base de connaissances constituée d'un ensemble de règles de sorte que le raisonnement sur des dépendances temporelles au sein de réseaux de gènes soit possible. Zhou et al. (Zhou *et al.*, 2004) utilisent l'AFC pour découvrir des règles d'association à partir de fichiers d'usage du web pouvant être utilisées pour des applications en ligne telles que la personnalisation et la recommandation de sites web. Richards et al. (Richards & Malik, 2003) discutent de la découverte de connaissances multi-niveaux à partir de bases de règles. Ces connaissances multi-niveaux sont importantes afin de permettre des requêtes à l'intérieur et à travers différents niveaux d'abstraction. L'AFC est utilisée pour extraire et représenter des connaissances sous la forme d'une base de règles canoniques non redondantes avec des implications minimales.

4.3 AFC et recherche d'informations

Les travaux de Carpineto et Romano (Carpineto & Romano, 1993) (initialement influencées par les travaux de Godin et al. (Godin *et al.*, 1989, 1993; Godin & Mili, 1993)) sont à l'origine du moteur CREDO qui facilite une méta-recherche des résultats de Google en se basant sur un treillis de concepts. Une revue de leurs travaux et de leurs applications en AFC et en recherche d'informations de manière générale peut-être trouvée dans (Carpineto *et al.*, 2004). Dans cet article, Carpineto et Romano argumentent

que l'AFC peut servir à trois objectifs en recherche d'informations.

1. Premièrement, l'AFC peut prendre en charge l'affinement de requêtes. Comme un treillis de concepts d'un contexte formel *documents* × *termes* structure un espace de recherche en classes de documents liés, un tel treillis peut être utilisé pour faire des suggestions pour l'extension de requêtes dans des cas où trop peu de documents sont retrouvés et pour l'affinement de requêtes dans des cas où trop de documents sont retrouvés.
2. Deuxièmement, les treillis de concepts supportent des actions d'interrogation et de navigation. Une requête initiale identifie un nœud de départ dans le treillis de concepts d'un contexte formel *documents* × *termes*. Un utilisateur peut alors naviguer au sein du treillis de concept en concept. Des requêtes supplémentaires sont ensuite utilisées pour "élaguer" le treillis de concepts afin d'aider les utilisateurs à orienter leurs recherches.
3. Troisièmement, une hiérarchie de thésaurus peut être intégrée à un treillis de concepts. Cette idée est abordée par différents chercheurs tels que (Carpineto & Romano, 1996), (Skorsky, 1997) et (Priss, 1997) mais n'est pas encore totalement résolue.

En dehors de CREDO, une seconde application de l'AFC qui a atteint une qualité professionnelle est le logiciel Mail-Sleuth (Eklund *et al.*, 2004). Ce logiciel est vendu par une entreprise Australienne et consiste en un plug-in, pour le logiciel MS Outlook, qui peut être utilisé pour la fouille de grandes archives d'emails. Le développement de ce logiciel est fondée sur des travaux antécédents sur la recherche d'informations à partir de textes semi-structurés (Cole & Stumme, 2000; Cole & Eklund, 2001).
De manière générale, les logiciels d'AFC semblent être une promesse pour des applications à la recherche d'informations, cependant avec quelques limitations. L'AFC n'est pas appropriée pour la manipulation directe de sources de données très grandes. Selon (Priss, 2006) il est difficile de donner des limites supérieures précises car cela dépend de chaque application. La taille des ensembles d'objets et d'attributs compte également. Priss précise que l'AFC a été appliqué des milliers de documents (Rock & Wille, 2000), mais n'est probablement pas directement applicable à la base de données complète de Google. Néanmoins elle peut être appliquée comme outil secondaire pour réorganiser un ensemble de documents résultant d'une recherche Google (comme le fait CREDO). Les technologies fondées sur l'AFC prétendent être centrées sur l'humain à cause des fondements philosophiques de l'AFC, mais peu d'études pratiques telle que celle de (Eklund *et al.*, 2004) sur leur usage existent.

4.4 Diverses autres applications de l'AFC

Depuis l'introduction des treillis de Galois dans les années 1970 et de l'AFC en 1982, les treillis de concepts sont devenus un outil relativement bien connu. D'après l'étude de (Poelmans *et al.*, 2013a) plus de 1000 articles ont été publiés sur l'AFC entre 2003 et 2013 et beaucoup d'entre eux contiennent des cas d'études montrant l'utilité de l'AFC dans la pratique. La première étude menée par Poelmans et al. concerne les méthodes de fouille de texte et de linguistique utilisant l'AFC. Au cours des dernières années l'AFC a été appliquée dans plusieurs projets de fouille de texte allant de l'adaptation de recettes

de cuisine à l'identification de criminels dans des centaines de milliers de rapports de polices. Bien que la majorité des papiers étudiés concernant la fouille de texte décrivent une preuve de concepts pour un système de fouille de texte fondé sur l'AFC, ils montrent clairement le potentiel de l'AFC dans ce domaine. L'AFC a également plusieurs applications en linguistique où cette théorie est utilisée pour visualiser et mieux comprendre des bases de données lexicales telles que le thésaurus de Roget [2] et la base de données WordNet [3].
Le second domaine étudié par Poelmans et al. est la fouille du Web. Beaucoup d'attention a été portée à ce domaine. En particulier, l'optimisation (Carpineto *et al.*, 2004; Koester *et al.*, 2005; Koester, 2006) et la personnalisation de résultats de recherche sur le Web (Cho & Richards, 2004; Beydoun, 2008; Beydoun *et al.*, 2007; Zhou *et al.*, 2004) reçoivent un intérêt considérable par la communauté de l'AFC.

Dans une autre étude de Poelmans et al. (Poelmans *et al.*, 2013b) un sous-ensemble relativement petit des articles concernant la découverte de connaissances et la fouille de données étudiés s'intéressent à la relation entre l'AFC et d'autres techniques d'apprentissage automatique. Les modèles d'apprentissage automatique sont initialisés à partir de données d'entrées constituées d'exemples positifs et négatifs d'une étiquette de classe cible w et tentent de construire une généralisation des exemples positifs qui ne couvriraient aucun exemple négatif. Des tâches importantes d'apprentissage automatique où des treillis de concepts sont utilisés comprennent la génération de bases d'implications, de bases de règles d'association et d'itemsets fermés. Plusieurs auteurs ont tenté de formuler un certain nombre de méthodes d'apprentissage automatique selon les termes de l'AFC. Ganter et Kuznetsov (Ganter & Kuznetsov, 2003) expriment l'apprentissage d'espace de versions (espace d'hypothèses pour induire des concepts généraux ou des règles en apprentissage supervisé) en terme d'AFC. Dans (Kuznetsov, 2004a) l'auteur décrit l'apprentissage d'espace de versions et d'arbres de décision dans le langage de l'AFC. La première approche pour la formulation de modèles d'apprentissage à partir d'exemples positifs et négatifs dans des treillis de concepts utilise l'AFC standard (Kuznetsov, 2004a,b; Ganter & Kuznetsov, 2000).
Fu et al. (Fu *et al.*, 2004) réalisent une étude comparative d'algorithmes de classification fondés sur l'AFC tels que GRAND (Oosthuizen, 1996), LEGAL (Liquière & Mephu Nguifo, 1990), GALOIS (Carpineto & Romano, 1993), RULEARNER (Sahami, 1995), CIBLe (Njiwoua & Mephu Nguifo, 1999) et CLNN & CLNB (Xie *et al.*, 2002). Nguifo et Njiwoua. (Nguifo & Njiwoua, 2001) proposent IGLUE, un algorithme combinant des techniques d'apprentissage fondées sur des treillis et des instances. Ricordeau et Liquière. (Ricordeau, 2003; Ricordeau & Liquiere, 2007) utilisent l'AFC pour généraliser des stratégies d'apprentissage par renforcement en regroupant des états similaires en utilisant leurs descriptions. Une stratégie représente la probabilité pour un agent de sélectionner une action a en étant dans un état s. Aoun-Alla et Mineau (Aoun-Allah & Mineau, 2006) proposent une méthode pour la fouille de données distribuées qui traitent d'abord des jeux de données de manière distribuée et utilise ensuite l'AFC pour recueillir les résultats au sein d'un ensemble de règles afin de former un méta-classifieur. Rudolph (Rudolph, 2007) propose d'utiliser l'AFC pour concevoir une architecture de réseaux de neurones dans le cas où des informations partielles sur le comportement désiré

2. http ://www.roget.org/
3. https ://wordnet.princeton.edu/

des réseaux sont déjà connues et peuvent être indiquées sous la forme d'implications sur l'ensemble des caractéristiques. Tsopze et al. (Tsopzé *et al.*, 2007) proposent l'algorithme CLANN qui utilise des treillis de concepts pour construire l'architecture d'un réseau de neurones. Nguifo et al. (Nguifo *et al.*, 2008) ont plus tard étendu cette approche aux jeux de données multi-classes en concevant l'algorithme M-CLANN.
Belohlavek et al. (Belohlavek *et al.*, 2009) utilisent l'AFC pour induire des arbres de décision à partir de tableaux de données. Dans une première étape, les attributs catégoriels sont transformés en attributs logiques. Ensuite une version modifiée de l'algorithme *next-neighbor* (Lindig, 2000) est utilisée pour construire un treillis de concepts réduit. A partir de ce treillis, des arbres de concepts peuvent être sélectionnés. Un arbre de concepts peut ensuite être transformé en arbre de décision. Outrata (Outrata, 2010) utilise l'AFC comme technique de pré-traitement de données. Les auteurs utilisent une analyse de facteurs booléens pour transformer l'espace des attributs afin d'améliorer les résultats d'un apprentissage automatique et en particulier l'induction des arbres de décision. L'analyse de facteurs booléens est une méthode qui décompose une matrice binaire I de taille $n \times m$ en un produit booléen $A \circ B$ d'une matrice binaire A de taille $n \times k$ et d'une matrice binaire B de taille $k \times m$, avec k le plus petit possible. Pour calculer la décomposition, les algorithmes présentés dans (Belohlavek & Vychodil, 2009, 2010) sont utilisés avec un critère modifié d'optimalité des facteurs calculés. Dans une première variante, l'ensemble F constitué de k facteurs est ajouté à la collection des m attributs utilisés en entrée de la méthode d'apprentissage. Dans une seconde variante, les m attributs originaux sont remplacés par k facteurs. Les auteurs valident la seconde variante de pré-traitement de données en appliquant des arbres de décision et une méthode d'apprentissage fondée sur des instances sur le dépôt de jeux de données d'apprentissage de l'UCI. Les modèles obtiennent une performance moyenne meilleure sur des jeux de données de test.
Visani et al. (Visani *et al.*, 2011) proposent Navigala, une approche fondée sur la navigation pour la classification supervisée et l'appliquent à la reconnaissance de symboles bruités. La majorité des articles sur l'utilisation de treillis de Galois pour la classification sélectionnent d'abord les concepts qui encodent des informations pertinentes. Une autre approche fondée sur la navigation réalise une classification en naviguant à travers le treillis complet (de manière similaire à la navigation dans un arbre de classification) sans appliquer aucune opération de sélection.

D'autres sujets, qui reçoivent un peu moins d'attention mais deviennent populaire depuis peu, sont la fouille de services Web (Azmeh *et al.*, 2008; Chollet *et al.*, 2012) et la fouille de réseaux sociaux (Cuvelier & Aufaure, 2011; Elzinga *et al.*, 2012). Plusieurs de ces articles décrivent une recherche de haute qualité qui a abouti à des systèmes pratiques qui sont disponibles pour les utilisateurs finaux. Aussi, la fouille de logiciel (Molloy *et al.*, 2008; Wermelinger *et al.*, 2009) est un sujet très populaire dans la communauté. Initialement, l'attention a été principalement consacrée à la fouille de codes source statiques mais au fil des années plus d'articles sur l'application de l'AFC à l'analyse de codes dynamiques sont apparus dans la littérature. Un autre domaine d'applications de l'AFC concerne les sciences de la vie. En biologie, beaucoup d'articles tels que (Choi *et al.*, 2008; Kaytoue *et al.*, 2011; Motameny *et al.*, 2008) sont consacrés à l'analyse de données d'expression génique. Les auteurs n'utilisent pas seulement des structures d'AFC de base telles qu'un treillis de concepts, mais aussi des descriptions plus complexes d'AFC incluant les dits "pattern structures". En médecine l'AFC est appliquée

entre autres, aux données de séries temporelles, aux données de questionnaires et aux données textuelles, combinée à d'autres techniques telles que la fouille de motifs séquentiels, l'élagage fondée sur la stabilité, des modèles de Markov cachés, etc. Plusieurs articles tels que (Messai *et al.*, 2011; Rouane-Hacene *et al.*, 2009; Villerd *et al.*, 2010; Egho *et al.*, 2011) décrivent des applications réelles avec des résultats réels impactant la pratique quotidienne des soins de santé. En chimie l'AFC est principalement utilisée pour l'analyse de relations structure-activité de composants chimiques et en particulier pour l'identification de sous-graphes moléculaires possédant un certain profile d'activité (Blinova *et al.*, 2003; Lounkine *et al.*, 2008; Stumpfe *et al.*, 2011). Poelmans et al. terminent leur étude en abordant l'impact significatif de l'AFC en ingénierie d'ontologies, fusion d'ontologies et gestion de la qualité d'ontologies. Plusieurs applications matures de l'AFC, comme celles décrites dans (Jiang *et al.*, 2003; Soon & Kuhn, 2004; Cimiano *et al.*, 2005; Xu *et al.*, 2006), pour la construction d'ontologies dans différents domaines ont été étudiées.

4.5 Conclusion

Dans ce Chapitre, nous avons présenté les notions relatives à l'analyse formelle de concepts, telles que la notion de concept d'un contexte formel, la notion de treillis de concepts et la notion de règle d'association. Les divers domaines d'application de l'AFC connus à ce jour, ont également été présentés. L'analyse formelle de concepts fait partie des techniques largement utilisées en recherche d'images et de données de manière générale. A notre connaissance, aucune approche utilisant des notions relatives à l'AFC pour la localisation et l'extraction d'informations textuelles au sein d'images de documents n'est mentionnée dans la littérature.
Dans la Partie II nous présentons nos contributions dans le cadre de cette thèse. En particulier, l'approche présentée dans le Chapitre 6 est fondée sur des régions prototypes et des chemins déterminés à partir du treillis de concepts d'un contexte formel convenablement construit.

Deuxième partie

Contributions

Chapitre 5

Extraction d'informations textuelles au sein d'images de factures basée sur la décomposition quadtree

Sommaire

5.1 La décomposition quadtree 104
5.2 Notre approche de décomposition quadtree pour l'extraction d'informations textuelles au sein d'images de factures . 106
5.3 Évaluation expérimentale 109
5.4 Conclusion 113

5.1 La décomposition quadtree

La décomposition quadtree initialement développée par Finkel et Bentley (Finkel & Bentley, 1974) est une méthode de décomposition hiérarchique en cellules d'un environnement pouvant être représenté par une carte à deux dimensions. C'est une méthode de discrétisation d'un espace de recherche introduite en robotique pour la planification de trajectoire dans un espace contenant des obstacles (Barraquand & Latombe, 1991). La méthode consiste à diviser l'espace en quatre régions identiques. Pour chaque région, si celle-ci contient à la fois de l'espace libre et des obstacles, elle est de nouveau divisée en quatre régions égales. La division continue ainsi récursivement pour ces régions dites mixtes, jusqu'à atteindre un nombre maximum de divisions. Nous pouvons noter que les régions contenant uniquement soit de l'espace libre soit un obstacle ne sont plus divisées. Parallèlement au processus de subdivisions un arbre dont chaque nœud correspond à une région est construit. Les nœuds sont étiquetés de sorte que chaque nœud indique la nature de la région correspondante, selon que la région ne contienne que de l'espace libre ou est sans espace libre ou mixte. Pour un nœud n correspondant à une région r, les descendants de n correspondent aux régions résultantes de la subdivision de r. Un tel arbre est montré dans la Figure 5.1. L'algorithme A* (A-star) proposé par Hart (Hart *et al.*, 1968) est une implémentation de cette méthode. Cet algorithme est en fait une extension de l'algorithme de Dijkstra (Huijuan *et al.*, 2011).

La décomposition quadtree est aussi largement utilisée dans la littérature pour le traitement d'images et la recherche d'images (El-Qawasmeh, 2003; Manolopoulos *et al.*, 2005; Dagher & Taleb, 2014). Le principe appliqué dans ce cadre est de décomposer un espace à deux dimensions (définit au sein d'une image) en quatre régions initiales si une condition de décomposition est vérifiée. La décomposition de chacune des régions initiales continue jusqu'à ce qu'il ne reste plus de régions à diviser ou qu'un certain critère d'arrêt soit satisfait.
El-Qawasmeh (El-Qawasmeh, 2003) utilise la décomposition quadtree pour séparer une base de données d'images en plusieurs sous ensembles en y ajoutant plusieurs informations supplémentaires afin de faciliter la recherche d'images. Dans (Manolopoulos *et al.*, 2005), la structure quadtree est utilisée dans plusieurs approches de recherche d'images fondées sur le contenu pour la capture de la composition spatiale des éléments des images, comme les couleurs, les textures, les formes. Dans (Dagher & Taleb, 2014) la décomposition quadtree est combinée à une transformée en ondelettes et est appliquée à une image pour la suppression de bruit. L'utilisation de la structure quadtree pour la segmentation d'images dans (Atzori *et al.*, 2001) et dans (Minaee *et al.*, 2014) est un autre exemple d'application. Des algorithmes tels que DISIMA (Oria *et al.*, 2001) ou IKONA (Boujemaa *et al.*, 2001), implémentant la décomposition quadtree, sont également utilisés pour la recherche d'images fondée sur leur contenu. D'autres applications de la décomposition quadtree ou de l'utilisation la structure quadtree, telles que l'encodage de vidéos (Yuan *et al.*, 2012) et la classification de documents de type formulaire (Perea & López, 2004), peuvent être trouvées dans la littérature.
Dans ce mémoire, les images de factures sont utilisées comme cas d'étude. Les factures contiennent des informations obligatoires (numérotation de la facture, identifiant unique de la société émettrice, montant hors taxe, taux de tva, montant de tva, montant net, la mention "facture", date d'émission de la facture, etc.) qui, selon l'émetteur peuvent se trouver à des endroits différents dans le document. Néanmoins, pour un ensemble d'émetteurs différents des similitudes peuvent apparaitre localement pour une ou plusieurs informations données. Prenons l'exemple des informations concernant le montant d'une prestation de service : montant hors taxte, montant tva, montant net. Dans le système français, ces informations sont généralement positionnées en bas à droite des factures. De la même manière, l'identité de la société émettrice apparait généralement en haut à gauche des factures. A l'inverse, pour d'autres informations telles que la date d'émission et le numéro de la facture, il est plus délicat d'établir une tendance car ces informations ne semblent pas avoir une localisation spécifique au sein des factures.
Dans ce Chapitre, nous décrivons notre approche de décomposition quadtree pour localiser et extraire des informations textuelles au sein d'images de factures. Les images de factures sont divisées en régions dans lesquelles nous tentons d'extraire une information textuelle donnée à l'aide d'un moteur d'OCR. Dans ce qui suit, une région est une zone rectangulaire définie au sein d'une image. A notre connaissance aucun usage de la décomposition quadtree pour cette tâche n'est mentionné dans la littérature. Notre méthode est présentée dans la Section 5.2. La Section 5.3 présente des résultats expérimentaux illustrant l'efficacité de notre méthode.

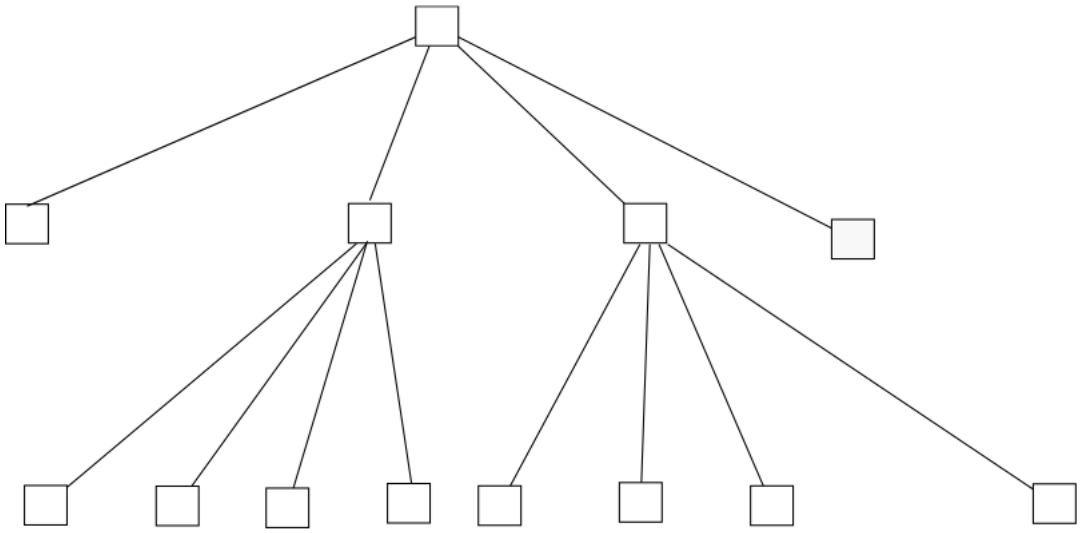


FIGURE 5.1 – Un quadtree.

5.2 Notre approche de décomposition quadtree pour l'extraction d'informations textuelles au sein d'images de factures

Dans cette Section nous considérons une information textuelle T, par exemple le montant net d'une facture et une image de facture I dans laquelle nous voulons localiser et extraire T. L'image I peut être obtenue à partir d'un appareil d'acquisition d'images numériques tel qu'un scanner. Pour localiser et extraire T au sein de I nous construisons une expression régulière E décrivant T. Par exemple, l'expression régulière décrivant le montant total en euro d'une facture peut s'écrire "(\bTOTAL\b) ?\d+ [,]\d+[€] ?". Les expressions régulières sont utiles pour trouver des chaines de caractères spécifiques dans du texte. Comme le souligne (Brauer *et al.*, 2011), c'est une technique intéressante pour extraire des informations pertinentes au sein de textes bruts dès lors que ces informations suivent un modèle syntaxique stricte. Nous verrons par la suite que ce critère n'est pas toujours respecté en ce qui concerne les informations qui peuvent être contenues au sein de factures.

Afin de transformer le texte contenu au sein des images de factures en texte brut nous utilisons le moteur d'OCR nommé Tesseract OCR [1]. Tesseract OCR est un moteur de reconnaissance de caractères open source. Il a été développé initialement par HP Labs entre 1985 et 1995. Il appartient à présent à Google. Tesseract OCR est produit sous la licence Apache 2.0 et est disponible sur SourceForge.net [2]. Nous avons mentionné, dans l'état de l'art de ce mémoire, les faiblesses de nombreuses solutions d'OCR qui ne reconnaissent que partiellement ou de manière incorrecte des images de documents. De plus, ces outils ne permettent pas l'extraction d'informations ciblées dans un format compréhensible par un programme informatique. Par exemple, un moteur d'OCR tel que Tesseract OCR n'est pas capable d'extraire uniquement le montant total présent au sein d'une image de facture à la demande. En effet, les moteurs d'OCR, tels que Tesseract OCR, sont développés pour extraire la masse entière de texte contenu dans

1. https ://github.com/tesseract-ocr/
2. https ://sourceforge.net/projects/tesseract-ocr/

une image de document sans distinction de la pertinence, ni de la nature de l'une par rapport aux autres.
La décomposition quadtree que nous adoptons consiste à subdiviser récursivement une image I en quatre régions rectangulaires identiques. Puis, une reconnaissance de caractères est réalisée sur chaque région afin d'en extraire le texte contenu. Un de nos critères d'arrêt de la décomposition d'une région est la découverte d'un élément textuel dans celle-ci, qui matche avec l'expression régulière E. Aussi, un second critère d'arrêt est le nombre maximum fixé de décomposition à effectuer. Dans le cas où aucune chaine de caractères ne matche, nous avons donc définit le nombre maximum e de décomposition à considérer durant le processus. Notre approche de décomposition consiste à :

1. décomposer l'image entière de document I en quatre rectangles égaux (r_1,r_2,r_3,r_4). Fixer une cellule, disons r_1 ;
2. tenter de trouver une chaine de caractères dans r_1 qui corresponde à l'expression régulière E ;
3. si aucune chaine de caractères correspondant à E n'est trouvée, nous fixons une deuxième région, disons r_2. Nous ré-appliquons l'étape de recherche de chaine de caractères qui correspond ;
4. les étapes consistant à fixer une région non visitée r_i et à trouver une chaine de caractères matchant avec E dans r_i sont répétées jusqu'à ce que :
   (a) soit les quatre régions sont visitées et aucune chaine de caractères valide n'est trouvée,
   (b) soit une chaine de caractères valide est trouvée et toutes les régions ont été visitées ou pas.

   Si le cas (a) est rencontré, nous décomposons chacune des régions précédentes en quatre sous-régions (r_{j1},r_{j2},r_{j3},r_{j4}). La variable j correspond à l'indice de la région originale. Les étapes de l'algorithme sont répétées pour chaque sous-région. Si le cas (b) est rencontré la décomposition est arrêtée.
5. Si le nombre maximum e de décomposition est atteint avant qu'une chaine de caractères valide ne soit trouvée la décomposition est arrêtée.

L'approche présentée ci-dessus, pour l'extraction d'une information textuelle, peut être généralisée pour l'extraction d'un ensemble de n informations textuelles. Pour l'ensemble des informations textuelles à extraire, l'ensemble $\mathcal{E} = \{E_1, \cdots, E_n\}$ des expressions régulières correspondantes doit être déterminé. Notre approche de décomposition consiste alors à :

1. décomposer l'image entière de document I en quatre rectangles égaux (r_1,r_2,r_3,r_4). Fixer une cellule, disons r_1 ;
2. pour chaque $E_i \in \mathcal{E}$, où $i = 1, \cdots, n$, tenter de trouver une chaine de caractères dans r_1 qui corresponde à l'expression régulière E_i ;
3. si il existe une expression régulière E_i pour laquelle aucune chaine de caractères correspondant n'est trouvée, nous fixons une deuxième région, disons r_2. Nous ré-appliquons l'étape de recherche de chaines de caractères qui correspondent ;
4. les étapes consistant à fixer une région non visitée r_i et à trouver une chaine de caractères matchant avec E_i dans r_i sont répétées jusqu'à ce que :
   (a) soit les quatre régions sont visitées et il existe encore au moins une chaine de caractères valide à trouver,

(b) soit une chaine de caractères valide est trouvée pour chaque $E_i \in \mathcal{E}$ et toutes les régions ont été visitées ou pas.

Si le cas (a) est rencontré, nous décomposons chacune des régions précédentes en quatre sous-régions (r_{j1},r_{j2},r_{j3},r_{j4}). La variable j correspond à l'indice de la région originale. Les étapes de l'algorithme sont répétées pour chaque sous-région. Si le cas (b) est rencontré la décomposition est arrêtée.

5. Si le nombre maximum e de décomposition est atteint avant qu'une chaine de caractères valide ne soit trouvée la décomposition est arrêtée.

Le pseudo-code de l'algorithme correspondant à notre approche de décomposition est présenté dans l'Algorithme 2.

Entrées:
e : le nombre maximum de décomposition à effectuer
I : une image
$\mathcal{E}$=$\{E_1, \cdots, E_n\}$: un ensemble d'expressions régulières correspondant à un ensemble d'informations textuelles à extraire au de I
Sorties:
S : un ensemble de chaines de caractères

```
1  début
2      décomposer (I,ℰ,e) :
3      si e>0 alors
4          couper I en quatre sous-régions r1, r2 , r3 , r4 ;
5          pour chaque ri faire
6              pour chaque Ei faire
7                  s = OCR(ri) ;
8                  si match(s,Ei) alors
9                      retirer Ei de ℰ ;
10                     insérer s dans S ;
11                 fin
12             fin
13         fin
14         e = e-1 ;
15         si cardinal(S)<cardinal(ℰ) alors
16             pour chaque ri faire
17                 décomposer(ri,ℰ,e) ;
18             fin
19         fin
20     fin
21 fin
```

Algorithm 2: Algorithme de décomposition quadtree pour l'extraction d'informations textuelles au sein d'images

Pour pallier la possibilité que les divisions successives de I en régions séparent le contenu textuel et le dispersent en plusieurs parties dans différentes régions, nous introduisons une constante s qui est ajoutée à chaque paramètre des dimensions (hauteur et largeur) d'une région avant chaque division. Par exemple lors d'un premier décou-

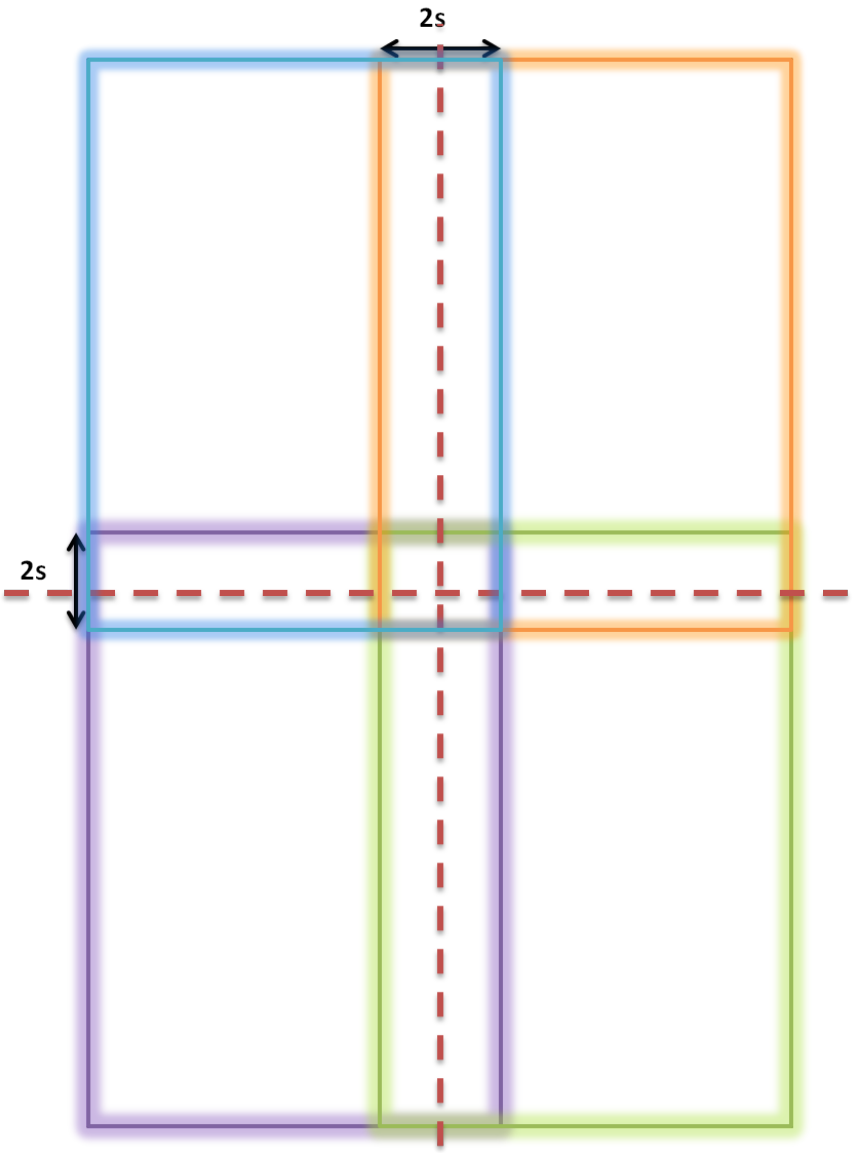


FIGURE 5.2 – Principe de décomposition en quatre régions adopté dans l'algorithme proposé.

page en quatre sous-régions d'une image I de dimension 1648px de largeur et 2336px de hauteur[3], le principe de la décomposition veut que l'image soit découpée en quatre rectangles égaux soit quatre rectangles de dimension 824px de largeur et 1168px de hauteur. Dans nos expérimentations nous découperons quatre rectangles de dimension (824px+s) de largeur et (1168px+s) de hauteur (voir Figure 5.2). De cette façon, nous proposons un algorithme plus efficace lors de la phase de reconnaissance de caractères.

5.3 Évaluation expérimentale

Un programme JAVA mettant en œuvre notre algorithme de décomposition d'une image de facture a été développé. Dans ce programme, pour chaque région, ses dimen-

3. dimension en pixel d'une image au format A4 et avec une résolution de 200dpi

sions et ses informations de localisation sont stockées : hauteur et largeur en pixels, les coordonnées x et y du coin supérieur gauche de la région dans l'image originale. Une liste S constituée des chaines de caractères qui matchent avec l'information à extraire est également construite. Puis, un vecteur constitué de ces paramètres (dimensions, point supérieur gauche, liste de chaines de caractères) est construit. Ce vecteur est utilisé en tant que descripteur de la région correspondante à un nœud du quadtree à construire. Un nœud dont le descripteur contient une liste non vide d'éléments est marqué comme plein, dans le cas contraire le nœud est marqué vide. La Figure 5.3 montre un exemple de quadtree obtenu. Un nœud plein est indiqué par un carré noir et un nœud vide par un carré blanc.

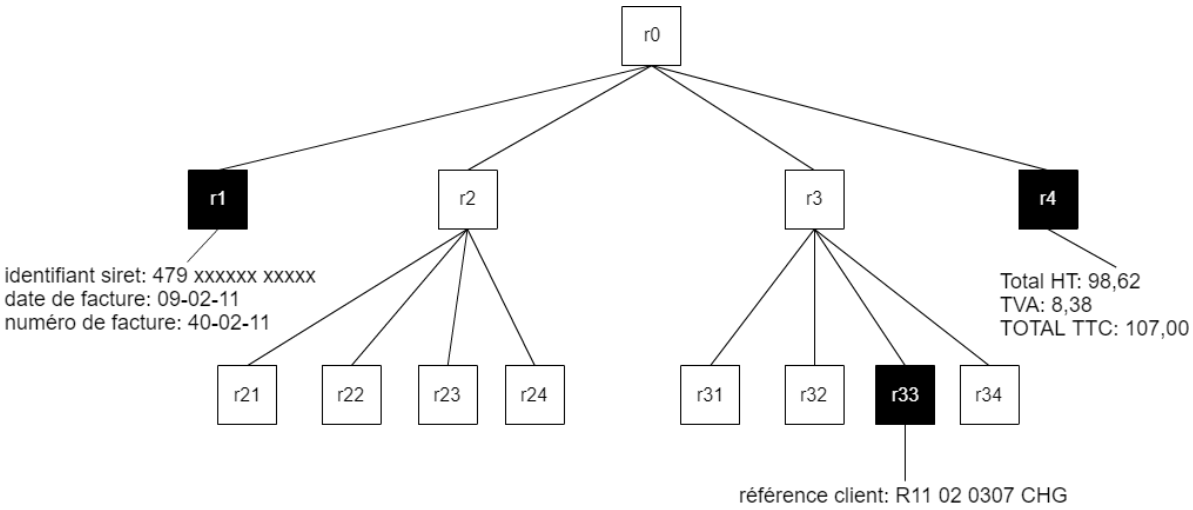


FIGURE 5.3 – Exemple de quadtree obtenu en appliquant notre algorithme sur une image de facture.

Nous avons expérimenté notre méthode sur un corpus d'images de factures émises par différents prestataires de services. Ces images de factures ont été obtenues grâce à un matériel d'acquisition de type scanner. Les images sont au format JPEG. Elles proviennent des archives de GAA. Elles sont issues de différentes sociétés de services telles que des sociétés de remorquage, de dépannage, de taxi et de location de véhicules. Des exemples sont disponibles en Annexe A, B, C, D. Elles présentent donc des mises en page hétérogènes. Les images de factures ont été sélectionnées de manière aléatoire à partir de l'ensemble des archives de GAA. Le corpus contient 65 images de factures. La distribution originale des factures au sein du corpus est présentée dans le Tableau 5.2. Le programme JAVA développé prend en entrée le chemin absolu d'un répertoire contenant toutes les images du corpus. Le programme produit en sortie des fichiers XML. Chaque fichier XML reprend pour chaque image traitée le descripteur décrit plus haut et obtenu par l'algorithme présenté ci-dessus. Un exemple de fichier XML pouvant être obtenu est présenté ci-dessous. Dans ce fichier, nous pouvons voir, qu'au sein de la facture "f-234" (ligne 1) trois informations ont été trouvées au sein de deux sous-régions distinctes : la région r_1 de l'image initiale (balises <region1></region1> des lignes 3 et 10) contient les informations date facture et référence client ; l'information date facture contient la chaine de caractères "13-04-11" (ligne 7) et l'information référence client contient la chaine de caractères "R 1104 0753 X04" ; la région r_4 (balises <region4></region4> des lignes 11 et 17 contient l'information montant TTC et la chaine de caractère correspondante extraite est "107.00" (ligne 15). Les coordonnées (x,y) du point supérieur gauche (lignes

4 et 12) et les dimensions (largeur et hauteur en pixels) (lignes 5 et 13) de chaque région sont également relevées.

```
1<facture f-234>
2 <regions>
3 <region1>
4 <origine x=0,y=0/>
5 <dimension largeur=927,hauteur=1269/>
6 <information>
7 <dateFacture> 13-04-11 </dateFacture>
8 <referenceClient> R 1104 0753 X04 </referenceClient>
9 </information>
10 </region1>
11 <region4>
12 <origine x=727,y=1069/>
13 <dimension largeur=927,hauteur=1269/>
14 <information>
15 <totalTTC> 107.00 </totalTTC>
16 </information>
17 </region4>
18 </regions>
19</facture f-234>
```

FIGURE 5.4 – Exemple de fichier XML obtenu en sortie de notre algorithme de décomposition.

Dans le cadre de notre expérimentation nous avons fixé la valeur de s à 100px. Comme évoqué plus tôt, s est utilisé pour élargir les surfaces des régions à découper afin de prendre en compte les cas où les informations recherchées chevauchent des régions adjacentes. Par ailleurs, nous avons fixé le nombre maximal e de décomposition d'une région issue de la première phase de décomposition à 3. Différentes valeurs de s et de e ont été testées. En outre, nous avons observé que pour des valeurs supérieures à 3 de e les régions obtenues étaient trop petites pour contenir des informations complètes vis à vis du type de documents du corpus. Par ailleurs, des valeurs plus petites de s ne permettent pas de récupérer les informations qui chevauchent des régions adjacentes. Des valeurs plus grandes diminuent les gains de reconnaissance de l'OCR sur les régions obtenues.
L'expérimentation consiste à extraire une liste L de cinq informations au sein du corpus d'images de factures. Ces informations sont :

- le montant total,
- la référence client,
- le numéro siret de la société émettrice de la facture,
- la date de la facture,
- numéro d'immatriculation.

Pour extraire du texte nous avons choisi le moteur d'OCR Tesseract OCR que nous avons mentionné plus tôt. Nous avons réalisé deux types de reconnaissance :

Information textuelle recherchée	Expression régulière correspondante
montant total	\d+[,] ?\d+[€] ?
référence client	[RGMYTPC]\s ?\d2\s ?\d2\s ?\d4\s ?[A-Z0-9]3\s ?[GH] ?
siret	[0-9]3\s ?[0-9]3\s ?[0-9]3\s ?[0-9]5)|([0-9]3\s ?[0-9]3\s ?[0-9]3
date facture	(DATE) ?(.)*\s ?\d2[/-]\d2[/-]\d2,4
immatriculation	(\d1,3\s ?- ?[A-Z]3\s ?- ?\d*)|([A-Z]2\s ?- ?\d3\s ?- ?[A-Z]2)

TABLE 5.1 – Liste des expressions régulières correspondant à chacune des cinq informations recherchées.

1. une reconnaissance des images entières,
2. une reconnaissance en appliquant notre algorithme mettant en œuvre la décomposition quadtree.

Pour les deux types d'extraction, le dénominateur commun est l'utilisation du même moteur d'OCR et la recherche de chaines de caractères valides se basant sur les mêmes expressions régulières. Par exemple, l'expression régulière utilisée pour décrire le montant total de la facture en euro est : "\d+[,] ?\d+[€] ?". Cette expression régulière est utilisée dans notre programme JAVA afin de trouver toutes les chaines de caractères extraites des images de documents qui matchent et qui peuvent donc potentiellement correspondre à l'information recherchée qu'elle représente, c'est à dire un montant total. Le Tableau 5.1 liste toutes les expressions régulières utilisées dans notre programme. Les listes de chaines de caractères valides extraites sont stockées dans des fichiers XML. Nous considérons deux mesures pour évaluer notre approche :

— la proportion d'informations détectées parmi le nombre total d'informations à extraire (rappel),
— la proportion d'informations correctes parmi le nombre total d'informations détectées (précision).

L'évaluation du nombre d'informations détectées est basée sur le comptage du nombre de chaines de caractères qui matchent avec une expression régulière E trouvées. Par ailleurs, une information est considérée comme correcte si une chaine de caractères détectée correspond exactement au visuel de l'information recherchée dans l'image originale. Nous observons que nous obtenons un nombre moyen équivalent d'informations détectées pour les deux types de reconnaissance expérimentés. En effet, pour le premier type de reconnaissance, 90% des informations recherchées sont détectées. Dans le deuxième type de reconnaissance (notre méthode) la proportion d'informations détectées est sensiblement la même (94%). Cependant, la proportion d'informations correctes relevée, dans le cas de la reconnaissance appliquant notre algorithme de décomposition, est remarquablement plus élevée que dans le cas de la reconnaissance sans décomposition. En effet, après évaluation, la précision obtenue par le moteur d'OCR choisi est de 53% lorsqu'il s'agit de reconnaitre les images de factures entières (sans décomposition), tandis que la précision obtenue pour la reconnaissance des images de factures en appliquant notre algorithme de décomposition quadtree est de 87%. Nous expliquons notre gain concernant la proportion d'informations correctement extraites par deux faits : d'une part le découpage a un effet de zoom sur une partie précise de l'image à reconnaitre. En effet, les éventuels bruits à cause d'une résolution et d'une qualité de l'image trop faible sont amoindris. D'autre part, la mise en page d'une région d'image est simplifiée par rap-

	nombre de prestataires	nombre de factures
remorquage	3	23
dépannage	2	12
location de voiture	3	18
taxi	2	12
total	10	65

TABLE 5.2 – Distribution originale du corpus d'images de factures de test.

	rappel	précision
images entières	90%	53%
images décomposées par notre méthode	94%	87%

TABLE 5.3 – Résultats

port à celle de l'image entière. Les moteurs d'OCR gratuits, tels que Tesseract OCR sont connus pour être sensibles à la présence de bruit au sein d'images. Leurs performances sont faibles par rapport à celles des moteurs de reconnaissance commerciaux. Ainsi, nous montrons que la décomposition effectuée permet d'obtenir des portions d'images à analyser, les unes indépendamment des autres, qui sont plus homogènes et simplifiées que l'image entière pour un moteur d'OCR tel que Tesseract OCR.

5.4 Conclusion

Dans ce Chapitre nous présentons notre méthode d'extraction d'informations textuelles au sein d'images de factures fondée sur la décomposition quadtree. La décomposition quadtree que nous adoptons consiste à subdiviser récursivement une image de factures en quatre régions rectangulaires identiques. Puis, une reconnaissance de caractères est réalisée sur chaque région afin d'en extraire le texte contenu. A partir d'un moteur d'OCR gratuit (présumé moins efficace que des logiciels commerciaux) notre méthode permet d'extraire des informations ciblées et caractérisées au sein d'images de factures avec une précision de 87%. Notre méthode d'extraction simplifie les traitements annexes tels que l'intégration automatique de ces informations dans un système d'information. Notre méthode de décomposition présente plusieurs avantages :

- aucune construction de modèle n'est requise ;
- le traitement d'images de factures de source et de mise en page hétérogènes est facilité ;
- elle est applicable à l'extraction d'informations textuelles au sein d'images de documents autres que des factures ; en effet on peut imaginer appliquer notre méthode à des documents tels que des bons de commande ou des devis.

Néanmoins, dans l'algorithme que nous présentons la détermination empirique des paramètres e et s est un inconvénient. En effet, ces paramètres dépendent des images à traiter et plusieurs phases d'entrainement pour tester différentes valeurs de ces paramètres sont alors nécessaires.
Par ailleurs, l'utilisation d'expressions régulières afin d'établir une correspondance entre des chaines de caractères extraites et des informations recherchées est une limite de la

méthode. En effet, les cinq informations prises en compte dans nos expérimentations sont assez facilement représentables avec des expressions régulières. Par exemple, il est facile d'identifier qu'un numéro d'immatriculation de véhicule est constituée (dans le système français) de deux lettres, de trois chiffres, puis de deux lettres, avec la présence éventuelle d'espace ou de tiret entre chaque séquence de chiffres ou de lettres. De plus, plus le formatage est précis, plus la probabilité de rencontrer une autre information ayant le même format diminue et de ce fait la probabilité d'extraire une information incorrecte est plus faible également. Toutefois, pour d'autres informations qu'il serait pertinent d'extraire, ce type de représentation peut devenir complexe. Par exemple, le numéro de facture est une information qui n'obéit à aucune règle de formatage. Ainsi, les sociétés émettrices de factures sont libres de choisir le format d'écriture de cette information. Dans ce cas, l'expression régulière correspondante devient lourde voire impossible à établir. Le risque est alors d'extraire au sein des images des informations incorrectes. La méthode de décomposition présentée dans ce Chapitre repose sur le principe diviser pour régner. Bien que notre méthode montre une certaine efficacité, elle nécessite le parcours de toute la surface de l'image alors que les informations à extraire ne se trouvent que dans quelques régions précises.
Dans le Chapitre suivant nous présentons un système plus élaboré qui :

- ne nécessite pas l'établissement d'expressions régulières pour l'extraction de tous les types d'informations ;
- ne nécessite pas le parcours de toute la surface des images ;
- ne nécessite pas l'estimation empirique de paramètres d'exécution.

Dans le Chapitre suivant nous présentons notre système de localisation et d'extraction d'informations textuelles données, au sein d'images de factures. Ce système tente de pallier aux inconvénients de l'approche présentée dans ce Chapitre.

Publications
– Pitou, Cynthia, & Diatta, Jean. 2014. Localisation d'informations textuelles basée sur la décomposition quadtree. *In : 21ième Rencontres de la Société Francophone de Classification, 105–110.* – Pitou, Cynthia, & Diatta, Jean. 2016. Textual Information Localization and Retrieval in Document Images Based on Quadtree Quadtree Decomposition. *Analysis of Large and Complex Data, 71–78. Springer.*

Chapitre 6

Système d'extraction d'informations textuelles au sein d'images de factures

Sommaire

6.1 Création du jeu de données synthétiques 117
6.2 Classification des données synthétiques et détermination des régions prototypes 123
6.2.1 Classification des données synthétiques 124
6.2.2 Détermination des régions prototypes 127
6.3 Détermination de chemins pour naviguer au sein des régions prototypes à partir d'un treillis de concepts 128
6.3.1 Construction du treillis de concepts 128
6.3.2 Détermination de chemins à partir du treillis de concepts . . 129
6.4 Traitement d'une image de facture inconnue fondé sur des modèles incrémentaux 135
6.4.1 Mise à jour de l'ensemble de régions prototypes 136
6.4.2 Mise à jour des chemins pour naviguer au sein de l'ensemble des régions prototypes 139
6.5 Conclusion 142

Dans ce Chapitre nous présentons notre système d'extraction d'informations textuelles au sein d'images de factures. Le système se nomme **PReTIE** (***P**rototype **Re**gions based **T**extual **I**nformation **E**xtraction*) et consiste en un ensemble de régions prototypes et de chemins pour parcourir ces régions prototypes. Les étapes principales du système que nous présentons sont schématisées dans la Figure 6.1. Le système est composé de cinq étapes :

1. Construction d'un jeu de données synthétiques à partir d'images de factures réelles contenant les informations d'intérêts.
2. Partitionnement des données produites et détermination de régions prototypes à partir de la partition obtenue.
3. Détermination de chemins pour parcourir les régions prototypes, à partir du treillis de concepts d'un contexte formel convenablement construit.
4. Extraction d'informations textuelles au sein d'une image à l'aide des régions prototypes et des chemins pour les parcourir.

5. Mise à jour du système de manière incrémentale suite à l'insertion de nouvelles données par l'utilisateur.

La première étape consiste à construire un jeu de données synthétiques à partir d'images de documents synthétiques générées par un programme informatique.
La deuxième étape consiste à partitionner le jeu de données synthétiques obtenu à l'étape précédente puis à déterminer des régions prototypes à partir des classes de la partition obtenue.
La troisième étape consiste à déterminer des chemins pour naviguer dans l'ensemble des régions prototypes de manière efficace, à partir du treillis de concepts d'un contexte formel convenablement construit.
La quatrième étape consiste à extraire un ensemble d'informations textuelles ciblées au sein d'une image candidate, à partir des régions prototypes et des chemins obtenus à l'étape précédente.
Enfin, dans le cas où le système ne parvient pas à traiter une image candidate inconnue, l'utilisateur est invité à renseigner dans le système les données relatives à une information à extraire. L'insertion de ces nouvelles données déclenche la cinquième étape, qui consiste à effectuer une mise à jour incrémentale des régions prototypes existantes et des chemins existants.
Dans ce Chapitre, nous présentons la construction des différents modèles impliqués dans les étapes 1, 2, 3 et 5. La tâche d'extraction d'informations textuelles est détaillée dans le Chapitre suivant. La Section 6.1 présente l'étape de construction du jeu de données synthétiques. Le partitionnement du jeu de données construit ainsi que la détermination des régions prototypes à partir de la partition obtenue sont présentés dans la Section 6.2. La Section 6.3 présente l'étape 3 qui consiste à déterminer des chemins pour parcourir les régions prototypes, à partir du treillis de concepts d'un contexte formel convenablement construit. La mise à jour incrémentale du système pour la prise en compte d'une nouvelle donnée est présentée dans la Section 6.4. Le Chapitre se termine par une discussion.

6.1 Création du jeu de données synthétiques

Nous disposons d'un corpus de 1270 factures réelles numérisées, extraites des archives de GAA. Chaque image numérique représente une facture constituée d'une page au format A4 en couleur ou non. Les factures proviennent de 18 prestataires de service dont l'activité est le remorquage et le dépannage automobile. Le Tableau 6.1 présente la distribution originale du corpus. Les informations que nous cherchons à extraire sont :

prestataire	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
nb images	30	32	34	36	38	40	50	50	63	69	74	81	92

prestataire	P14	P15	P16	P17	P18	TOTAL
nb images	95	65	112	130	179	1270

TABLE 6.1 – Distribution originale du corpus d'images de factures réelles.

— I1 : le numéro de facture
— I2 : la date de facture

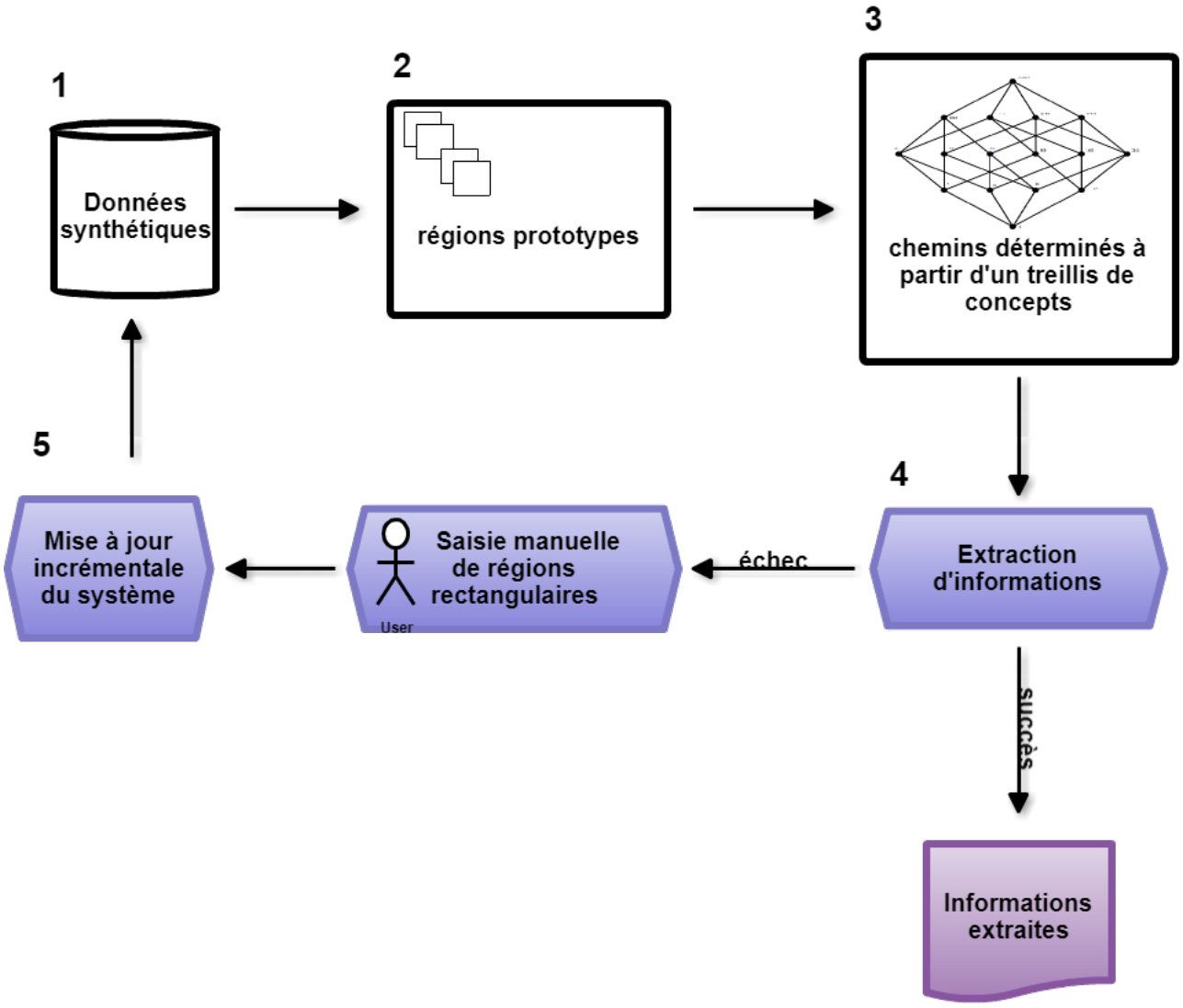


FIGURE 6.1 – Vue d'ensemble du système d'extraction d'informations textuelles au sein d'images de documents.

— I3 : une référence client
— I4 : le numéro d'immatriculation du véhicule assisté
— I5 : l'identifiant siret du prestataire de service.

Pour un même prestataire, la disposition des informations au sein de ses factures reste approximativement la même d'une facture à une autre. Des différences peuvent apparaitre si des distorsions sont introduites au sein des images de factures lors de la phase d'acquisition numérique par un scanner. Pour deux prestataires différents, il est possible, qu'une même information (la référence client par exemple) se trouve à des positions différentes. Dans ce contexte, nous cherchons à regrouper les factures pour lesquelles une information donnée possède des localisations proches (selon une mesure de distance donnée) au sein de ces factures, quel que soit l'émetteur.
Au sein des factures réelles, chaque information considérée est localisée dans une région délimitée par une zone rectangulaire définie autour du contenu textuel de cette information. Une région est représentée par les coordonnées (x, y) en pixels du point supérieur gauche et (z, t) du point inférieur droit du rectangle correspondant. Initialement, à partir des images de factures réelles nous ne disposons pas des descriptions (x, y, z, t) des régions. C'est pourquoi, nous avons développé un programme informatique doté d'une interface graphique qui permet de : (i) reproduire manuellement et approximativement la position d'une information d'intérêt contenue dans une image de facture réelle ; (ii) générer en sortie un nombre $n >= 1$ fixé d'images de factures synthétiques au format JPEG. En observant une image de facture réelle, l'interface graphique de notre programme permet de dessiner une région rectangulaire autour d'une information textuelle d'intérêt. Ensuite, le programme stocke dans une base de données les coordonnées (x, y, z, t) de la région rectangulaire indiquée par l'utilisateur. Pour chaque image synthétique générée, le type d'information (immatriculation, date de facture, numéro de facture, etc.) et les coordonnées des régions correspondantes sont sauvegardées en base de données. Une image de facture synthétique, ainsi générée, est une image d'une seule page en noir et blanc au format A4. La Figure 6.2 montre l'interface principale de notre programme. Cette interface est divisée en trois zones :

— zone 1 (Figure 6.3) : une zone principale dans laquelle l'utilisateur peut positionner manuellement un rectangle autour d'une information d'intérêt, à partir d'une image de facture réelle chargée dans l'interface ;
— zone 2 (Figure 6.4) : une zone présentant des champs de saisie et des boutons d'action pour : saisir le nom de l'image synthétique à générer, saisir le chemin absolu d'un emplacement où sera stockée l'image synthétique, saisir le nombre n d'images synthétiques à générer, charger une nouvelle image de facture réelle, valider et procéder à la création automatique de n images de factures synthétiques, réinitialiser la zone principale en supprimant tous les rectangles existants.
— zone 3 (Figure 6.5) : une zone permettant de saisir, pour chaque rectangle dessiné, quel est le type d'information correspondant (immatriculation, date de facture, numéro de facture, etc.).

Par exemple, à partir d'une image de facture réelle, via notre interface de génération automatique d'images de factures synthétiques, il est possible de générer 1000 images de factures synthétiques nommées "f-p1-1.jpg", $\cdots$, "f-p1-1000.jpg", contenant les informations I1 à I5 et stockées à l'emplacement "C :\", en suivant les étapes ci-dessous :

1. charger une nouvelle image de facture réelle ;

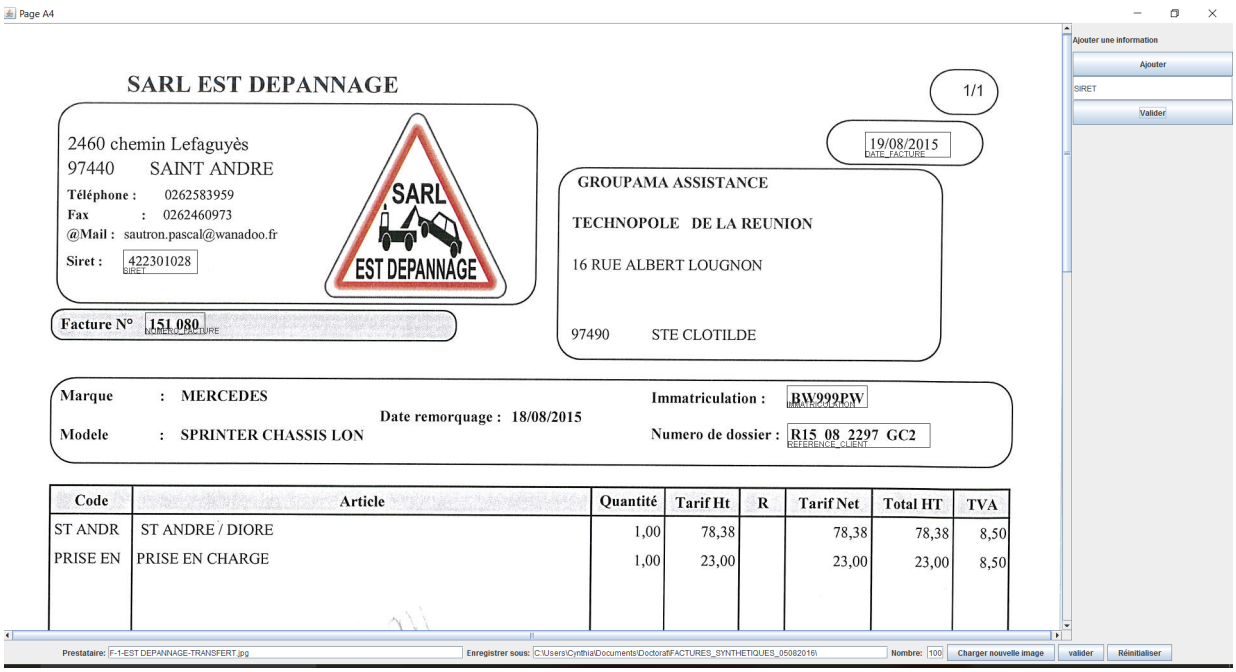


FIGURE 6.2 – Interface graphique du programme JAVA de création d'images de factures synthétiques.

2. saisir "f-p1-1.jpg" dans le champ **Prestataire** ;
3. saisir "C :\" dans le champ **Enregistrer sous** ;
4. saisir le nombre n=1000 dans le champ **nombre** ;
5. répéter pour chaque information I1 à I5 :
   (a) dessiner un rectangle autour de l'information Ii visible dans l'image réelle ;
   (b) saisir le type d'information contenu dans le rectangle (ex : NUMERO FACTURE) dans le champ de saisie se trouvant entre les boutons **Ajouter** et **Valider** de la zone 3 ;
   (c) cliquer sur **Valider**, afin de valider le type d'information saisi.
6. cliquer sur le bouton **valider** présent dans la zone 2, afin de procéder à la génération automatique de n images synthétiques.

Lors de la génération automatique des images de factures synthétiques, celles-ci sont nommées automatiquement en ajoutant un numéro d'incrément au nom de fichier indiqué par l'utilisateur. Par ailleurs, les rectangles dessinés dans la zone principale sont complétés automatiquement par des informations (immatriculation, date de facture, numéro de facture, etc.) fictives par souci de confidentialité.
Au sein du corpus d'images de factures réelles nous avons observé que pour un lot de factures émises par le même prestataire, les positions d'une information donnée pouvaient varier. Ces variations peuvent être dues à une distorsion des images lors de la phase de numérisation avec un scanner, ou à une modification de la fonte, de la taille de police, ou de la longueur du texte. Afin de simuler au plus près les images de factures réelles du corpus, pour chaque image synthétique générée automatiquement, les positions des rectangles définis par l'utilisateur sont décalées de quelques pixels, de manière aléatoire dans une des quatre directions suivantes : gauche, droite, haut ou bas. Le nombre de

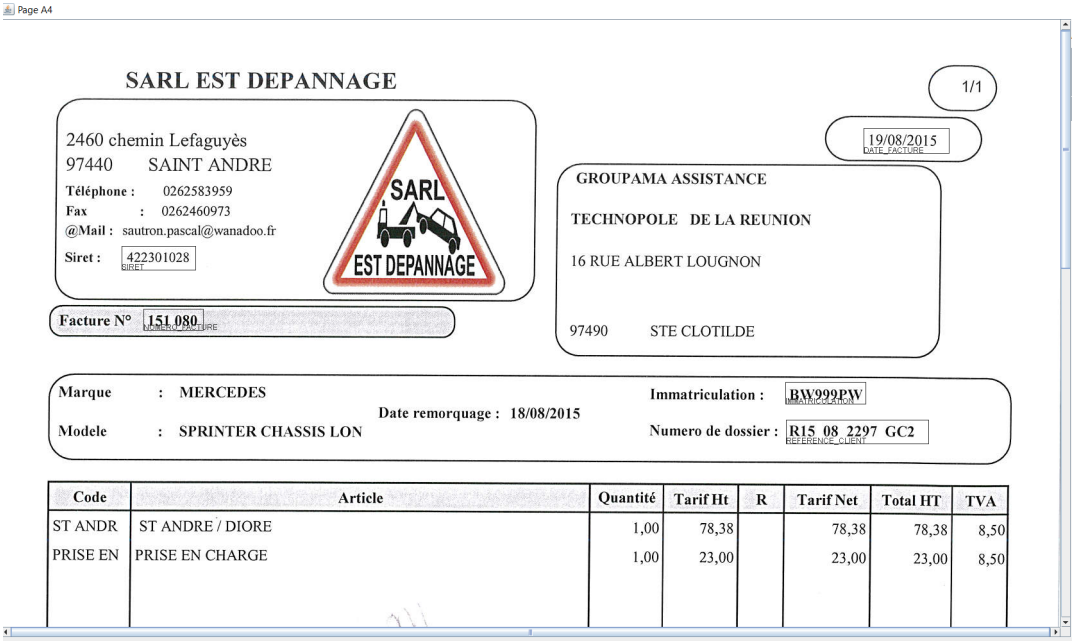


FIGURE 6.3 – Zone 1 de l'interface graphique de la Figure 6.2.

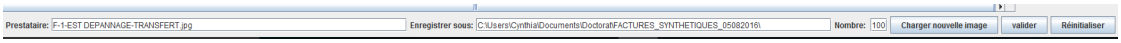
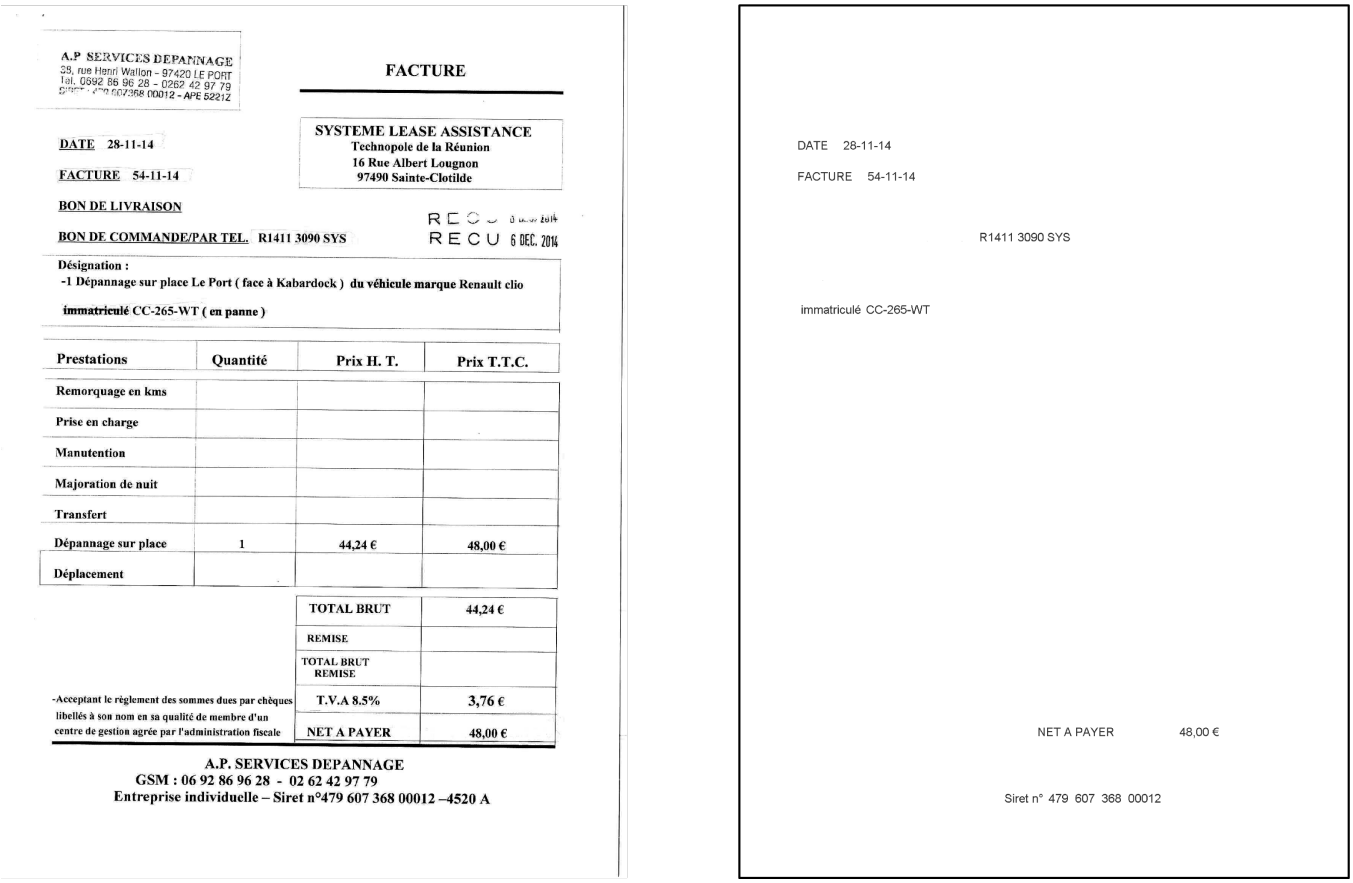


FIGURE 6.4 – Zone 2 de l'interface graphique de la Figure 6.2.

FIGURE 6.5 – Zone 3 de l'interface graphique de la Figure 6.2.



A.P SERVICES DEPANNAGE
39, rue Henri Wallon - 97420 LE PORT
Tel. 0692 86 96 28 – 0262 42 97 79
SIRET : 479 607368 00012 - APE 52212

FACTURE

SYSTEME LEASE ASSISTANCE
Technopole de la Réunion
16 Rue Albert Lougnon
97490 Sainte-Clotilde

DATE 28-11-14

FACTURE 54-11-14

BON DE LIVRAISON

RECU 6 DEC. 2014

BON DE COMMANDE/PAR TEL. R1411 3090 SYS

Désignation :
-1 Dépannage sur place Le Port (face à Kabardock) du véhicule marque Renault clio
immatriculé CC-265-WT (en panne)

Prestations	Quantité	Prix H. T.	Prix T.T.C.
Remorquage en kms			
Prise en charge			
Manutention			
Majoration de nuit			
Transfert			
Dépannage sur place	1	44,24 €	48,00 €
Déplacement			

	TOTAL BRUT	44,24 €
	REMISE	
	TOTAL BRUT REMISE	
-Acceptant le règlement des sommes dues par chèques libellés à son nom en sa qualité de membre d'un centre de gestion agrée par l'administration fiscale	T.V.A 8.5%	3,76 €
	NET A PAYER	48,00 €

A.P. SERVICES DEPANNAGE
GSM : 06 92 86 96 28 - 02 62 42 97 79
Entreprise individuelle – Siret n°479 607 368 00012 –4520 A

DATE 28-11-14

FACTURE 54-11-14

R1411 3090 SYS

immatriculé CC-265-WT

NET A PAYER 48,00 €

Siret n° 479 607 368 00012

FIGURE 6.6 – A droite une image de facture réelle. A gauche une image de facture synthétique générée par notre programme.

pixels à appliquer pour le déplacement est choisi de manière aléatoire dans un intervalle compris entre 5 et 100 pixels. Cela correspond à un déplacement de 3mm à 1cm environ dans une direction fixée (cas d'une image A4 numérisée en 200dpi).
Au sein d'une image de facture synthétique générée par notre programme, chaque rectangle (par conséquent chaque chaine de caractères qu'il contient) est caractérisé par le type d'information (immatriculation, date de facture, numéro de facture, etc.) indiqué par l'utilisateur. De cette manière, à partir de la base de données constituée, pour une image de facture synthétique, il est possible de retrouver l'ensemble des informations (type, contenu textuel et coordonnées du rectangle correspondant) qu'elle contient.
Pour chaque prestataire de service identifié au sein du corpus de factures réelles, à partir d'une facture réelle, faisant office de modèle, nous avons créé de manière automatique le nombre d'images synthétiques correspondant à sa distribution originale au sein du corpus. Nous avons ainsi créé un corpus d'images de factures synthétiques dont les contenus textuels et les coordonnées des régions qui les contiennent sont stockés dans une base de données. La Figure 6.6 montre à droite une image de facture réelle et à gauche une image de facture synthétique pouvant être générée par notre programme. De cette base de données est extraite un jeu de 1270 observations décrites selon 20 variables numériques (les 4 coordonnées (x, y, z, t) pour chacune des 5 informations I1 à I5).
Dans la section suivante nous décrivons l'étape de classification du jeu de données synthétiques obtenu.

6.2 Classification des données synthétiques et détermination des régions prototypes

A partir du jeu de données synthétiques évoqué dans la Section précédente, nous souhaitons déterminer des groupes homogènes d'images (les observations) pour lesquelles les positions des informations I1 à I5 sont proches selon une mesure de distance. Deux types de classification du jeu de données sont possibles :

- une classification supervisée ;
- une classification non supervisée.

Pour rappel, dans ce Chapitre nous présentons un système permettant d'extraire des informations textuelles au sein d'images de factures. Contrairement à une majorité de méthodes s'appuyant sur la construction de modèles (un modèle par prestataire) indiquant les localisations exactes de chacune des informations à extraire selon l'émetteur (Bartoli *et al.*, 2014; Medvet *et al.*, 2011; Belaïd *et al.*, 2011), notre système tente de généraliser les localisations des informations à extraire sans construire de modèles particuliers selon l'émetteur.
Notre système s'appuie sur l'hypothèse suivante : pour une information donnée, il existe un ensemble fini de régions globales dans lesquelles cette information peut être localisée au sein d'une image, quel que soit l'émetteur. Ainsi, nous montrons qu'il n'est pas nécessaire de connaitre pour chaque émetteur la position exacte de l'information considérée. Nous pensons qu'il suffit de connaitre les positions des régions globales. Ces régions globales sont des zones rectangulaires, définies au sein des images de factures, dans lesquelles il est possible de retrouver une information textuelle quel que soit l'émetteur. Nous appelons ces zones rectangulaires des *régions prototypes*. La détermination

des régions prototypes est présentée dans la Section 6.2.2. Ces régions prototypes sont déterminées à partir des classes obtenues par la classification du jeu de données synthétiques. La classification que nous adoptons est non supervisée. En effet, les régions prototypes que nous souhaitons déterminer sont inconnues et nous ne disposons pas de connaissance a priori sur celles-ci. La Section 6.2.1 présente la classification adoptée.

6.2.1 Classification des données synthétiques

L'objectif est de déterminer, pour un ensemble d'observations, des groupes homogènes dans lesquels les positions des informations I1 à I5 sont proches selon une mesure de distance.
Le jeu de données synthétiques construit dans la phase précédente peut être classé de deux manières : (a) selon une vue globale tenant compte de l'ensemble des 20 variables, ou (b) selon 5 vues indépendantes correspondant chacune à l'une des 5 informations I1 à I5 et tenant compte, pour chaque vue, des 4 variables associées. L'approche de classification selon une vue globale consiste à réaliser une classification traditionnelle du jeu de données synthétiques. L'approche de classification selon cinq vues indépendantes consiste à créer cinq jeux de données {D1, D2, D3, D4, D5} et à classer chaque jeu de données Di de manière indépendante. Les observations d'un jeu Di sont décrites par les quatre coordonnées des régions contenant l'information Ii. A titre d'exemple, une partie des jeux de données D1 relatif à l'information I1 et D5 relatif à l'information I5 est présentée dans les Tableaux 6.2 (a) et (b). Pour chacune des deux approches (selon une vue globale et selon cinq vues indépendantes) nous avons comparé les classes obtenues en appliquant plusieurs méthodes de classification non supervisée. Les méthodes de classification utilisées sont :

1. CAH (Classification ascendante hiérarchique) avec le lien de Ward ;
2. k-medoids muni de la distance Euclidienne ;
3. k-means muni de la distance Euclidienne.

	I1x	I1y	I1z	I1t
1	151	748	862	781
...				
84	130	725	683	758
...				

(a) D1

	I5x	I5y	I5z	I5t
1	1194	1959	1302	1992
...				
84	1476	1818	1575	1851
...				

(b) D5

TABLE 6.2 – Exemples de données du jeu de données D1 relatif à l'information I1 (a) et du jeu de données D5 relatif à l'information I5 (b).

Comme évoqué dans la Partie I Chapitre 3 Section 3.2.2, le choix du nombre optimal de classes est une question importante en classification non supervisée. Dans le cas de la CAH le choix du nombre de classes peut se faire a posteriori de la classification, à partir du dendrogramme obtenu, en le coupant à des niveaux différents, puis en comparant les classes obtenues par les différentes coupures. Dans le cas d'une classification avec k-means ou k-medoids, le nombre de classes est à fournir en paramètre de la méthode. Dans la pratique, une approche pour déterminer le nombre optimal k de classes consiste

à exécuter la méthode de classification pour plusieurs valeurs de k puis à comparer les partitions obtenues. C'est l'approche que nous adoptons pour la détermination du nombre k de classe dans le cadre de la classification du jeu de données synthétiques avec k-means et k-medoids.
Afin de déterminer la qualité des partitions obtenues par chacune des trois méthodes de classification, nous nous appuyons sur trois indices de qualité : PBM (Pakhira *et al.*, 2004), le Silhouette (silh.) (Kaufman & Rousseeuw, 1990a; Rousseeuw, 1987) et Calinksi-Harabasz (CH) (Caliński & Harabasz, 1974). Pour rappel, comme évoqué dans la Partie I Chapitre 3 Section 3.2.2, ces trois indices sont ceux qui obtiennent les meilleurs performances pour l'évaluation de la qualité d'une partition d'après l'étude de (Vendramin *et al.*, 2010). Pour chacune des trois méthodes de classification adoptées, des valeurs de k allant de 2 à 18 sont testées et pour chaque indice la valeur de k optimale est relevée. En résumé, pour chaque approche de classification (selon une vue globale et selon cinq vues indépendantes) et pour chaque méthode de classification adoptée, le protocole adopté est le suivant :

1. pour chaque valeur de k comprise entre 2 et 18 :
   (a) appliquer une méthode de classification,
   (b) mesurer la qualité des classes obtenues à l'aide des trois indices choisis,
   (c) relever les valeurs de chaque indice,
2. relever la valeur de k optimale selon chaque indice.

Les Tableaux 6.3 et 6.4 présente les valeurs de k optimales relevées en considérant chaque indice, respectivement dans le cadre de la classification selon une vue globale et de la classification selon cinq vues indépendantes.

	CAH	k-medoids	k-means
silh.	13	10	10
CH	13	10	10
PBM	9	9	9

Table 6.3 – Valeurs de k optimales relevées dans le cas de la classification avec CAH, k-means et k-medoids selon une vue globale.

Dans le cas de la classification selon une vue globale, le Tableau 6.3 montre que les trois indices ne s'accordent pas tous sur le même nombre de classes quelle que soit la méthode de classification utilisée. Néanmoins, deux des indices (Silhouette, Calinski-Harabasz) s'accordent sur le même nombre de classes pour chacune des méthodes de classification utilisée. En adoptant le principe du vote majoritaire, nous relevons qu'une valeur optimale de k pour la classification du jeu de données est soit k=10, soit k=13. Afin de départager les deux valeurs possibles de k, nous comparons les valeurs relevées pour chaque indice (Silhouette, Calinski-Harabasz) en appliquant chacune des méthodes de classification. Le Tableau 6.5 récapitule les valeurs de ces indices. Finalement, nous retenons la partition constituée de k=10 classes, qui obtient les valeurs d'indices les plus élevées.

Dans le cas de la classification selon cinq vues indépendantes, le Tableau 6.4 montre que, quelle que soit la méthode de classification utilisée, les trois indices s'accordent sur la même valeur de k=10 classes pour le jeu de données D1. Pour les autres jeu de

	D1 : I1	D2 : I2	D3 : I3	D4 : I4	D5 : I5
silh.	10	9	9	3	10
CH	10	10	10	2	10
PBM	10	2	10	2	9

(a) CAH

	D1 : I1	D2 : I2	D3 : I3	D4 : I4	D5 : I5
silh.	10	10	10	3	9
CH	10	10	9	3	8
PBM	10	13	10	3	13

(b) k-medoids

	D1 : I1	D2 : I2	D3 : I3	D4 : I4	D5 : I5
silh.	10	10	3	3	9
CH	10	7	10	6	10
PBM	10	7	6	8	11

(c) k-means

TABLE 6.4 – Valeurs de k optimales relevées dans le cas de la classification selon cinq vues indépendantes des jeux de données D1, D2, D3, D4 et D5 en appliquant les trois méthodes de classification : k-means, k-medoids et CAH.

	CAH	k-medoids	k-means
silh.	0.43	0.45	0.45
CH	5914	4906	7701

(a) k=13

	CAH	k-medoids	k-means
silh.	0.58	0.57	0.58
CH	15490	16967	19637

(b) k=10

TABLE 6.5 – Valeurs de Silhouette et de Calinski-Harabasz relevées pour une classification du jeu de données selon une vue globale en k=10 et k=13 classes.

données, il n'y a pas d'accord immédiat entre les indices. Pour rappel, dans le cas de la classification selon une vue globale nous avons tout d'abord pré-sélectionné des valeurs de k possibles en appliquant un vote majoritaire, avant de comparer les valeurs relevées par chaque indice, afin de départager plusieurs valeurs de k. Le même principe est adopté, dans le cas de classification selon cinq vues indépendantes, afin de déterminer des valeurs de k optimales. Les valeurs de k retenues pour chaque vue sont présentées dans le Tableau 6.6. Les valeurs des indices de qualité des partitions obtenues avec k-means pour les valeurs finales de k du Tableau 6.6, sont présentées dans le Tableau 6.7. L'étude des deux approches (classification selon une vue globale et selon cinq vues indépendantes) a montré que les partitions obtenues dans le cas d'une classification selon cinq vues indépendantes sont de meilleure qualité. Nous retenons donc le partitionnement selon cinq vues indépendantes. Nous observons également que, dans le cas de la classification selon cinq vues indépendantes, les partitions obtenues en appliquant la méthode k-means sont de meilleure qualité que celles obtenues avec les deux autres méthodes de classification. Finalement, une fois la valeur de k fixée pour chaque jeu de données Di, nous avons adopté un partitionnement avec k-means et une initialisation préalable des centres de classe avec les centres obtenus par CAH.

	D1 : Info I1	D2 : Info I2	D3 : Info I3	D4 : Info I4	D5 : Info I5
k	10	10	10	3	10

Table 6.6 – Nombre de classes retenu pour le partitionnement des jeux de données D1, D2, D3, D4 et D5.

	D1 : Info I1	D2 : Info I2	D3 : Info I3	D4 : Info I4	D5 : Info I5
Silh.	0.64	0.68	0.61	0.55	0.73
CH	8587	9501	7715	6815	19621
PBM	2602573	2495616	1535354	1295331	51127458

Table 6.7 – Valeurs relevées pour les indices de qualité Silhouette, Calinski-Harabasz et PBM, des partitions obtenues avec k-means dans le cas de la classification selon cinq vues indépendantes. Les valeurs de k utilisées pour la classification de chaque jeu de données Di sont présentées dans le Tableau 6.6.

6.2.2 Détermination des régions prototypes

Suite au partitionnement du jeu de données synthétiques initial avec k-means, nous disposons pour chaque jeu de données Di d'un ensemble de classes formant une partition. Au sein d'une classe Di-Cj, chaque observation est décrite par les coordonnées (x, y, z, t) d'une région contenant l'information Ii relative à Di. Nous cherchons à déterminer des régions prototypes permettant de généraliser les différentes positions que peuvent prendre une information au sein d'un corpus hétérogène d'images de factures. Pour déterminer les régions prototypes, à partir des classes des partitions obtenues dans la section précédente, nous nous plaçons dans l'ensemble R de tous les rectangles du plan représenté par une image de facture. L'ensemble R est une convexité au sens de van de Vel (van De Vel, 1993) car il contient l'ensemble vide et est stable par intersection quelconque et par réunions emboitées. Ainsi, pour chaque jeu de données Di, donc pour pour chaque information textuelle Ii, nous associons une région prototype à chaque classe de la partition obtenue pour Di. Cette région prototype est définie comme étant l'enveloppe convexe, dans R, des rectangles contenant l'information Ii dans les documents qui constituent la classe considérée. Rappelons que l'enveloppe convexe, dans R, d'un ensemble $\{r_1, \cdots, r_n\}$ est le plus petit rectangle r contenant les r_i. Ainsi, nous obtenons 43 régions prototypes telles que, les 10 premières sont relatives à l'information I1, les 10 suivantes à I2, les 10 suivantes à I3, les 3 suivantes à I4 et les 10 dernières à I5. La Figure 6.7 montre les 3 régions prototypes relatives à l'information I4 dans un repère orthonormé. Cette représentation permet également d'observer les 3 classes obtenues pour le partitionnement du jeu de données D4.

Dans cette Section nous avons présenté notre méthode pour déterminer des régions prototypes généralisant les positions d'informations textuelles ciblées au sein d'images de factures. En disposant de ces prototypes, nous sommes en mesure de localiser une information textuelle donnée au sein d'une image de facture, sans parcourir cette image entièrement. Un moteur de reconnaissance permet d'extraire du texte présent dans des images et de les convertir en texte brut. Le texte brut peut ensuite être manipulé par un programme informatique approprié, un éditeur de texte par exemple. Cependant,

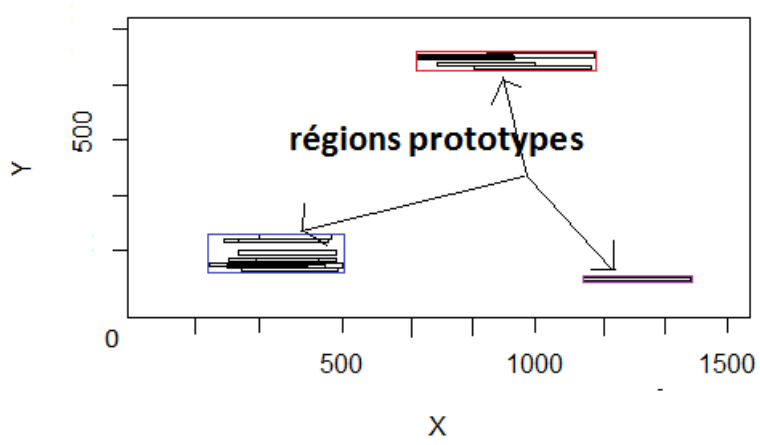


FIGURE 6.7 – Représentation des 3 régions prototypes relatives à l'information I4 dans un repère orthonormé.

selon (Valveny *et al.*, 2013) une majorité de solution de reconnaissance ne sont pas en mesure d'isoler automatiquement des éléments textuels au sein d'images de documents non contraints dans la mise en page. Cela est particulièrement vrai dans le cas des documents administratifs tels que les factures. L'étape suivante de construction de notre système consiste à déterminer des chemins pour parcourir efficacement l'ensemble des régions prototypes définies dans cette Section. Ces chemins sont obtenus à partir du treillis de concepts d'un contexte formel convenablement construit.

6.3 Détermination de chemins pour naviguer au sein des régions prototypes à partir d'un treillis de concepts

Dans la Section précédente nous avons indiqué comment nous déterminons des régions prototypes contenant les informations textuelles à extraire. Dans cette Section nous présentons l'étape 3 de notre système. Cette étape consiste à définir des chemins permettant de parcourir efficacement les régions prototypes construites. Pour cela, l'Analyse Formelle de Concepts (AFC) semble très appropriée. En effet, les chemins que nous cherchons à déterminer peuvent être obtenus à partir du treillis de concepts d'un contexte formel convenablement construit.
La Section 6.3.1 présente la construction du treillis de concepts. La détermination des chemins à partir du treillis est présentée dans la Section 6.3.2.

6.3.1 Construction du treillis de concepts

Pour rappel, un contexte formel est un triplet $\mathbb{K} = (\mathcal{O}, \mathcal{A}, \mathcal{R})$, où $\mathcal{O}$ et $\mathcal{A}$ sont des ensembles et $\mathcal{R}$ une relation binaire de $\mathcal{O}$ vers $\mathcal{A}$ (Wille, 1982; Ganter & Wille, 1999).

Les éléments de $\mathcal{O}$ sont appelés objets et ceux de $\mathcal{A}$ attributs. Un *concept formel* de $\mathbb{K}$ est une paire (X, Y) telle que $Y = X' = \{a \in A : x\mathcal{R}a \text{ pour tout } x \in X\}$ et $X = Y' = \{x \in O : x\mathcal{R}a \text{ pour tout } a \in Y\}$. Notons que la double application de l'opérateur de dérivation $(.)'$ est un opérateur de fermeture, c'est à dire que $(.)''$ est extensif, idempotent et monotone. Les ensembles $X \subseteq O, Y \subseteq A$, tels que $X = X''$ et $Y = Y''$ sont dits fermés. Le sous-ensemble $X \subseteq O$ est nommée extension du concept (X, Y) et Y son intention. Le treillis de concepts du contexte formel $\mathbb{K}$ (Wille, 1982), aussi connu sous le nom de treillis de Galois de la relation binaire $\mathcal{R}$ (Barbut & Monjardet, 1970), est un treillis (complet) $(G(\mathbb{K}), \leq)$, où $G(\mathbb{K})$ est l'ensemble des concepts formels de $\mathbb{K}$ et $\leq$ la relation d'ordre partiel de subconcept/superconcept. Ainsi, un treillis de concepts contient un élément minimum (respectivement un maximum) selon la relation $\leq$, nommé le concept "bottom" (respectivement le concept "top").
Dans cette Section nous considérons un contexte formel où les objets sont des images de factures et les attributs des prédicats $I_i = j$, où $i = 1, \cdots, 5$ désigne les 5 informations textuelles mentionnées dans la Section 6.1, et $j = 1, \cdots, 43$ désigne l'indice des 43 régions prototypes R_1, $\cdots$,R_{43} obtenues en appliquant la méthode présentée dans la Section 6.2.
Une image de facture o_n est en relation avec un prédicat $I_i = j$ si l'information textuelle I_i est localisée dans la région prototype R_j au sein de l'image de facture o_n. Un résumé de ce contexte formel est montré dans le Tableau 6.8. La Figure 6.8 montre le diagramme de Hasse du treillis de concepts dérivé du contexte formel d'images de factures synthétiques. Ce treillis de concepts a été obtenu avec l'outil *conexp*[1].

	I1=1	I1=2	I1=3	I1=4	I1=5	I1=6	I1=7	I1=8	I1=9	I1=10	...	I4=31	I4=32	I4=33	I5=34	I5=35	I5=36	I5=37	I5=38	I5=39	I5=40	I5=41	I5=42	I5=43
o_1	X										...	X			X									
...											...													
o_{895}										X	...	X				X								
...											...													
o_{1270}							X				...		X								X			

TABLE 6.8 – Extrait du contexte formel de l'ensemble des images de factures du jeu de données synthétiques.

6.3.2 Détermination de chemins à partir du treillis de concepts

Pour rappel, étant donné un contexte formel $\mathbb{K} = (O, A, \mathcal{R})$, une règle d'association est une paire (X, Y), définie telle que $X \to Y$, où X et Y sont des ensembles disjoints de A (Agrawal *et al.*, 1993). L'ensemble X est appelé antécédent de la règle $X \to Y$ et Y son conséquent.
Le support d'une règle d'association $X \to Y$ est la proportion d'objets qui contiennent tous les attributs dans $X \cup Y$, i.e. $\frac{|(X \cup Y)'|}{|O|}$.
La confiance d'une règle d'association $X \to Y$ est la proportion d'objets qui contiennent Y, parmi ceux contenant X.
Une règle d'association valide est une règle d'association dont le support et la confiance sont au moins égaux à un seuil minimum fixé de support et un seuil minimum fixé de

1. https ://sourceforge.net/projects/conexp/

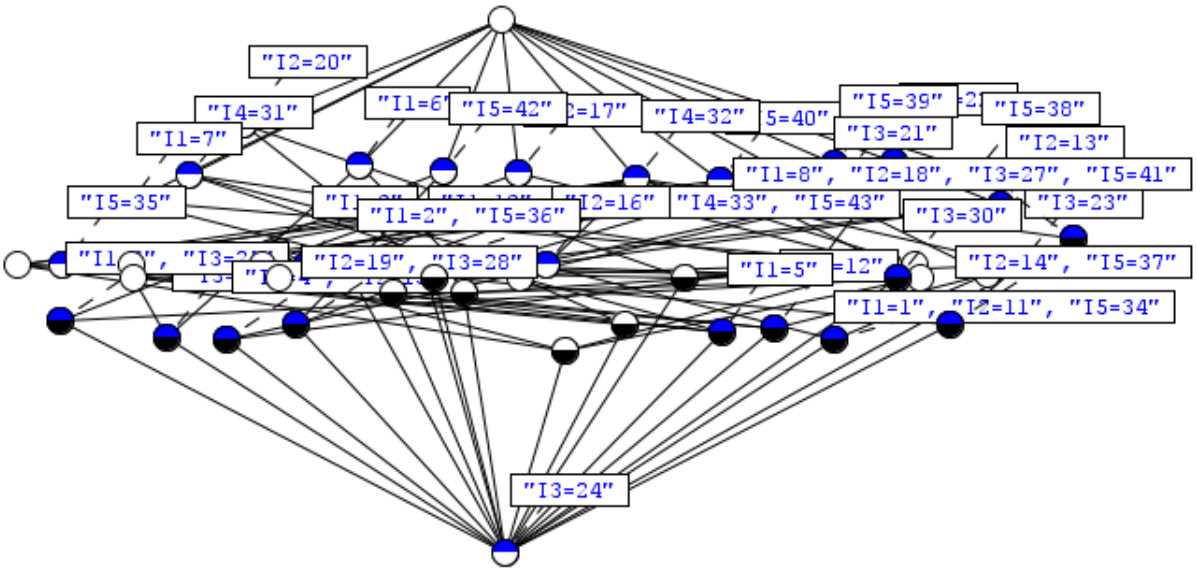


FIGURE 6.8 – Treillis de concepts du contexte formel d'images de factures.

confiance, respectivement. Une règle d'association approximative est une règle d'association dont la confiance est inférieure à 1.
Lorsque le seuil minimum de support est fixé à 0, la base de Luxenburger des règles d'association approximatives est l'ensemble des règles de la forme $X \rightarrow Y \setminus X$, où $X = X''$, $Y = Y''$, $X \subset Y$ et il n'existe pas de Z tel que $Z'' = Z$ et $X \subset Z \subset Y$ (Kuznetsov & Makhalova, 2015).
La base de Luxenburger peut être directement visualisé sur le diagramme de Hasse d'un treillis de concepts. Chaque règle d'association approximative correspond exactement à une arrête du diagramme de Hasse. La Figure 6.8 représente le treillis de concepts du contexte formel d'images de factures synthétiques et montre par conséquent toutes les règles d'association approximatives de la base de Luxenburger pour un seuil minimum de support $minsupp = 0$. L'ensemble des règles d'association approximatives de la base de Luxenburger correspondant au treillis de concepts dérivé du contexte formel d'images de factures synthétiques est présenté dans les Tableaux [2] 6.9 et 6.10. Sur le diagramme de Hasse de la Figure 6.8 une représentation succincte est utilisée pour représenter les informations à propos des intentions et des extensions de concepts formels. Dans cette représentation succincte, si une étiquette d'attribut A est attachée à un concept, cela signifie, que cet attribut apparait dans les intentions de tous les concepts atteignables, en descendant dans le treillis, à partir de ce concept jusqu'au "concept bottom" (le concept le plus bas du treillis). Si une étiquette d'objet O est attachée à un concept, cela signifie, que l'objet O figure dans les extensions de tous les concepts atteignables, en remontant dans le treillis, à partir de ce concept jusqu'au "concept top" (le concept le plus haut du treillis). Sur le diagramme, un nœud bleu et noir signifie qu'il y a un attribut attaché au concept représenté par ce nœud. Un nœud blanc et noir signifie qu'il y a un objet attaché au concept représenté par ce nœud. Par exemple, dans la Figure 6.9, l'arête (tracée en bleu) entre le nœud étiqueté "I5=42" et le nœud étiqueté "I1=9, I3=25" représente la règle d'association approximative I5=42 → I1=9, I3=25 de la base de Luxenburger.

2. Dans les Tableaux le support est exprimé en nombre d'objets contenant tous les attributs $X \cup Y$ étant donnée une règle d'association $X \rightarrow Y$

règle d'association approximative	support	confiance
I2=16 I4=32 → I1=5	207	85 %
I1=6 → I4=31	202	84%
I2=13 I3=22 → I1=2 I4=32 I5=36	215	83 %
I5=39 → I2=16 I4=32	180	80 %
I2=16 I4=32 I5=39 → I1=5 I3=24	144	78 %
I2=17 I4=32 → I3=25	178	72 %
I3=22 I4=31 → I1=4 I2=15 I5=38	128	72 %
I2=17 I5=40 → I1=6 I3=30 I4=31	134	72 %
I2=20 → I1=10 I3=29 I4=33 I5=43	114	71 %
I3=22 → I2=13	307	70 %
I2=16 I4=32 → I5=39	207	70 %
I3=21 → I4=32	207	69 %
I1=1 I3=21 → I2=11 I4=31 I5=34	94	68 %
I3=23 I4=32 → I1=3 I2=14 I5=37	102	68 %
I1=7 → I4=32	109	67 %
{ } → I4=32	1270	65 %
I2=17 → I4=32	274	65 %
I5=38 → I1=4 I2=15 I3=22 I4=31	142	65 %
I5=42 → I1=6 I2=19 I3=28 I4=31	124	60 %
I5=40 → I4=32	230	58 %
I5=40 → I2=17	230	58 %
I4=32 I5=35 → I1=7 I2=17 I3=25	70	57 %
I1=3 I2=17 I4=32 → I3=21 I5=38	88	57 %
I1=6 I4=31 → I2=17 I3=30 I5=40	170	56 %
I1=3 I4=32 → I2=17	157	56 %
I1=7 I4=32 → I2=17 I3=25 I5=35	73	55 %
I2=17 I4=32 → I1=3	178	49 %
I2=17 → I5=40	274	49 %
I4=32 I5=40 → I1=5 I2=16 I3=21	134	47 %
I4=31 → I1=6	362	47 %
I3=21 → I1=1	207	45 %

Table 6.9 – Ensemble des règles d'association approximatives de la base de Luxenburger du treillis de concepts dérivé du contexte formel d'images de factures synthétiques (1ère partie).

règle d'association approximative	support [3]	confiance
I1=7 I4=32 → I2=20 I3=23 I5=40	73	45 %
I3=21 I4=32 → I1=5 I2=16 I5=40	143	44 %
I1=3 I4=32 → I2=14 I3=23 I5=37	157	44 %
I1=6 I4=31 → I2=19 I3=28 I5=42	170	44 %
I1=3 I2=17 I4=32 → I3=25 I5=40	88	43 %
I4=32 I5=35 → I1=1 I2=12 I3=21	70	43 %
I3=22 → I4=31	307	42 %
I5=42 → I1=9 I2=17 I3=25 I4=32	124	40 %
I1=5 I2=16 I4=32 → I3=21 I5=40	175	36 %
I4=31 → I3=22	362	35 %
I1=7 → I2=13 I3=22 I4=31 I5=39	109	33 %
I3=23 I4=32 → I1=7 I2=20 I5=40	102	32 %
I2=20 → I1=7 I3=23 I4=32 I5=40	114	29 %
{ } → I4=31	1270	29 %
I2=16 I4=32 I5=39 → I1=6 I3=26	144	22 %
I4=32 → I1=8 I2=18 I3=27 I5=41	826	16 %
I1=9 I2=17 I3=25 I4=32 I5=42 → I1=1 I1=2 I1=3 I1=4 I1=5 I1=6 I1=7 I1=8 I1=10 I2=11 I2=12 I2=13 I2=14 I2=15 I2=16 I2=18 I2=19 I2=20 I3=21 I3=22 I3=23 I3=24 I3=26 I3=27 I3=28 I3=29 I3=30 I4=31 I4=33 I5=34 I5=35 I5=36 I5=37 I5=38 I5=39 I5=40 I5=41 I5=43	50	0 %

TABLE 6.10 – Ensemble des règles d'association approximatives de la base de Luxenburger du treillis de concepts dérivé du contexte formel d'images de factures synthétiques (2ème partie).

Le diagramme de Hasse d'un treillis de concepts contient des chemins avec lesquels il est possible de se déplacer du concept "top" vers le concept "bottom". Les chemins que nous adoptons pour naviguer à travers l'ensemble des régions prototypes sont exactement ceux correspondant à des séquences de règles d'association de la base de Luxenburger, c'est à dire des séquences consécutives d'arêtes du treillis de concepts en allant du haut vers le bas. En d'autres termes, un *chemin* est une séquence $Y_0 \rightarrow Y_1 \rightarrow \cdots \rightarrow Y_n$, où Y_0 est l'intention du concept formel "top" et pour tout $0 \leq i < n$, $Y_i \rightarrow Y_{i+1}$ est une règle d'association de la base de Luxenburger.
Étant donné un nœud du treillis de concepts, il existe autant de règles d'association approximatives dont l'antécédent est l'intention de ce nœud, qu'il y a de nœuds fils de ce nœud dans le treillis de concepts. Entre deux règles d'association ayant le même antécédent, celui dont le support est le plus élevé est considéré en premier. Par exemple, soit deux chemins p_1 et p_2 :

p_1 : I5=42 → I1=9 → I3=25,
p_2 : I5=42 → I2=19→ I3=28,

dont les valeurs de support sont respectivement de 4% et 6%. Dans le but de localiser et d'extraire les informations I1 à I5 au sein d'une image de facture candidate, et en supposant que I5=42 est un nœud fils direct du nœud "top" et qu'il possède la valeur de support la plus élevée parmi tous les nœuds fils direct du nœud "top", la région prototype R_{42} est visitée en premier afin de trouver l'information I5. Ensuite, en utilisant le chemin p_2, les régions prototypes R_{19} et R_{28} sont visitées afin de trouver les informations I2 et I3

Chemins pour naviguer au sein de l'ensemble des régions prototypes	séquence correspondante de régions prototypes à visiter
p_1	20 10 29 33 44
p_2	6 31 19 28 42
p_3	6 31 17 30 40
p_4	7 32 20 23 40
p_5	7 32 17 25 35
p_6	39 16 32 6 26
p_7	39 16 32 5 24
p_8	39 7 13 22 31
p_9	21 32 3 17 38
p_{10}	21 32 5 16 40
p_{11}	21 32 1 12 35
p_{12}	21 1 11 31 34
p_{13}	21 1 12 32 35
p_{14}	32 16 5 21 40
p_{15}	32 16 5 24 39
p_{16}	32 2 13 22 36
p_{17}	32 17 25 3 40
p_{18}	32 17 25 7 35
p_{19}	32 17 25 9 42
p_{20}	32 17 3 25 40
p_{21}	32 17 3 21 38
p_{22}	32 3 17 21 38
p_{23}	32 3 17 25 40
p_{24}	32 3 14 23 37
p_{25}	32 21 5 16 40
p_{26}	32 21 3 17 38
p_{27}	32 21 1 12 35

Table 6.11 – Ensemble des chemins déterminés à partir de la base de Luxenburger du treillis de concepts dérivé du contexte formel d'images de factures synthétiques (1ère partie).

respectivement. Lorsqu'une information Ii n'est pas trouvée dans une région prototype indiquée par le chemin p_2, alors le chemin p_1 peut être utilisé pour la trouver. Ainsi, toutes les règles d'association approximatives données par la base de Luxenburger sont utilisées pour la localisation et l'extraction des informations I1 à I5. Pour deux chemins de valeurs de support identiques, celui dont la valeur de confiance est la plus élevée est considéré en premier. L'ensemble des chemins obtenus à partir de la base de Luxenburger est disponible dans les Tableaux 6.11 et 6.12. Dans ces Tableaux, les chemins sont ordonnés de p_1 à p_{55} par ordre décroissant de valeur de support. Pour un chemin la séquence correspondante est décrite par les indices des régions prototypes à visiter. Par exemple, le chemin p_1 dont la séquence correspondante est "20 10 29 33 44" indique que les régions prototypes à visiter sont les régions prototypes R_{20}, puis R_{10}, R_{29}, R_{33} et enfin R_{44}.

Dans les systèmes tels que CREDO (Carpineto *et al.*, 2004) et SearchSleuth (Ducrou

Chemins pour naviguer au sein de l'ensemble des régions prototypes	séquence correspondante de régions prototypes à visiter
p_{28}	32 40 5 16 21
p_{29}	32 40 3 17 25
p_{30}	32 40 7 20 23
p_{31}	32 8 18 27 41
p_{32}	32 23 7 20 40
p_{33}	32 23 3 14 37
p_{34}	32 7 17 25 35
p_{35}	32 7 20 23 40
p_{36}	20 10 29 33 43
p_{37}	20 7 23 32 40
p_{38}	31 6 17 30 40
p_{39}	31 6 19 28 42
p_{40}	31 22 4 15 38
p_{41}	31 22 7 13 39
p_{42}	31 22 1 11 34
p_{43}	38 4 15 22 31
p_{44}	38 3 17 21 32
p_{45}	40 17 6 30 31
p_{46}	40 17 3 25 32
p_{47}	40 32 5 16 21
p_{48}	40 32 3 17 25
p_{49}	40 32 7 20 23
p_{50}	22 13 2 32 36
p_{51}	22 13 7 31 39
p_{52}	22 31 4 15 38
p_{53}	22 31 7 13 39
p_{54}	38 6 19 28 31
p_{55}	38 9 17 25 32

TABLE 6.12 – Ensemble des chemins déterminés à partir de la base de Luxenburger du treillis de concepts dérivé du contexte formel d'images de factures synthétiques (2e partie).

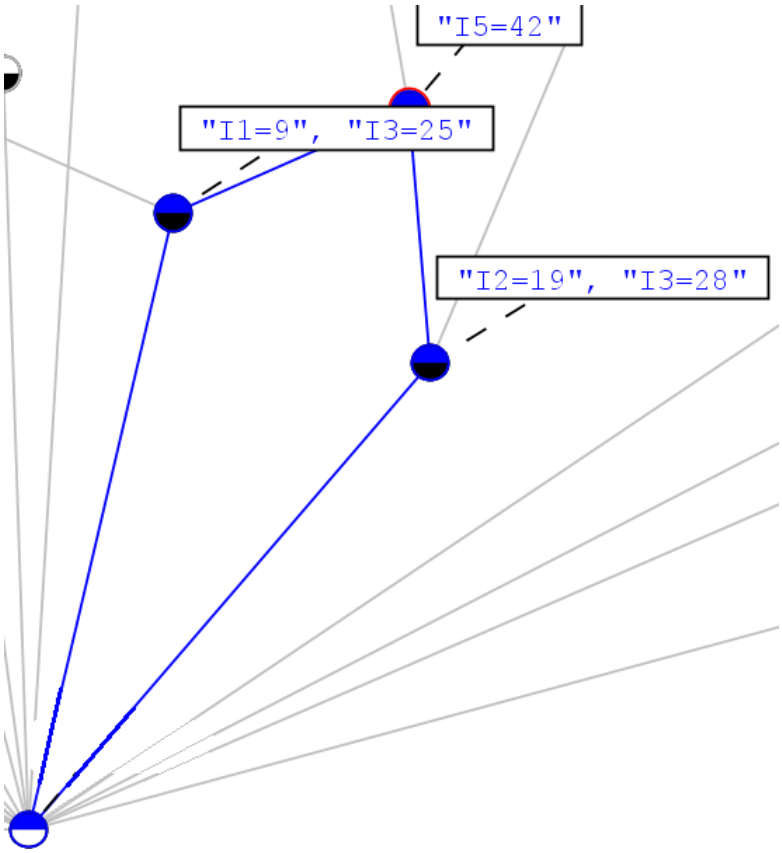


Figure 6.9 – Sous-partie du treillis de concepts de la Figure 6.8. Les arrêtes apparaissant en bleu représentent deux règles d'association approximatives de la base de Luxenburger.

& Eklund, 2007) la stratégie de navigation consiste à se concentrer sur un concept et ses voisins, ce qui peut s'apparenter à un parcours en largeur d'un treillis de concepts. L'efficacité et la performance de l'utilisation de ce type de stratégie pour la recherche Web ont été démontrées dans (Carpineto *et al.*, 2004; Ducrou & Eklund, 2007). Dans notre approche, notre stratégie de navigation consiste à se concentrer sur un concept et ses fils (parcours en profondeur). Les expérimentations que nous avons menées pour comparer notre stratégie de parcours et la stratégie de navigation utilisée dans CREDO et SearchSleuth, montrent que notre stratégie est la plus efficace pour la localisation et l'extraction des informations textuelles I1 à I5 au sein d'images de factures. Les résultats des expérimentations sont présentées dans le Chapitre 7 Section 7.2.2. Nous avons donc choisi notre stratégie de navigation pour notre système.

6.4 Traitement d'une image de facture inconnue fondé sur des modèles incrémentaux

Dans les Sections précédentes nous avons présenté comment à partir d'un jeu de données synthétiques notre système est capable de construire des régions prototypes, puis des chemins pour naviguer efficacement au sein des régions prototypes. Étant donnée une image de facture candidate, notre système est capable d'extraire des informations textuelles en s'appuyant sur les régions prototypes et les chemins. Supposons qu'en

appliquant notre système sur l'image de facture candidate, une ou plusieurs des informations recherchées ne sont pas correctement extraites à partir des régions prototypes et des chemins disponibles. En effet, l'image de facture candidate considérée peut contenir des informations localisées à des positions inconnues dans le jeu initial de données synthétiques. Afin de pallier ce cas de figure, nous avons choisi d'intégrer dans notre système des modèles permettant d'enrichir le système avec de nouvelles données de manière incrémentale. Dans la littérature, des méthodes incrémentales de k-means permettent de mettre à jour dynamiquement des classes existantes avec des données supplémentaires ajoutées de manière incrémentale à un jeu de données initial. Par ailleurs, des algorithmes de construction incrémentale de treillis de concepts sont également décrites dans la littérature. Dans la Section 6.4.1, nous présentons notre méthode de mise à jour incrémentale de l'ensemble de régions prototypes qui s'inspire de l'algorithme k-means incrémental de Chakraborty et al. (Chakraborty & Nagwani, 2011; Chakraborty *et al.*, 2014). Nous présentons également, dans la Section 6.4.2, notre méthode de mise à jour incrémentale des chemins, pour naviguer au sein des régions prototypes, fondée sur l'algorithme incrémental de construction de treillis de concepts de van Der Merwe et al. (Van Der Merwe *et al.*, 2004).

6.4.1 Mise à jour de l'ensemble de régions prototypes

A partir d'un jeu de données existant, notre méthode de mise à jour de l'ensemble de régions prototypes est composée de trois étapes :

1. Création manuelle d'une image synthétique, via notre interface graphique présentée dans la Section 6.1, à partir d'une image de facture inconnue ; de nouvelles données relatives à l'image synthétique sont automatiquement enregistrées en base de données. Ces nouvelles données concernent les caractéristiques des régions rectangulaires renseignées par l'utilisation via l'interface graphique de notre programme de génération d'images synthétiques. Pour chaque région rectangulaire renseignée, le type de l'information contenue (date facture, numéro facture, référence client, etc.), le contenu textuel, les coordonnées (x, y, z, t) de la région rectangulaire considérée sont enregistrées en base de données.
2. Partitionnement du jeu de données synthétiques existant, dans lequel les nouvelles données sont insérées de manière incrémentale.
3. Détermination d'un nouvel ensemble de régions prototypes à partir des classes obtenues à l'étape précédente.

En considérant un image de facture inconnue, dans la première étape, l'utilisateur est invité à créer manuellement une image de facture synthétique correspondante en utilisant l'interface graphique présentée dans la Section 6.1. Nous avons également mentionné dans la Section 6.1 que lorsqu'une image de facture synthétique est créée depuis notre interface, les coordonnées (x, y, z, t) de la région rectangulaire contenant l'information textuelle d'intérêt sont stockées dans une base de données.
La seconde étape consiste à partitionner le jeu de données initial et les nouvelles données en appliquant une version incrémentale de l'algorithme k-means.
La troisième étape consiste à déterminer le nouvel ensemble de régions prototypes, en déterminant, pour chaque classe des partitions obtenues à l'étape précédente, l'enveloppe convexe des rectangles contenant une information Ii dans les documents qui constituent

la classe considérée, comme présentée dans la Section 6.2.2
Dans la Section 6.2.1, nous avons mentionné que l'ensemble des régions prototypes est dérivé d'une partition de classes (que nous appellerons classes existantes par la suite) obtenue à partir de la classification du jeu de données synthétiques initial (aussi nommé jeu de données existant) avec k-means. Dans le scénario spécifique décrit dans cette Section, l'algorithme k-means incrémental que nous présentons est appliqué sur les classes existantes et les données de l'image de facture synthétique précédemment créée. L'algorithme incrémental que nous proposons s'inspire de l'algorithme k-means incrémental proposé par Chakraborty et Nagwani (Chakraborty & Nagwani, 2011) (Algorithme 4). Notre adaptation de cet algorithme est présenté dans l'Algorithme 3.

Entrées:
F_i : $<x_i, y_i, z_i, t_i>$ (où $i = 1, 2, 3, \cdots, m$) les nouvelles données
K : $\{K_1, \cdots, K_k\}$ partition constituée des classes existantes
D : $\{X_1, \cdots, X_n\}$ le jeu de données initial d'images de facture synthétiques
k : nombre de classes contenues dans K
k_f : nombre de classes final
s : seuil minimum pour l'indice Silhouette
p : seuil minimum pour l'indice PBM
v : seuil minimum pour l'indice Calinski-Harabasz
Sorties:
C : une partition

```
début
    initialiser C avec K;
    initialiser k_f avec k;
    pour i = 1 to m faire
        déterminer la classe K_j de centre M telle que dist(F_i,M) est la plus petite;
        K'_j = K_j ∪ F_i et calculer le centre de K'_j;
        ind_1 = silh(K'_j) /*calcul du Silhouette de la classe K'_j*/;
        ind_2 = pbm(K'_j) /*calcul de l'indice PBM de la classe K'_j*/;
        ind_3 =CH(K'_j) /*calcul de l'indice Calinski-Harabasz de la classe K'_j*/;
        si deux indices parmi les trois indices ind_1, ind_2, ind_3 sont ≥ à leur seuil
        (s, p, v) respectif alors
            retirer K_j de C et insérer K'_j dans C
        sinon
            ;
        fin
        créer une nouvelle classe C_i dont le centre est F_i;
        insérer C_i dans C;
        k_f = k_f+1;
    fin
    si k_f ≥ n alors
        k-means(D ∪ F_i, C);
    fin
fin
```

Algorithm 3: Algorithme k-means incrémental.

Lignes 2 et 3 : Une partition C est initialisée avec les classes de K et le nombre de classes final k_f est initialisé avec la valeur de k (le nombre de classes dans K).
Ligne 5 : La classe K_j de centre M dont F_i est le plus proche est déterminée.
Ligne 6 : F_i est insérée dans K_j formant une classe K'_j dont le centre est ensuite calculé.
Lignes 7 à 9 : Pour chaque nouvelle donnée F_i à traiter, nous considérons 3 indices de qualité (Silhouette, PBM, Calinski-Harabasz) pour déterminer si une nouvelle donnée en entrée peut être classée dans une classe existante, ou si une nouvelle classe doit être créée. Pour la nouvelle classe K'_j la valeur de chaque indice est calculée.
Lignes 10 et 11 : Ensuite, si les valeurs d'au moins deux des trois indices utilisés sont au moins égales à leur seuil respectif, alors la classe K_j est remplacée par la classe K'_j dans C. Cela signifie que la nouvelle donnée F_i a été classée avec succès au sein d'une classe existante dans la partition initiale K.
Lignes 13 à 15 : Sinon, une nouvelle classe C_i de centre F_i est créée. C_i est ajoutée à la partition C et le nombre de classes final k_f est incrémenté de 1.
Lignes 18 à 20 : Enfin, si le nombre final de classes k_f est supérieur au nombre de classes initial k, l'ensemble des données formé par la réunion des données initiales et des nouvelles données est classé en appliquant l'algorithme k-means traditionnel initialisé avec les centres de classes de la partition C.

Les seuils s, p, v sont fixés de sorte que la qualité des classes existantes et de la partition initiale ne se dégrade pas dans le meilleur des cas, ou dans le pire des cas, que l'éventuelle dégradation soit contrôlée. Ainsi, nous avons évalué pour chaque indice qu'une dégradation de 5% de la valeur initiale de l'indice était acceptable. Dans la Section 6.2.1 nous avons présenté la classification du jeu de données synthétiques que nous avons adoptée. Dans le Tableau 6.7, qui récapitule les valeurs des indices relevées pour la classification adoptée, nous pouvons observé que les valeurs relevées pour chaque indice laissent à discuter sur la qualité des partitions obtenues. En effet, concernant l'indice Silhouette, on peut lire dans la littérature qu'une partition (ou une classe) est de bonne qualité si la valeur de Silhouette est au moins égale à 0,7. L'indice Silhouette prenant ses valeurs entre 0 et 1, plus la valeur est proche de 1 plus la partition est de bonne qualité. Hors, dans le Tableau 6.7, la partition obtenue par D1 à une valeur de Silhouette égale à 0.64, D2 à une valeur de Silhouette égale à 0.68, le Silhouette de D3 est égale à 0.61, celui de D4 est égale à 0.55 et celui de D5 est égale à 0.73. Vis à vis de la littérature, il semble que la qualité des partitions obtenues pour les jeux de données D1 à D4 ne soit pas suffisamment bonne. Toutefois, dans notre étude nous travaillons à partir de données réelles (des factures de la société GAA) que nous avons simulées. Ces données réelles présentent plus de complexité et de difficulté qu'une majorité de jeux de données standards rencontrés dans la littérature. C'est pourquoi, nous considérons les valeurs de Silhouette (et des autres indices) obtenues dans la Section 6.2.1 acceptables dans le cadre de notre étude. Pour ces mêmes raisons, le taux de dégradation des indices utilisés par notre algorithme k-means incrémental a été établit de manière empirique à 5%, après avoir étudié des taux de dégradation compris entre 1% et 10%.
Enfin, le nouvel ensemble de régions prototypes est dérivé des partitions obtenues en appliquant la méthode décrite dans la Section 6.2.2.
Dans la Section suivante nous présentons notre modèle incrémental pour la mise à jour des chemins déterminés à partir du treillis de concepts du contexte formel d'images de factures synthétiques.

Data:
D : A dataset containing n objects {X1,X2, X3, . . . , Xn} and n : number of data items.
Result: K1 : A Set of clusters.

```
1  begin
2      Let, C_i (where i=1, 2, 3 . . . ) is the new data item;
3      Incremental K-means Pseudo-code :
4      Start
5      a>Let, K represents the already existing clusters.
6      b>Compute the means (M) of existing clusters. And directly clustered the
       new item Ci.
7      for i = 1 to n do
8          find some mean M in some cluster Kp in K such that dis(C_i, M ) is the
           smallest;
9          if dis ( C_i, M )=min then
10             Kp = Kp U Ci;
11             Recomputed the mean M and compare it again.
12         else
13             if dis(C_i != min) then
14                 C_i will be treated as outliers or noisy data.
15                 Update the existing cluster.
16             end
17         end
18     end
19     c>Repeat step b till all the data samples are clustered.
20     End;
21 end
```

Algorithm 4: K-means incrémental de Chakraborty et Nagwani (Chakraborty & Nagwani, 2011).

6.4.2 Mise à jour des chemins pour naviguer au sein de l'ensemble des régions prototypes

La mise à jour des régions prototypes implique également la mise à jour des chemins permettant de naviguer efficacement au sein des régions prototypes. En effet, le nombre de régions prototypes peut avoir augmenté et/ou les coordonnées (x, y, z, t) des régions prototypes peuvent avoir changé, suite à l'application du modèle de mise à jour incrémentale des régions prototypes présenté dans la Section précédente. Pour la mise à jour des chemins deux cas sont à considérer :

- le cas où le nombre de régions prototypes reste inchangé suite à l'insertion d'une nouvelle donnée ;
- le cas où le nombre de régions prototypes a évolué suite à l'insertion d'une nouvelle donnée.

Les deux cas sont présentés ci-dessous.

Cas où le nombre de régions prototypes reste inchangé

Dans le cas où le nombre de régions prototypes reste inchangé (43 régions prototypes) et que seules les coordonnées des régions prototypes ont subi des modifications, la mise à jour du contexte formel et du treillis de concepts qui en découle ne semble a priori pas nécessaire. En effet, la variation des coordonnées des régions prototypes seule n'a pas d'effet sur la structure du contexte formel initial et par conséquent sur la structure du treillis de concepts dérivé. Toutefois, il est à noter que l'insertion de nouveaux objets au sein d'un contexte formel existant impacte directement les valeurs de support, puisque le support d'une règle d'association est fonction du nombre d'objets du contexte formel considéré. Comme notre stratégie de navigation, au sein de l'ensemble des régions prototypes à partir des chemins, s'appuie en partie sur le support des règles d'association approximatives de la base Luxenburger, il peut être intéressant de mettre à jour le contexte formel initial dans ce cas là.
Van der Merwe et al. (Van Der Merwe *et al.*, 2004) proposent un algorithme nommé "AddIntent" permettant de construire le treillis de concepts d'un contexte formel dans lequel les objets peuvent être insérés un à un. AddIntent est un algorithme incrémental qui prend en entré un treillis de concepts G_i construit à partir des i premiers objets d'un contexte formel et qui insère un objet supplémentaire o afin de générer un nouveau treillis de concepts G_{i+1} (Algorithme 5). Une implémentation de l'algorithme AddIntent est disponible sur le Web [4].
Dans notre scénario spécifique, le treillis de concepts G_i est le treillis de concepts existant dérivé du contexte formel initial d'images de factures synthétiques. L'objet supplémentaire o à ajouter peut être l'image synthétique de la facture candidate inconnue. En appliquant l'algorithme incrémental de van der Merwe, nous obtenons un treillis de concepts actualisé G_{i+1}. Finalement, en appliquant notre méthode, décrite dans la Section 6.3.2, pour déterminer des chemins à partir d'un treillis de concepts, nous sommes en mesure de déterminer un nouvel ensemble de chemins pour naviguer au sein d'un nouvel ensemble de régions prototypes.

```
1 Procedure CreateLatticeIcrementally (O, A, R)
  Data: BottomConcept := (∅, A )
  Result: G := {BottomConcept}
2 begin
3 |   for each o in O do
4 |   |   ObjectConcept = AddIntent(o', BottomConcept, G)
5 |   |   Add o to the extent of ObjectConcept and all concepts above
6 |   end
7 end
```

Algorithm 5: Algorithme incrémental AddIntent de Van der Merwe et al. (Van Der Merwe *et al.*, 2004).

4. https ://github.com/tims/addintent

Cas où le nombre de régions prototypes change

Dans le cas où, suite à l'insertion d'une nouvelle donnée, la mise à jour de l'ensemble des régions prototypes produit un nombre différent de régions prototypes, il est clair que les chemins existants ne sont plus adaptés pour naviguer au sein du nouvel ensemble de régions prototypes. En effet, soit L^1 l'ensemble des chemins existants permettant de naviguer parmi l'ensemble R^1 des 43 régions prototypes obtenues dans la Section6.2.2, soit R^2 un nouvel ensemble de régions prototypes obtenu suite à une mise à jour de R^1 et contenant 44 régions prototypes. A partir de l'ensemble des chemins de L^1 il ne sera pas possible d'atteindre la région $R_j \in R^2$ absente de R^1 et qui n'est donc pas connue des chemins de L^1. Par conséquent, il apparait dans ce cas, que le contexte formel initial (a fortiori le treillis de concepts dérivé) peut ne plus convenir vis à vis des nouvelles données ajoutées.Pour rappel, dans la Section 6.3.1 nous considérons un contexte formel où les objets sont les m images de factures synthétiques initiales et les attributs des prédicats $I_i = j$, où $i = 1, \cdots, 5$ désigne les 5 informations textuelles mentionnées dans la Section 6.1, et $j = 1, \cdots, 43$ désigne l'indice des 43 régions prototypes R_1, $\cdots$,R_{43} obtenues dans la Section6.2.2. Une image de facture $\mathrm{o_n}$ est en relation avec un prédicat $I_i = j$ si l'information textuelle I_i est localisée dans la région prototype R_j au sein de l'image de facture $\mathrm{o_n}$.
En considérant l'insertion d'une nouvelle donnée (une nouvelle image de facture synthétique) dans un ensemble $E = \{e_1, \cdots, e_m\}$ constitué des m images de factures synthétiques initiales, E est alors constituée de $m + 1$ images de factures synthétiques. Supposons que l'insertion de la nouvelle donnée déclenche la mise à jour de l'ensemble des 43 régions prototypes R_1, $\cdots$,R_{43} produisant un nouvel ensemble $R^2 = \{R_1^2, \cdots, R_{44}^2\}$ de 44 régions prototypes. Le contexte formel correspondant est donc un contexte formel $\mathbb{K} = (\mathcal{O}, \mathcal{A}, \mathcal{R})$ où les objets sont les $m + 1$ éléments de E et les attributs des prédicats $I_i = j$, où $i = 1, \cdots, 5$ désigne les 5 informations textuelles, et $j = 1, \cdots, 44$ désigne l'indice des 44 régions prototypes de R^2. Nous pouvons noter que le contexte formel $\mathbb{K}$ ainsi construit est distinct du contexte formel initial considéré dans la Section 6.3.1. Par conséquent, le treillis de concepts dérivé du contexte formel $\mathbb{K}$, décrit ci-dessus, est distinct du treillis de concepts dont nous disposons initialement.
Dans le cas précédent nous avons vu comment l'algorithme AddIntent construit un treillis de concepts à partir d'un treillis de concepts G_i existant et d'un nouvel objet à insérer. Dans le cadre d'utilisation de cet algorithme, la structure du contexte formel initial ne peut pas subir de modification suite à l'insertion d'un nouvel objet. Hors, nous venons de voir comment l'insertion d'une nouvelle donnée dans notre système peut impacter la structure du contexte formel existant et par conséquent la structure du treillis de concepts existant. L'algorithme AddIntent ne semble donc pas approprié dans ce cas là. Finalement, dans le cas où l'insertion d'une nouvelle donnée produit un nouvel ensemble de régions prototypes, la mise à jour des chemins permettant de naviguer au sein de cet ensemble se déroule de la manière suivante :

1. Construction d'un contexte formel approprié, à partir du nouvel ensemble de régions prototypes et du nouvel ensemble d'images de factures synthétiques.

2. Construction du treillis de concepts du contexte formel précédemment construit.

3. Détermination des chemins à partir du treillis de concepts obtenu en appliquant la méthode décrite dans la Section 6.3.2.

6.5 Conclusion

Dans ce Chapitre nous avons présenté notre système d'extraction d'informations textuelles au sein d'images de factures, fondé sur des régions prototypes et des chemins pour naviguer au sein de l'ensemble des régions prototypes. Le système est constitué de cinq étapes :

1. Produire un jeu de données synthétiques à partir d'images de factures réelles contenant les informations d'intérêts.
2. Partitionner les données produites, puis déterminer les régions prototypes à partir de la partition obtenue.
3. Déterminer des chemins pour parcourir les régions prototypes, à partir du treillis de concepts d'un contexte formel convenablement construit.
4. Extraire à l'aide d'un moteur de reconnaissance, une liste d'informations textuelles au sein des régions prototypes en étant guidé par les chemins.
5. Mettre à jour le système de manière incrémentale suite à l'insertion de nouvelles données.

La première étape est importante lorsque le système doit extraire des informations au sein d'une image de facture inconnue. En effet, si des informations à extraire, au sein d'une telle facture, ne sont pas retrouvées, alors une représentation synthétique de l'image de facture peut être produite. Ceci déclenche alors une mise à jour incrémentales de l'ensemble des régions prototypes et du treillis de concepts dont est déduit l'ensemble des chemins pour parcourir l'ensemble des régions prototypes.
En comparaison, les méthodes proposées par (Bartoli *et al.*, 2014; Belaïd *et al.*, 2011; Cesarini *et al.*, 2003) consistent à prédire la classe (l'émetteur) d'une facture en s'appuyant sur un modèle de classification supervisé. De telles méthodes, permettent de localiser et d'extraire une information textuelle à extraire de manière précise à partir de la position absolue d'une région contenant cette information, étant donnée la classe de l'image de facture. Dans ces méthodes, l'extraction d'informations textuelles est guidée par un modèle (ou masque). Le modèle définit une région rectangulaire unique pour chaque information à extraire au sein d'une image de facture similaire au modèle. Comme évoqué dans l'état de l'art, la construction automatique de modèles est une approche intéressante mais est limitée par l'hétérogénéité des émetteurs des documents. L'élaboration des modèles de classification supervisée sur lesquels s'appuient ces méthodes, nécessitent l'intervention d'un expert afin d'étiqueter un ensemble d'images sur lequel un modèle peut être entrainé. Les phases de paramétrage et d'entrainement peuvent être couteuses en temps et en ressource et pas toujours faciles à réaliser.
Notre méthode, fondée sur des modèles de classification non supervisée (partitionnement de données, analyse formelle de concepts) présente plusieurs avantages :

— capacité à traiter des images de documents d'émetteurs hétérogènes : notre système est capable de construire des régions prototypes génériques pour un ensemble d'images d'émetteurs variés ;
— indépendance à la mise en page adoptée : notre système en utilisant des régions prototypes est capable d'extraire une information textuelle au sein d'une région rectangulaire ciblée au sein d'une image ;
— capacité à traiter des images de documents sans connaissances a priori : notre système est fondé sur un partitionnement de données synthétiques et la construction

du treillis de concepts d'un contexte formel convenablement construit ; l'intervention d'un expert, en amont du système, pour identifier des classes au sein des images n'est pas nécessaire, contrairement aux approches fondées sur un modèle de classification supervisée.

Le Chapitre suivant présente la mise en œuvre de la tâche d'extraction d'informations textuelles au sein d'images de facture avec notre système. Les résultats d'une évaluation expérimentale de notre système, pour l'extraction d'informations textuelles au sein d'un corpus d'images de factures réelles, sont également présentés.

Publications
– Pitou, Cynthia, & Diatta, Jean. 2016. Construction de régions prototypes pour la localisation et l'extraction d'informations textuelles dans des documents numérisés : cas des factures.*In : AAFD & SFC 2016 : Francophone International Conference on Data Science.* – Pitou, Cynthia, & Diatta, Jean. 2016. Textual Information Extraction in Document Images Guided by a Concept Lattice. *In : Proceedings of the Thirteenth International Conference on Concept Lattices and Their Applications, 325–336.*

Chapitre 7

Extraction d'informations textuelles au sein d'images de factures à l'aide de notre système

Sommaire

7.1 Processus d'extraction d'informations textuelles 147

7.2 Évaluation expérimentale de notre système pour l'extraction d'informations textuelles au sein d'images de factures . 149

7.2.1 Évaluation expérimentale de notre système pour la localisation et l'extraction d'informations textuelles 150

7.2.2 Évaluation expérimentale de notre stratégie de parcours des régions prototypes guidée par les chemins 153

7.2.3 Évaluation des modèles de mise à jour incrémentale des régions prototypes et des chemins pour naviguer dans l'ensemble des régions prototypes 153

7.3 Conclusion 155

Dans ce Chapitre, nous nous intéressons à la tâche de localisation et d'extraction d'informations textuelles au sein d'images de factures. Dans notre système, présenté dans le Chapitre 6, la tâche d'extraction de texte est réalisée à l'aide d'un moteur d'OCR nommé Tesseract OCR [1]. Nous avons choisi Tesseract OCR car c'est un outil open source, gratuit et qui fournit une API JAVA nous permettant de l'intégrer facilement à notre système, dont le code source principal est écrit en JAVA. Des moteurs d'OCR commerciaux tels que ABBYY sont connus pour obtenir des taux de reconnaissance de texte meilleurs que ceux d'outils gratuits comme Tesseract OCR. Néanmoins, aucun de ces deux moteurs d'OCR n'est capable de localiser de manière ciblée une information donnée telle que le montant d'une facture par exemple. En effet, leur tâche consiste uniquement à transformer, aussi efficacement que possible, tout le texte contenu au sein d'images en texte brut. Le système que nous proposons permet, quant à lui, d'extraire un ensemble d'informations textuelles données au sein d'images de factures de manière ciblée sans parcourir les images toutes entières. Le processus d'extraction d'informations textuelles à l'aide de notre système est présenté dans la Section 7.1. La Section 7.2

1. https ://github.com/tesseract-ocr

présente quelques résultats expérimentaux permettant d'évaluer notre système pour l'extraction d'informations textuelles au sein d'images de factures.

7.1 Processus d'extraction d'informations textuelles

Afin d'extraire des informations d'intérêts, notre système, présenté dans le Chapitre 6, réalise une reconnaissance optique de caractères sur des régions prototypes en utilisant les chemins déterminés dans la Section 6.3 du Chapitre 6. La reconnaissance optique de caractères est réalisée avec Tesseract OCR. Pour rappel, un chemin est une séquence $Y_0 \rightarrow Y_1 \rightarrow ... \rightarrow Y_n$, où Y_0 l'intention du concept formel "top" et pour tout $0 \leq i < n$, $Y_i \rightarrow Y_{i+1}$ est une règle d'association approximative de la base de Luxenburger. Il est à noter que chaque nœud Y_α d'une telle séquence représente un ensemble de prédicats I$i = j$ indiquant que l'information Ii a été observée dans la région prototype R_j. Étant donnée une image de facture F, un ensemble $\mathcal{I} = \{I1, ..., I\text{k}\}$ d'informations à extraire et un ensemble de chemins $\mathcal{P} = \{p_1, \cdots, p_n\}$ permettant de naviguer au sein d'un ensemble $R = \{R_1, \cdots, R_r\}$ de régions prototypes, le processus d'extraction des informations textuelles Ii au sein de F consiste à :

1. Ordonner l'ensemble des chemins par ordre décroissant des valeurs de support des intentions
2. Tant qu'il y a des informations à extraire :
   1. Fixer le premier chemin, disons p_1, de la liste ordonnée de chemins
   2. Pour chaque nœud donné par p_1 : réaliser un OCR sur chaque région prototype indiquée par le nœud considéré, dans le but d'extraire l'information Ii correspondante
   3. Retirer les informations Ii extraites de l'ensemble $\mathcal{I}$
   4. Si l'ensemble $\mathcal{I}$ contient encore des informations, fixer le chemin suivant donné par la liste $\mathcal{P}$, disons p_2 et repartir à l'étape 2.1 en considérant p_2.
3. S'arrêter lorsque l'ensemble $\mathcal{I}$ ne contient plus d'informations ou que tous les chemins ont été utilisés.

Le processus décrit ci-dessus est présenté dans l'Algorithme 6. La Figure 7.1 montre un exemple de parcours d'un chemin (une séquence de nœuds) visible dans un treillis de concepts. Cette Figure est une partie du diagramme de Hasse de la Figure 6.8. Le chemin mis en valeur sur le diagramme est la séquence de nœuds "I4=32 → I2=16 → I1=5". Cette séquence de nœuds correspond à la règle d'association approximative "I4 = 32 I5=40 → I1=5 I2=16 I3=21" de la base de Luxenburger. La liste des chemins dérivés de l'ensemble des règles d'association de la base de Luxenburger est disponible dans les Tableaux 6.11 et 6.12. Sur le diagramme de Hasse de la Figure 7.1 une représentation succincte est utilisée pour représenter les informations à propos des intentions et des extensions de concepts formels. Dans cette représentation succincte, si une étiquette d'attribut A est attachée à un concept, cela signifie, que cet attribut apparait dans les intentions de tous les concepts atteignables, en descendant dans le treillis, à partir de ce concept jusqu'au "concept bottom" (le concept le plus bas du treillis). Si une étiquette d'objet est attachée à un concept, cela signifie, que cet objet figure dans les extensions de tous les concepts atteignables, en remontant dans le treillis, à partir de ce concept jusqu'au "concept top" (le concept le plus haut du treillis). Sur le diagramme, un nœud

bleu et noir signifie qu'il y a un attribut attaché au concept représenté par ce nœud. Un nœud blanc et noir signifie qu'il y a un objet attaché au concept représenté par ce nœud.

Entrées:
$\mathcal{F}$: une image de facture
$\mathcal{I} = \{I1, \cdots, I\text{k}\}$: un ensemble d'informations à extraire
$\mathcal{P} = \{p_1, \cdots, p_n\}$: un ensemble de chemins
$R = R_1, \cdots, R_r$: un ensemble de régions prototypes
Sorties:
$\mathcal{S}$: un ensemble de chaines de caractères

```
1  début
2      i=1
3      j=1
4      tant que i < k or j < n faire
5          Fixer p_j : pour chaque nœud donné par p_j faire
6              pour chaque région prototype R_m indiquée par ce nœud faire
7                  réaliser un OCR sur la région prototype R_m
8                  if le texte brut extrait est non vide then
9                      insérer le texte brut extrait dans la liste S de résultats
10                     retirer l'information Ii correspondant à la région R_m de la liste
                       I
11                     i++
12                 end
13             fin
14         fin
15         j++
16     fin
17 fin
```

Algorithm 6: Procédure d'extraction d'informations textuelles au sein d'une image de facture.

A l'issue du processus d'extraction, il est possible que certaines informations n'ont pas pu être extraites. En effet, l'image de facture traitée par le système peut contenir des informations localisées à des positions inconnues dans le jeu initial de données synthétiques. Dans ce cas là, notre système invite l'utilisateur à renseigner les positions (x, y, z, t) des régions rectangulaires contenant les informations textuelles manquantes, en utilisant l'interface graphique permettant de créer des images de factures synthétiques (Chapitre 6 Section 6.1). Les coordonnées des régions ainsi renseignées par l'utilisateur, sont ensuite insérées dans la base de données d'images de factures. L'insertion de ces nouvelles informations déclenche, si il y a lieu, la mise à jour incrémentale des régions prototypes et des chemins pour parcourir les régions prototypes obtenues avec la méthode décrite dans le Chapitre 6 Section 6.4. De cette manière, si l'image de facture précédemment traitée se représente, les informations auparavant non extraites peuvent cette fois-ci être extraites par notre système. La Section suivante présente quelques résultats expérimentaux de l'application de notre système sur un corpus d'images de factures réelles.

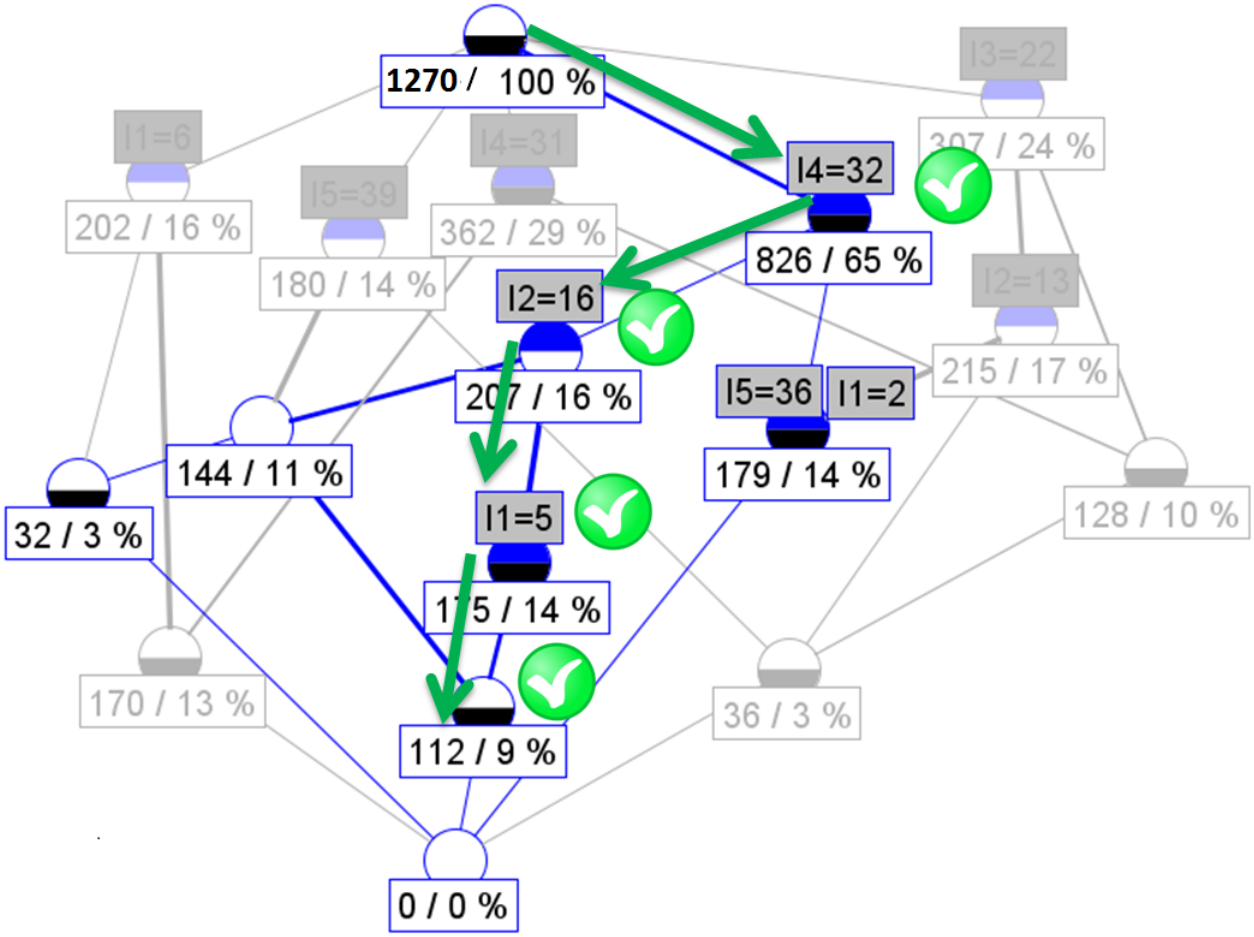


FIGURE 7.1 – Exemple de parcours d'un chemin (une séquence de nœuds) du treillis de concepts du contexte formel d'images de factures (Tableau 6.8).

7.2 Évaluation expérimentale de notre système pour l'extraction d'informations textuelles au sein d'images de factures

Afin d'évaluer notre système pour l'extraction d'informations textuelles au sein d'images de factures, nous avons réalisé des expérimentations sur des images de factures réelles. Pour rappel, nous disposons d'un corpus d'images de factures réelles fournies par la société GAA ainsi que d'un corpus d'images de factures synthétiques dont la construction est évoquée dans le Chapitre 6 Section 6.1. Dans cette Section, nous présentons trois types d'évaluation de notre système :

— évaluation globale du système pour la localisation et l'extraction d'informations textuelles au sein d'images de factures,
— évaluation de notre stratégie de parcours des régions prototypes guidée par des chemins,
— évaluation de notre approche de mise à jour des régions prototypes et des chemins

pour naviguer au sein de l'ensemble des régions prototypes.

7.2.1 Évaluation expérimentale de notre système pour la localisation et l'extraction d'informations textuelles

Les expérimentations consistent à extraire les informations I1 à I5, dans 4 ensembles de respectivement 200, 400, 800 et 1000 échantillons indépendants. Les échantillons sont obtenus en suivant une méthode d'échantillonnage arithmétique (Langley, 1971; Sug, 2009) au sein du corpus de 1270 factures réelles dont la distribution est rappelée dans le Tableau 7.1. Pour rappel, les informations que nous cherchons à extraire sont :

— I1 : le numéro de facture
— I2 : la date de facture
— I3 : une référence client
— I4 : le numéro d'immatriculation du véhicule assisté
— I5 : l'identifiant siret du prestataire de service.

prestataire	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
nb images	30	32	34	36	38	40	50	50	63	69	74	81	92

prestataire	P14	P15	P16	P17	P18	TOTAL
nb images	95	65	112	130	179	1270

TABLE 7.1 – Distribution originale du corpus d'images de factures réelles.

Toutes les images de factures sont des images d'une page, en couleur ou en niveau de gris, au format A4. En dépit du fait que notre système est entrainé sur des images de factures synthétiques, dans cette Section, nous présentons les résultats de tests de notre système sur des images de factures réelles. En effet, les images de factures réelles peuvent contenir du bruit qui n'est pas présent au sein des images de factures synthétiques, comme par exemple, une annotation manuscrite, une zone de texte surlignée, ou un logo. De plus, les images de factures réelles peuvent être en couleur, ou avoir été numérisées avec une faible qualité de numérisation, ou encore présenter des distorsions. Ce bruit constitue une difficulté supplémentaire dans la réalisation de la tâche d'extraction d'informations. Par conséquent, le corpus d'images de factures réelles nous a semblé plus intéressant pour tester notre système de localisation et d'extraction d'informations textuelles.
Nous avons réalisé trois types d'extraction :

1. Extraction au sein des images entières : un OCR est réalisé sur les images entières sans prise en compte de régions spécifiques ;
2. Extraction au sein de sous-régions aléatoires : un OCR est réalisé uniquement sur des sous-régions de dimensions fixes et positionnées aléatoirement au sein des images ;
3. Extraction au sein de sous-régions prototypes en utilisant les chemins présentés dans la Section 6.3 : un OCR est réalisé uniquement sur les sous-régions correspondant à des régions prototypes en utilisant les chemins.

L'OCR est réalisé à l'aide du moteur de reconnaissance Tesseract OCR. Nous considérons deux mesures pour évaluer notre système :

1. la proportion d'informations détectées parmi le nombre total d'informations à extraire (rappel),

2. la proportion d'informations correctes parmi le nombre total d'informations détectées (précision).

Étant donnée une image de facture, d'une part, une information à extraire est considérée détectée, si une quelconque chaine de caractères est trouvée au sein de l'image. D'autre part, une information à extraire est considérée correctement extraite, si la chaine de caractères extraite correspond exactement à l'information visuelle qui peut être lue au sein de l'image. Par exemple, si le montant total est une information à extraire et en supposant que ce montant total est de 107€ dans l'image de facture. Pendant le traitement, si la chaine de caractères retrouvée est "101€", le montant total n'est pas considéré correctement extrait car le montant total réellement mentionné dans l'image de facture originale est "107€".

	Nb info. détectées	Nb info. correctes	Rappel	Précision
OCR seul	5641	3494	55,02%	61,94%
OCR + régions aléatoires	5105	1988	31,30%	38,94%
Notre système	4770	4520	71,18%	94,76%

Table 7.2 – Résultats obtenus pour l'extraction des informations textuelles I1 à I5 au sein de l'ensemble de 1000 échantillons d'images de factures réelles.

Les résultats des expérimentations, sur l'ensemble de 1000 échantillons d'images de factures réelles, sont présentés dans le Tableau 7.2. Le graphique de la Figure 7.2 présente l'évolution du rappel et de la précision pour les expérimentations menées sur les 4 ensembles d'échantillons d'images de factures, pour chaque type d'extraction. Notons, que d'après l'étude de (Patel *et al.*, 2012), le moteur de reconnaissance Tesseract OCR a un taux de reconnaissance de 70% pour la reconnaissance d'images de plaque d'immatriculation en niveaux de gris. D'après les valeurs de rappel et de précision relevées pour la tâche d'extraction avec l'OCR seul sur les images de factures entières, sans prise en compte de régions spécifiques, nous observons que l'efficacité du moteur de reconnaissance Tesseract OCR est relativement faible (55%). A l'inverse, nous observons que la performance relevée pour notre système est d'environ 75%. L'extraction d'informations à partir de sous-régions aléatoires obtient un rappel de 30% et une précision de 37%. Nous pouvons donc dire que la tâche d'extraction d'informations textuelles est largement améliorée en utilisant notre système. Il est à noter que la valeur-p [2] obtenue, pour l'extraction des informations textuelles avec notre système, est de 8.799e-05, ce qui signifie que les résultats obtenus sont significatifs.

2. Test statistique permettant de mesurer la compatibilité des données avec l'hypothèse privilégiée - *https ://onlinecourses.science.psu.edu/statprogram/node/138*

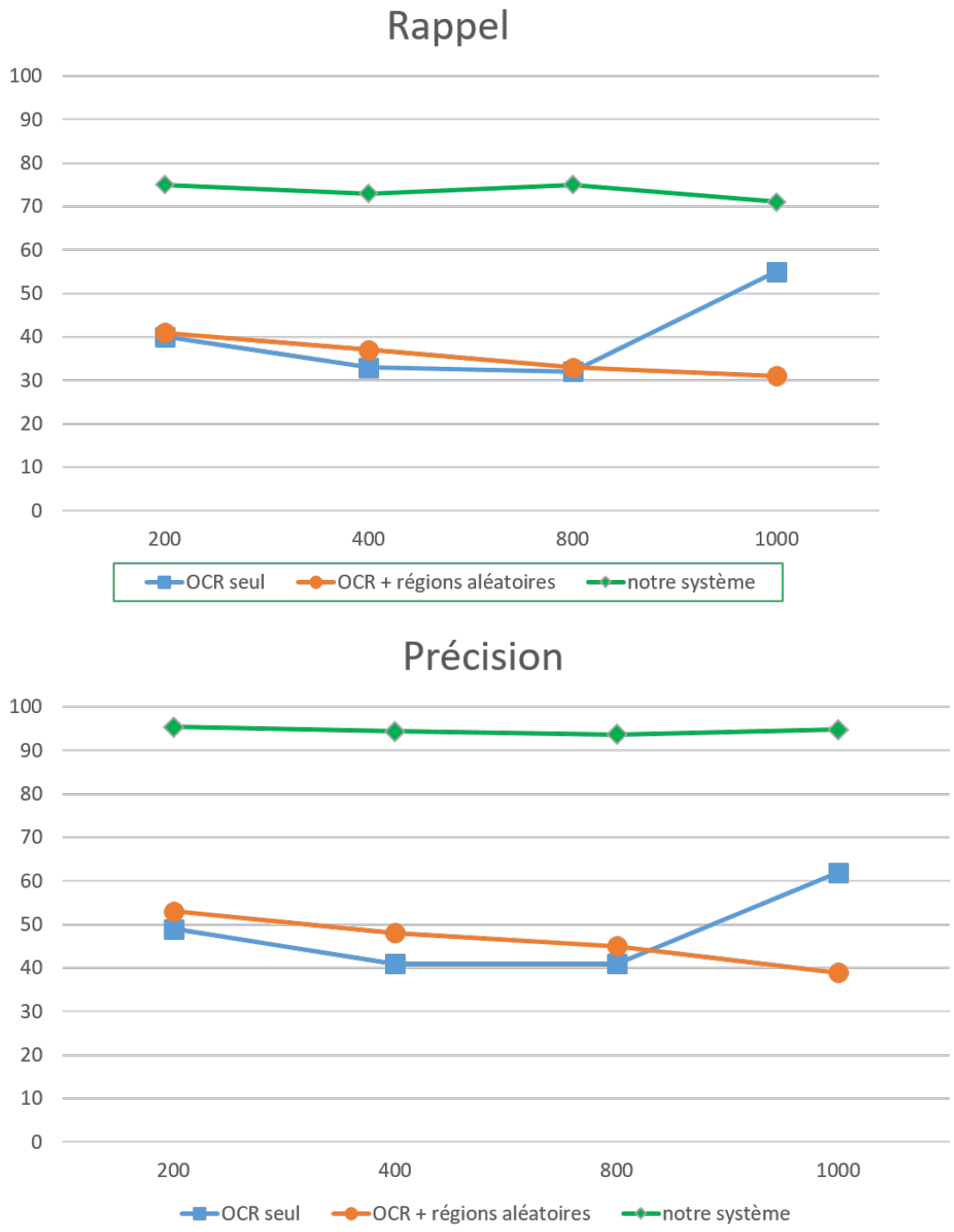


FIGURE 7.2 – Rappel et précision obtenus pour chaque type d'extraction réalisé sur les 4 ensembles de 200, 400, 800 et 1000 échantillons d'images de factures réelles respectivement.

7.2.2 Évaluation expérimentale de notre stratégie de parcours des régions prototypes guidée par les chemins

Afin d'évaluer l'efficacité de l'utilisation des chemins dont nous disposons, pour l'extraction des informations textuelles I1 à I5, d'autres expérimentations ont été menées, en plus de celles présentées dans la Section précédente. A partir de l'ensemble des 43 régions prototypes et d'un échantillon de 100 images de factures réelles, tirées aléatoirement au sein du corpus initial de 1270 images de factures réelles, nous avons réalisé 5 types d'extraction :

- E1 extraire les informations I1 à I5 en parcourant les régions prototypes de manière aléatoire ;
- E2 extraire les informations I1 à I5 en parcourant les régions prototypes par ordre d'indice en commençant par la région prototype R_1, puis R_2 et ainsi de suite jusqu'à la région prototype R_{43} ;
- E3 extraire les informations I1 à I5 en parcourant les régions prototypes dans l'ordre inverse des indices, en commençant par la région prototype R_{43}, puis R_{42} et ainsi de suite jusqu'à la région prototype R_1 ;
- E4 extraire les informations I1 à I5 en parcourant les régions indiquées par les nœuds du treillis de concepts à la manière de CREDO et SearchSleuth (parcours en largeur) ;
- E5 extraire les informations I1 à I5 en parcourant les régions prototypes à partir des séquences de régions prototypes à visiter indiquées par les chemins.

Les expérimentations consistent à extraire les informations textuelles I1 à I5 au sein de 100 images de factures réelles. Pour l'ensemble des 100 images à traiter, une tâche d'extraction peut visiter 4300 régions au maximum. Pour chaque type d'extraction, nous avons relevé le nombre total de régions visitées ainsi que le rappel et la précision obtenue par l'OCR pour la reconnaissance. Les résultats obtenus sont présentés dans le Tableau 7.3. Les valeurs de rappel et de précision obtenues sont approximativement les mêmes quelque soit le type d'extraction réalisé. Cela s'explique par le fait que pour chaque type d'extraction, le même ensemble de régions prototypes est considéré et le même moteur de reconnaissance (Tesseract OCR) est utilisé. Finalement, nos expérimentations montrent que le type d'extraction qui obtient le plus petit nombre de régions visitées est l'extraction guidée par les chemins. Dans le Tableau 7.3 le temps moyen (en seconde) de traitement d'une image est également reporté pour chaque expérimentation. Les temps moyens de traitement relevés montrent que pour l'expérimentation E5, appliquant notre méthode, une image de facture est traitée plus rapidement. La p-value mesurée est de 10.712e-05.

7.2.3 Évaluation des modèles de mise à jour incrémentale des régions prototypes et des chemins pour naviguer dans l'ensemble des régions prototypes

Afin d'évaluer les modèles de mise à jour incrémentale présentée dans le Chapitre 6 Section 6.4, nous avons réalisé une expérimentation consistant à observer l'évolution de l'ensemble des régions prototypes et des chemins. Nous avons appliqué notre système à un ensemble de 10 images de factures émises par 10 prestataires de services incon-

	Nb total de régions prototypes visitées	Rappel	Précision	Temps moyen de traitement par image
E1	2045	80%	54,0%	10,33s
E2	2008	80%	56,4%	9,22s
E3	1920	80%	54,5%	9,03s
E4	1723	80%	55,5%	8,94s
E5	1657	80%	55,5%	7,24s

TABLE 7.3 – Résultats obtenus pour l'extraction des informations textuelles I1 à I5 au sein d'un ensemble de 100 échantillons d'images de factures réelles.

nus de notre système. D'une part, nous avons observé que le processus de mise à jour incrémentale des régions prototypes n'est pas systématiquement déclenché. En effet, le processus de mise à jour a été déclenché 4 fois sur 10. Cela signifie que l'ensemble des régions prototypes initial et l'ensemble des chemins initial ont été efficaces 6 fois sur 10 pour l'extraction des informations recherchées. Dans les 4 cas où le processus de mise à jour incrémental a été déclenché, nous avons observé que dans tous les cas, le nombre de régions prototypes initial est resté stable. Seules les propriétés (coordonnées (x, y, z, t)) des régions prototypes ont subit des variations. En effet, les régions prototypes qui ont subies des modifications ont été élargies (la surface de la région prototype finale est plus grande que la surface de la région prototype initiale). Cependant, nous avons constaté que l'agrandissement de ces régions prototypes n'a pas eu d'effet sur la tâche d'extraction d'informations. Une information non correctement extraite, avant le déclenchement de la mise à jour, n'était toujours pas correctement extraite suite à la mise à jour. Nous avons identifié deux causes principales à cela :

— la performance de l'OCR utilisé peut être mis en cause : nous pensons que la performance de l'OCR utilisé impacte directement la performance globale de notre système en ce qui concerne la proportion d'informations textuelles correctement extraites ;

— l'élargissement d'une région prototype peut, dans une certaine mesure, introduire du bruit au sein de celle-ci : en supposant qu'une région prototype contient initialement uniquement une information à extraire, après élargissement de celle-ci, elle peut potentiellement contenir non seulement l'information à extraire mais aussi d'autres informations textuelles apparaissant aux alentours de l'information recherchée au sein d'une image de facture ;

Pour rappel, suite à l'échec du système pour l'extraction d'un certain nombre d'informations recherchées au sein d'une image de facture candidate, l'utilisateur est invité à renseigner manuellement les données concernant les informations non extraites. L'insertion des nouvelles données est réalisée depuis l'interface graphique de notre programme de génération automatique d'images synthétiques présenté dans le Chapitre 6 Section 6.1. Se déclenche alors, le processus de mise à jour incrémentale de l'ensemble des régions prototypes et des chemins. De ce fait, tester les modèles incrémentaux sur un nombre conséquent d'images de factures inconnues peut devenir une tâche couteuse en terme de temps. C'est pourquoi, dans la phase de test des modèles incrémentaux nous nous sommes limités au traitement de 10 images de factures inconnues.

7.3 Conclusion

Dans ce Chapitre nous avons décrit comment effectuer la tâche d'extraction d'informations textuelles au sein d'images de factures à l'aide de notre système. Pour rappel, le système est constitué de cinq étapes :

1. Produire un jeu de données synthétiques à partir d'images de factures réelles contenant les informations d'intérêts.
2. Partitionner les données produites, puis déterminer les régions prototypes à partir de la partition obtenue.
3. Déterminer des chemins pour parcourir les régions prototypes, à partir du treillis de concepts d'un contexte formel convenablement construit.
4. Extraire à l'aide d'un moteur de reconnaissance, une liste d'informations textuelles au sein des régions prototypes en étant guidé par les chemins.
5. Mettre à jour le système de manière incrémentale suite à l'insertion de nouvelles données.

Les résultats expérimentaux présentés dans la Section 7.2 montrent que notre système améliore significativement l'exactitude de l'extraction d'une information textuelle. Bien que, nous observons que les performances de reconnaissance de notre système sont fortement impactées par les performances du moteur de reconnaissance utilisé (Tesseract OCR), les résultats obtenus par notre système et présentés dans la Section 7.2 semblent prometteurs. Le système que nous proposons semble donc utile pour l'apprentissage (en utilisant une méthode d'analyse de classes) d'un ensemble de régions prototypes contenant des informations ciblées, à partir d'un corpus d'images de factures synthétiques. A partir de ces régions prototypes, le système peut ensuite extraire les informations ciblées automatiquement au sein d'une image de facture candidate, sans parcourir l'image toute entière, ceci indépendamment de la mise en page utilisée par l'émetteur. L'interface graphique que nous proposons permet de reproduire un grand nombre d'images synthétiques à partir d'un nombre limité d'images de factures réelles. Pour notre étude, nous avons généré 1270 images de factures synthétiques à partir de 18 images de factures réelles provenant de 18 émetteurs différents. De plus, les chemins dérivés du treillis de concepts d'un contexte formel d'images de factures semblent être une bonne stratégie pour naviguer au sein de l'ensemble des régions prototypes. En effet, les régions prototypes à visiter sont sélectionnées successivement en étant guidé par les chemins, sans visiter la totalité des régions prototypes dans le meilleur des cas.
Le système que nous proposons a été entrainé et expérimenté pour l'extraction des informations I1 à I5 évoquées dans le Chapitre 6 Section 6.1. Toutefois, il peut être facilement généralisé pour l'extraction d'un plus grand nombre d'informations sans modification. En effet, un tel système généralisé consiste à :

1. produire des images de factures synthétiques à partir d'images de factures réelles : indiquer les positions des rectangles pour chaque information à extraire ;
2. construire le jeu de données synthétiques correspondant ;
3. partitionner le jeu de données synthétiques selon autant de vues qu'il y a d'informations à extraire ;
4. déterminer l'ensemble des régions prototypes à partir des partitions obtenues à l'étape précédente ;

5. construire le treillis de concepts d'un contexte formel convenablement construit ;
6. déterminer des chemins pour naviguer au sein de l'ensemble des régions prototypes à partir du treillis de concepts.

Par ailleurs, notre système est capable de traiter d'autres types d'images de documents, autres que des factures, par exemple des devis, des bons de commandes et des courriers types, contenant des informations ciblées à extraire et dont la nature est connue à l'avance, mais dont les localisations varient selon l'émetteur. Dans un devis par exemple, il peut être intéressant d'extraire de manière automatique l'objet du devis et le tarif indiqué. En particulier, notre système peut être entrainé de la manière suivante pour le traitement de devis :

1. produire des images de devis synthétiques à partir d'images de devis réels ;
2. construire le jeu de données synthétiques correspondant ;
3. déterminer un ensemble de régions prototypes, à partir des partitions obtenues d'un partitionnement du jeu de données ;
4. déterminer des chemins permettant de naviguer dans l'ensemble de régions prototypes, à partir du treillis de concepts d'un contexte formel convenablement construit.

Nous avons vu dans notre état de l'art que des méthodes, telles que celles proposées par (Bartoli *et al.*, 2014; Belaïd *et al.*, 2011; Cesarini *et al.*, 2003), consistent à prédire la classe (l'émetteur) d'une facture en s'appuyant sur un modèle de classification supervisé. De telles méthodes, permettent de localiser et d'extraire une information textuelle à extraire de manière précise à partir de la position absolue d'une région contenant cette information, étant donnée la classe de l'image de facture. Dans ces méthodes, l'extraction d'informations textuelles est guidée par un modèle (ou masque). Le modèle définit une région rectangulaire unique pour chaque information à extraire au sein d'une image de facture similaire au modèle.

Dans le cadre de ce mémoire nous nous sommes restreint à l'extraction d'un ensemble de 5 informations textuelles. Toutefois, un utilisateur peut être intéressé par l'extraction automatique d'autres informations pertinentes, telles que le montant TTC, le montant HT, le montant TVA et le type de service réalisé (dépannage, remorquage, location de véhicule, etc.). En utilisant notre interface de saisie manuelle de régions rectangulaires contenant une information d'intérêt, un utilisateur peut renseigner les positions géographiques d'une liste exhaustive d'informations désirées, à partir d'images réelles servant de modèles. Un ensemble d'images synthétiques seront générées et les positions (coordonnées (x, y, z, t)) de chaque information seront stockées au sein d'une base de données. A partir des données stockées, notre système est, par la suite, en mesure de déterminer des régions prototypes relatives aux informations d'intérêts, puis à déterminer des chemins permettant de naviguer au sein de l'ensemble des régions prototypes de manière efficace.
Notre système semble être adapté au traitement de différents types d'images de documents provenant d'émetteurs hétérogènes. De plus les résultats obtenus dans le cadre d'expérimentations sur des images de factures sont prometteurs.

Publications
– Pitou, Cynthia, & Diatta, Jean. 2016. Construction de régions prototypes pour la localisation et l'extraction d'informations textuelles dans des documents numérisés : cas des factures. *In : AAFD & SFC 2016 : Francophone International Conference on Data Science.* – Pitou, Cynthia, & Diatta, Jean. 2016. Textual Information Extraction in Document Images Guided by a Concept Lattice. *In : Proceedings of the Thirteenth International Conference on Concept Lattices and Their Applications, 325–336.*

Bilan et perspectives

Les travaux menés dans cette thèse portent essentiellement sur l'extraction d'informations textuelles au sein d'images de documents. Nous avons étudié le cas particulier des factures. L'extraction d'informations textuelles est une tâche importante en traitement automatique de documents. Les difficultés liées à cette tâche sont toujours des défis à relever à ce jour. Il existe sur le marché des solutions de traitement automatique de documents. Ces solutions sont généralement couteuses et dédiées au traitement d'un seul type de documents. De plus, au sein d'un même type de documents, la variété de mise en page et de présentation selon l'émetteur, présente une difficulté supplémentaire. D'après la littérature, les approches pour l'élaboration de telles solutions sont, pour une majorité, fondées sur des modèles de classification supervisée pour la prédiction de la classe (le plus souvent l'émetteur) d'un document candidat. Pour chaque classe (émetteur) de documents, des modèles (ou masques) sont construits. Ces modèles embarquent les positions exactes des informations textuelles d'intérêts à extraire, ainsi que d'autres caractéristiques jugées pertinentes. Pour une image de document candidate, l'approche consiste à : (i) déterminer la classe du document ; (ii) à partir du modèle associé, récupérer la position exacte d'une information à extraire ; (iii) extraire cette information à l'aide d'un outil de reconnaissance. Ces approches, pour être performantes, nécessitent de disposer de jeux de données conséquents, d'une part, pour entrainer un modèle de classification et, d'autre part, pour évaluer l'approche adoptée. Dans le contexte de cette thèse, nous ne disposons pas d'un tel jeu de données et la construction de celui-ci présente des contraintes de temps et de ressource importantes. Pour rappel, la problématique de la thèse concerne le besoin qu'à GAA d'automatiser la tâche de recueil d'informations d'intérêts présentes au sein des factures reçues. Nous avons répondu à cette problématique en proposant deux approches pour l'extraction d'informations textuelles au sein d'images de documents.

Les apports de la thèse

Les apports de cette thèse se concentrent autour des points suivants :

Élaboration d'une approche de décomposition d'images fondée sur la décomposition quadtree

En partant du constat que le moteur d'OCR utilisé dans notre étude présente des faiblesses pour la reconnaissance d'une image entière, l'intérêt de cette approche est

d'exécuter la tâche de reconnaissance sur des sous-régions d'une image entière. En effet, de cette manière nous obtenons des régions partielles contenant moins de bruit que l'image entière. Notre approche améliore, ainsi, les performances de reconnaissance du moteur utilisé.

Conception et développement d'un programme informatique pour la génération automatique de données synthétiques

Dans le cadre particulier du traitement automatique de documents, disposer de données synthétiques en nombre suffisant devient une alternative intéressante, d'une part pour l'élaboration de systèmes performants et, d'autre part, pour l'évaluation de ces systèmes. A partir d'un modèle d'image de document, notre programme permet à un utilisateur de renseigner les régions rectangulaires contenant des informations d'intérêts, puis de générer un très grand nombre d'images synthétiques simulant des images réelles. Les caractéristiques des régions rectangulaires contenant les informations d'intérêts sont stockées dans une base de données et un autre programme peut en disposer afin de construire un jeu de données synthétiques approprié.

Conception et développement des principaux modèles d'un système de traitement d'images de documents dans lequel les interventions humaines sont limitées

Ce système est fondé sur :

— une méthode pour la détermination de régions prototypes à partir du partitionnement d'un jeu de données synthétiques initial. D'une part, le partitionnement de données s'inscrivant dans un contexte de classification non supervisée, aucune connaissance à priori sur les données n'est nécessaire. D'autre part, les régions prototypes obtenues à partir des classes du partitionnement du jeu de données initial, sont des régions rectangulaires identifiables au sein d'une image de document. Par ailleurs, elles permettent de retrouver de manière précise une information textuelle encapsulée. De plus, ces régions prototypes sont suffisamment génériques pour permettre la localisation et l'extraction d'une information textuelle d'intérêt au sein d'images de documents d'émetteurs et de mises en page variés.

— une méthode pour la détermination de chemins à partir du treillis de concepts dérivé d'un contexte formel convenablement construit. Tout d'abord, à partir d'un ensemble d'observations et d'un ensemble de régions prototypes, un contexte formel approprié est construit. Ensuite, à partir du treillis de concepts dérivé du contexte formel obtenu, notre méthode consiste à déterminer des chemins permettant de naviguer dans l'ensemble des régions prototypes. L'ensemble des chemins obtenus constitue une stratégie de parcours des régions prototypes permettant de cibler des régions prototypes à visiter prioritairement. Cette stratégie de parcours permet de limiter le nombre de régions prototypes à visiter, réduisant ainsi le temps de traitement d'une image.

— une méthode pour la mise à jour incrémentale d'un ensemble de régions prototypes à partir de nouvelles données. Notre méthode consiste à adopter un par-

titionnement incrémental d'un jeu de données initial dans lequel est insérée une nouvelle donnée. Ensuite, à partir des classes obtenues de ce partitionnement, de nouvelles régions prototypes, prenant en compte la nouvelle donnée, peuvent être déterminées. Notre méthode permet, ainsi, d'enrichir notre système et contribue également à le rendre utilisable pour un plus grand nombre d'images hétérogènes.
— une méthode pour la mise à jour incrémentale de l'ensemble des chemins pour naviguer au sein d'un ensemble de régions prototypes. Notre méthode consiste à mettre à jour, si nécessaire, un ensemble de chemins existants, suite à la mise à jour de l'ensemble des régions prototypes. De cette manière, le système dispose toujours de chemins appropriés pour le parcours efficace d'un ensemble de régions prototypes.

Pour chacune de nos contributions, des expérimentations ont été menées. En particulier, l'évaluation de notre système et de ses différents composants a montré des résultats prometteurs pour l'extraction d'informations textuelles au sein d'images de factures.

Perspectives

Les travaux présentés dans ce mémoire soulèvent quelques axes d'étude pour des travaux futurs, afin d'améliorer les résultats obtenus dans cette thèse.

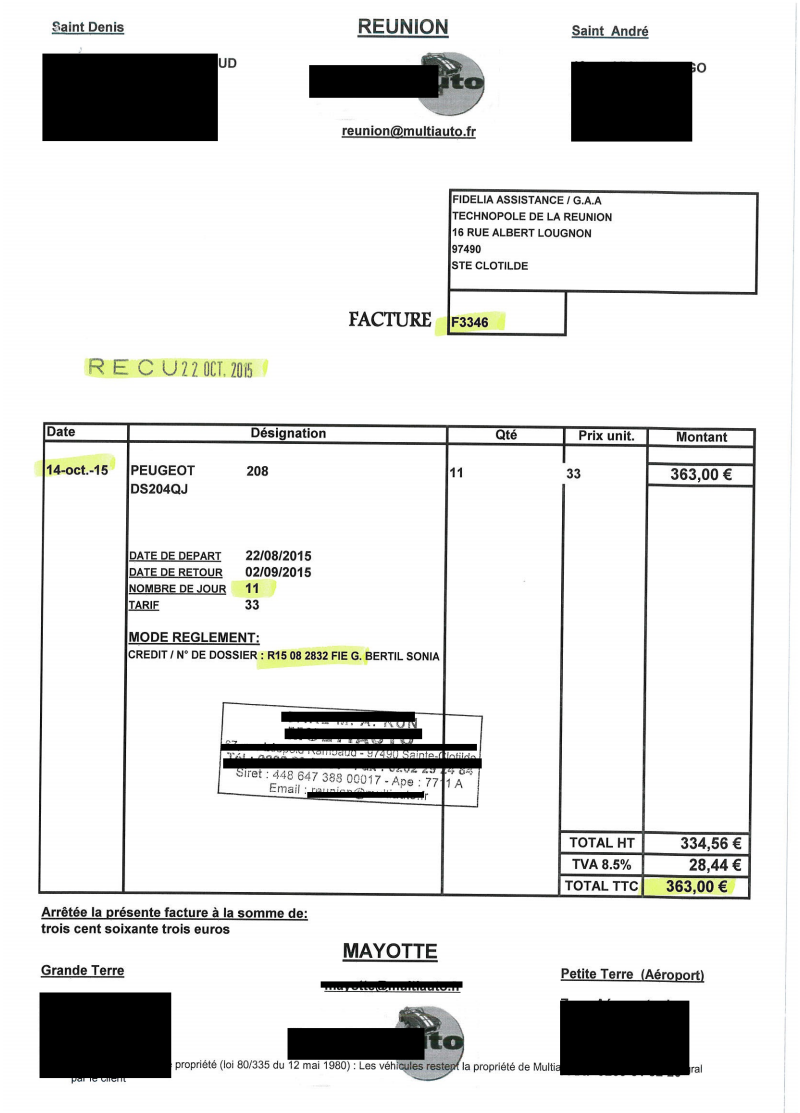
1. Concernant le partitionnement d'un jeu de données synthétiques présenté dans ce mémoire, il serait intéressant d'étudier la piste d'utilisation d'une mesure de (dis)similarité conçue spécifiquement pour des régions rectangulaires. En effet, Diatta (Diatta, 2003) présente plusieurs mesures de (dis)similarité fondées sur le contenu, permettant de mesurer une distance entre des polygones dans le cas général. Nous avons implémenté cette mesure dans le langage **R**. Nous avons également tenté de l'utiliser comme mesure de distance pour un partitionnement de données avec k-means. Les difficultés algorithmiques rencontrées, pour le partitionnement d'un grand jeu de données synthétiques ne nous ont pas permis d'approfondir cette piste.
2. Concernant le système d'extraction d'informations textuelles que nous présentons dans ce mémoire, il serait intéressant de mettre en place des liaisons entre les composants du système, afin qu'ils puissent interagir ensemble de manière automatisée. En effet, notre système est fondé sur (i) un programme de génération d'images synthétiques développé par nos soins en JAVA, (iii) un partitionnement de données réalisé avec **R**, (ii) un treillis de concepts dérivé d'un contexte formel d'images de factures synthétiques obtenu avec conexp et (iv) un programme développé par nos soins en JAVA pour la mise en œuvre du processus d'extraction d'informations prévu dans notre système. Actuellement, ces quatre sous-programmes ne communiquent pas entre eux de manière automatisée de part la variété d'outils tiers utilisés.
3. Dans ce mémoire, le moteur de reconnaissance utilisé est Tesseract OCR. Nous avons pu observer dans nos différentes expérimentations les faiblesses de ce moteur de reconnaissance. Il serait intéressant d'explorer d'autres solutions de reconnaissances plus performantes afin d'améliorer les résultats obtenus dans cette thèse.
4. Dans ce mémoire, nous avons traité exclusivement des images de documents d'une

page. Le traitement efficace d'images de documents de plus d'une page est également une piste de réflexion.

Annexe A

Exemple de facture émise par une société de location

Les informations contenues dans ce document sont confidentielles et ne doivent pas être utilisées en dehors du cadre prévu par cette thèse.



Saint Denis

[redacted]UD

REUNION

[redacted]

reunion@multiauto.fr

Saint André

[redacted]SO

FIDELIA ASSISTANCE / G.A.A
TECHNOPOLE DE LA REUNION
16 RUE ALBERT LOUGNON
97490
STE CLOTILDE

FACTURE F3346

RECU 22 OCT. 2015

Date	Désignation	Qté	Prix unit.	Montant
14-oct.-15	PEUGEOT 208 DS204QJ	11	33	363,00 €

DATE DE DEPART 22/08/2015
DATE DE RETOUR 02/09/2015
NOMBRE DE JOUR 11
TARIF 33

MODE REGLEMENT:
CREDIT / N° DE DOSSIER : R15 08 2832 FIE G. BERTIL SONIA

[redacted]
Siret : 448 647 388 00017 - Ape : 7711 A
Email : [redacted]

TOTAL HT	334,56 €
TVA 8.5%	28,44 €
TOTAL TTC	363,00 €

Arrêtée la présente facture à la somme de:
trois cent soixante trois euros

MAYOTTE

Grande Terre

[redacted]

Petite Terre (Aéroport)

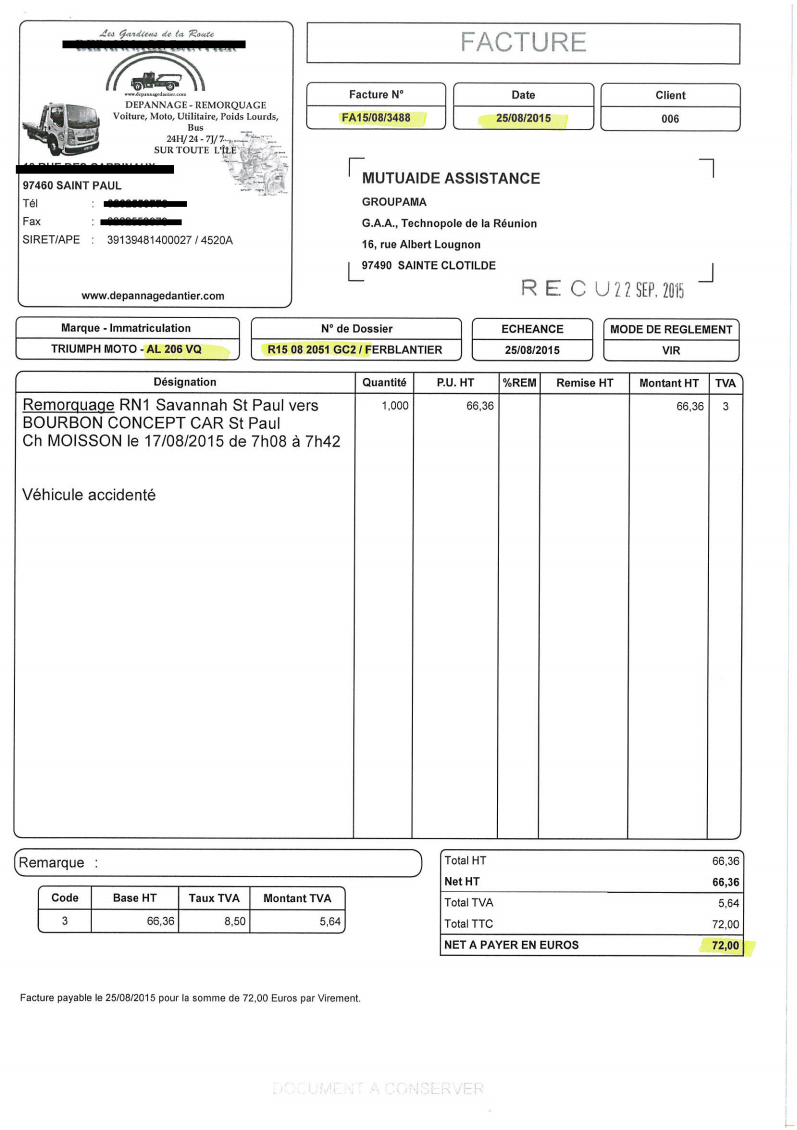
[redacted]

[redacted] propriété (loi 80/335 du 12 mai 1980) : Les véhicules restent la propriété de Multia[redacted]ral

Annexe B

Exemple de facture émise par une société de remorquage

Les informations contenues dans ce document sont confidentielles et ne doivent pas être utilisées en dehors du cadre prévu par cette thèse.



Les Gardiens de la Route

[redacted]

DEPANNAGE - REMORQUAGE
Voiture, Moto, Utilitaire, Poids Lourds, Bus
24H/24 - 7J/7
SUR TOUTE L'ILE

[redacted]
97460 SAINT PAUL
Tél : [redacted]
Fax : [redacted]
SIRET/APE : 39139481400027 / 4520A

www.depannagedantier.com

FACTURE

Facture N°	Date	Client
FA15/08/3488	25/08/2015	006

MUTUAIDE ASSISTANCE
GROUPAMA
G.A.A., Technopole de la Réunion
16, rue Albert Lougnon
97490 SAINTE CLOTILDE

REÇU 22 SEP. 2015

Marque - Immatriculation	N° de Dossier	ECHEANCE	MODE DE REGLEMENT
TRIUMPH MOTO - AL 206 VQ	R15 08 2051 GC2 / FERBLANTIER	25/08/2015	VIR

Désignation	Quantité	P.U. HT	%REM	Remise HT	Montant HT	TVA
Remorquage RN1 Savannah St Paul vers BOURBON CONCEPT CAR St Paul Ch MOISSON le 17/08/2015 de 7h08 à 7h42 Véhicule accidenté	1,000	66,36			66,36	3

Remarque :

Code	Base HT	Taux TVA	Montant TVA
3	66,36	8,50	5,64

Total HT	66,36
Net HT	**66,36**
Total TVA	5,64
Total TTC	72,00
NET A PAYER EN EUROS	**72,00**

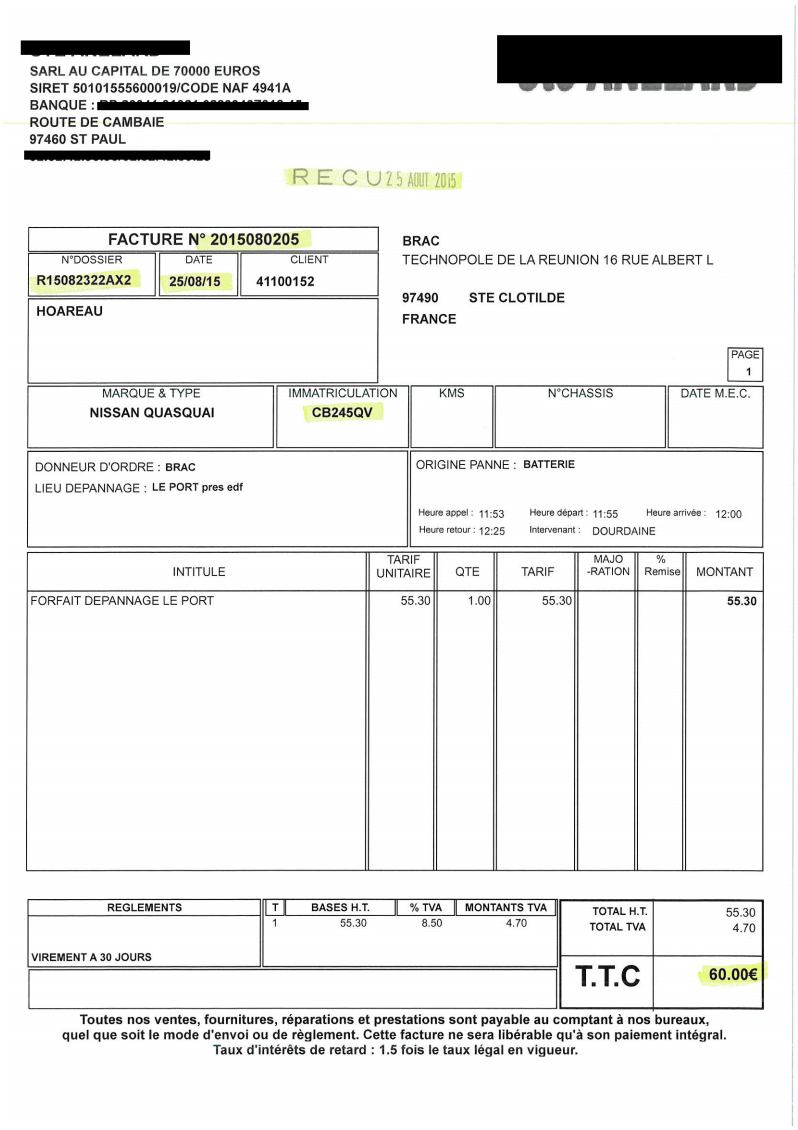
Facture payable le 25/08/2015 pour la somme de 72,00 Euros par Virement.

DOCUMENT A CONSERVER

Annexe C

Exemple de facture émise par une société de dépannage

Les informations contenues dans ce document sont confidentielles et ne doivent pas être utilisées en dehors du cadre prévu par cette thèse.



[redacted]
SARL AU CAPITAL DE 70000 EUROS
SIRET 50101555600019/CODE NAF 4941A
BANQUE : [redacted]
ROUTE DE CAMBAIE
97460 ST PAUL
[redacted]

[redacted]

REÇU 25 AOUT 2015

FACTURE N° 2015080205

N°DOSSIER	DATE	CLIENT
R15082322AX2	25/08/15	41100152

HOAREAU

BRAC
TECHNOPOLE DE LA REUNION 16 RUE ALBERT L

97490 STE CLOTILDE
FRANCE

PAGE 1

MARQUE & TYPE	IMMATRICULATION	KMS	N°CHASSIS	DATE M.E.C.
NISSAN QUASQUAI	CB245QV			

DONNEUR D'ORDRE : BRAC
LIEU DEPANNAGE : LE PORT pres edf

ORIGINE PANNE : BATTERIE

Heure appel : 11:53 Heure départ : 11:55 Heure arrivée : 12:00
Heure retour : 12:25 Intervenant : DOURDAINE

INTITULE	TARIF UNITAIRE	QTE	TARIF	MAJO -RATION	% Remise	MONTANT
FORFAIT DEPANNAGE LE PORT	55.30	1.00	55.30			**55.30**

REGLEMENTS	T	BASES H.T.	% TVA	MONTANTS TVA
VIREMENT A 30 JOURS	1	55.30	8.50	4.70

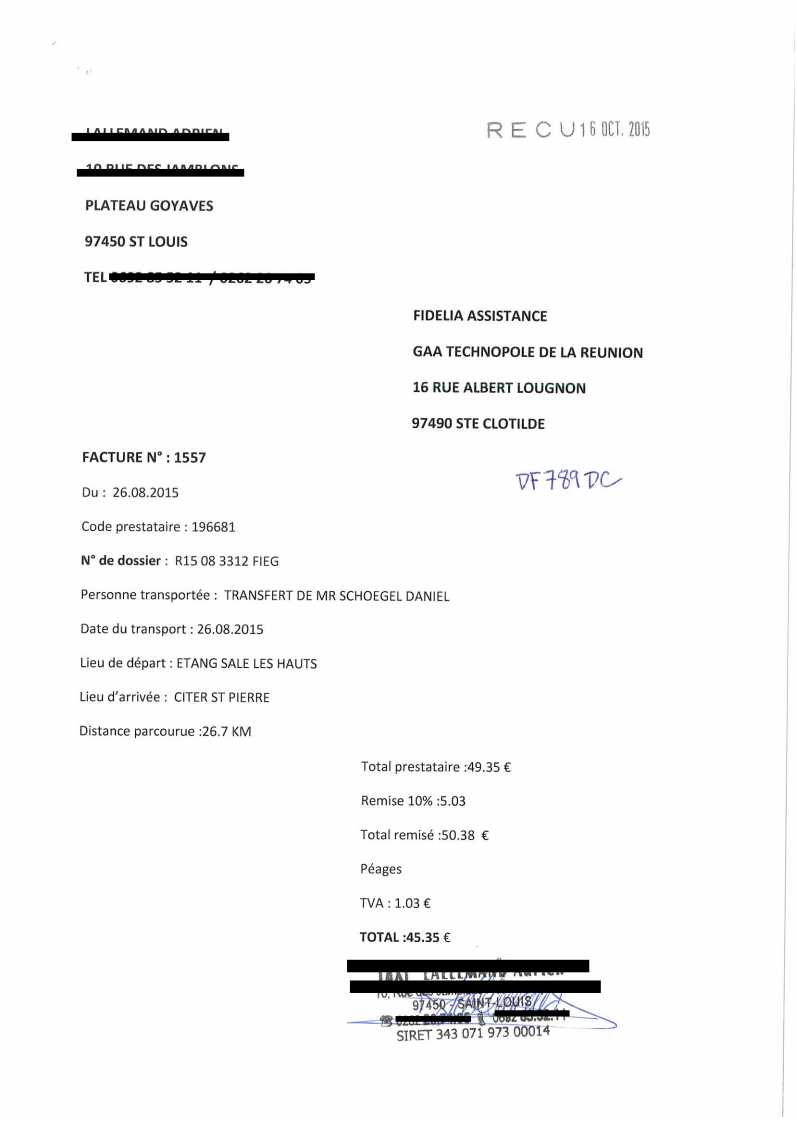
TOTAL H.T.	55.30
TOTAL TVA	4.70
T.T.C	**60.00€**

Toutes nos ventes, fournitures, réparations et prestations sont payable au comptant à nos bureaux, quel que soit le mode d'envoi ou de règlement. Cette facture ne sera libérable qu'à son paiement intégral. Taux d'intérêts de retard : 1.5 fois le taux légal en vigueur.

Annexe D

Exemple de facture émise par une société de taxi

Les informations contenues dans ce document sont confidentielles et ne doivent pas être utilisées en dehors du cadre prévu par cette thèse.



[redacted]

[redacted]

PLATEAU GOYAVES

97450 ST LOUIS

TEL [redacted]

RECU 16 OCT. 2015

FIDELIA ASSISTANCE

GAA TECHNOPOLE DE LA REUNION

16 RUE ALBERT LOUGNON

97490 STE CLOTILDE

FACTURE N° : 1557

DF 789 DC

Du : 26.08.2015

Code prestataire : 196681

N° de dossier : R15 08 3312 FIEG

Personne transportée : TRANSFERT DE MR SCHOEGEL DANIEL

Date du transport : 26.08.2015

Lieu de départ : ETANG SALE LES HAUTS

Lieu d'arrivée : CITER ST PIERRE

Distance parcourue :26.7 KM

Total prestataire :49.35 €

Remise 10% :5.03

Total remisé :50.38 €

Péages

TVA : 1.03 €

TOTAL :45.35 €

[redacted]

97450 SAINT-LOUIS

SIRET 343 071 973 00014

Bibliographie

Aamodt, Agnar, & Plaza, Enric. 1994. Case-based reasoning : Foundational issues, methodological variations, and system approaches. *AI communications*, **7**(1), 39–59.

Adelson, Edward H, Anderson, Charles H, Bergen, James R, Burt, Peter J, & Ogden, Joan M. 1984. Pyramid methods in image processing. *RCA engineer*, **29**(6), 33–41.

Agrawal, Paraag, & Varma, Rohit. 2012. Text extraction from images. *IJCSET*, **2**(4), 1083–1087.

Agrawal, Rakesh, Imieliński, Tomasz, & Swami, Arun. 1993. Mining association rules between sets of items in large databases. *Pages 207–216 of : Acm sigmod record*, vol. 22. ACM.

Aiello, Marco, Monz, Christof, Todoran, Leon, & Worring, Marcel. 2003. Document understanding for a broad class of documents. *International Journal on Document Analysis and Recognition*, **5**(1), 1–16.

Akaike, Hirotogu. 1992. Information theory and an extension of the maximum likelihood principle. *Pages 610–624 of : Breakthroughs in statistics*. Springer.

Akiyama, Teruo, & Hagita, Norihiro. 1990. Automated entry system for printed documents. *Pattern recognition*, **23**(11), 1141–1154.

Alonso-Montesinos, J, Martínez-Durbán, M, del Sagrado, J, del Águila, IM, & Batlles, FJ. 2016. The application of Bayesian network classifiers to cloud classification in satellite images. *Renewable Energy*, **97**, 155–161.

Anjewierden, Anjo. 2001. AIDAS : Incremental logical structure discovery in PDF documents. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, **2001-January**, 374–378.

Antonacopoulos, Apostolos, Karatzas, Dimosthenis, & Bridson, David. 2006. Ground truth for layout analysis performance evaluation. *Pages 302–311 of : International Workshop on Document Analysis Systems*. Springer.

Aoun-Allah, Mohamed, & Mineau, Guy. 2006. Rule validation of a meta-classifier through a Galois (concept) lattice and complementary means. *Pages 123–138 of : Proceedings of the 4th international conference on Concept lattices and their applications*. Springer-Verlag.

Atzori, L., De Natale, F., & Granelli, F. 2001. A real-time visual postprocessor for MPEG-coded video sequences. *Signal Processing : Image Communication*, **16**(8), 809–816.

Aurenhammer, Franz. 1991. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, **23**(3), 345–405.

Azmeh, Zeina, Huchard, Marianne, Tibermacine, Chouki, Urtado, Christelle, & Vauttier, Sylvain. 2008. Wspab : A tool for automatic classification & selection of web services using formal concept analysis. *Pages 31–40 of : on Web Services, 2008. ECOWS'08. IEEE Sixth European Conference*. IEEE.

Babu, G Phanendra, & Murty, M Narasimha. 1993. A near-optimal initial seed value selection in k-means means algorithm using a genetic algorithm. *Pattern Recognition Letters*, **14**(10), 763–769.

Baik, Sung, & Bala, Jerzy. 2004. A decision tree algorithm for distributed data mining : Towards network intrusion detection. *Computational Science and Its Applications–ICCSA 2004*, 206–212.

Baird, Henry S, Jones, Susan E, & Fortune, Steven J. 1990. Image segmentation by shape-directed covers. *Pages 820–825 of : Pattern Recognition, 1990. Proceedings., 10th International Conference on*, vol. 1. IEEE.

Baker, Matthew J, Trevisan, Júlio, Bassan, Paul, Bhargava, Rohit, Butler, Holly J, Dorling, Konrad M, Fielden, Peter R, Fogarty, Simon W, Fullwood, Nigel J, Heys, Kelly A, *et al.* 2014. Using Fourier transform IR spectroscopy to analyze biological materials. *Nature protocols*, **9**(8), 1771–1791.

Ball, Geoffrey H, & Hall, David J. 1965. *ISODATA, a novel method of data analysis and pattern classification.* Tech. rept. Stanford research inst Menlo Park CA.

Barbut, M, & Monjardet, B. 1970. Ordre et classification, Algèbre et combinatoire,. *Zbl0267*, **6001**.

Barraquand, J., & Latombe, J.C. 1991. Robot motion planning : A distributed representation approach. *The International Journal of Robotics Research*, **10**(6), 628–649.

Bartoli, Alberto, Davanzo, Giorgio, Medvet, Eric, & Sorio, Enrico. 2010. Improving features extraction for supervised invoice classification. *Page 401 of : Proceedings of the 10th IASTED International Conference*, vol. 674.

Bartoli, Alberto, Davanzo, Giorgio, Medvet, Eric, & Sorio, Enrico. 2014. Semisupervised wrapper choice and generation for print-oriented documents. *IEEE Transactions on Knowledge and Data Engineering*, **26**(1), 208–220.

Battiti, Roberto, & Masulli, Francesco. 1990. BFGS optimization for faster and automated supervised learning. *Pages 757–760 of : International neural network conference.* Springer.

Belaïd, Abdel, DAndecy, Vincent Poulain, Hamza, Hatem, & Belaïd, Yolande. 2011. Administrative document analysis and structure. *Learning Structure and Schemas from Documents*, 51–71.

Belohlavek, Radim, & Vychodil, Vilem. 2009. Factor analysis of incidence data via novel decomposition of matrices. *Formal Concept Analysis*, 83–97.

Belohlavek, Radim, & Vychodil, Vilem. 2010. Discovery of optimal factors in binary data via a novel method of matrix decomposition. *Journal of Computer and System Sciences*, **76**(1), 3–20.

Belohlavek, Radim, De Baets, Bernard, Outrata, Jan, & Vychodil, Vilem. 2009. Inducing decision trees via concept lattices. *International journal of general systems*, **38**(4), 455–467.

Benayade, Mohammed, & Diatta, Jean. 2008. Cluster structures and collections of Galois closed entity subsets. *Discrete Applied Mathematics*, **156**(8), 1295–1307.

Beydoun, Ghassan. 2008. Using formal concept analysis towards cooperative e-learning. *Pages 109–117 of : Pacific Rim Knowledge Acquisition Workshop*. Springer.

Beydoun, Ghassan, Kultchitsky, Roman, & Manasseh, Grace. 2007. Evolving semantic web with social navigation. *Expert Systems with Applications*, **32**(2), 265–276.

Bezdek, James C. 1981. Pattern recognition with fuzzy objective function algorithms.

Bhatia, Nitin, *et al.* 2010. Survey of nearest neighbor techniques. *arXiv preprint arXiv :1007.0085*.

Bielza, Concha, & Larrañaga, Pedro. 2014. Discrete Bayesian network classifiers : a survey. *ACM Computing Surveys (CSUR)*, **47**(1), 5.

Birkhoff, Garrett. 1940. *Lattice theory*. Vol. 25. American Mathematical Soc.

Bishop, Christopher M. 2006. *Pattern recognition and machine learning*. springer.

Bissacco, Alessandro, Cummins, Mark, Netzer, Yuval, & Neven, Hartmut. 2013. Photoocr : Reading text in uncontrolled conditions. *Pages 785–792 of : Proceedings of the IEEE International Conference on Computer Vision*.

Blinova, VG, Dobrynin, DA, Finn, VK, Kuznetsov, Sergei O, & Pankratova, ES. 2003. Toxicology analysis by means of the JSM-method. *Bioinformatics*, **19**(10), 1201–1207.

Bock, Hans H. 1996. Probabilistic models in cluster analysis. *Computational Statistics & Data Analysis*, **23**(1), 5–28.

Boujemaa, N., Fauqueur, J., Ferecatu, M., Fleuret, F., Gouet, V., LeSaux, B., & Sahbi, H. 2001. Ikona : Interactive specific and generic image retrieval. *In : Proceedings of International workshop on Multimedia Content-Based Indexing and Retrieval (MMCBIR 2001)*.

Bowman, KO, & Shenton, LR. 2004. Estimation : Method of moments. *Encyclopedia of statistical sciences*.

Bradley, Paul S, & Fayyad, Usama M. 1998. Refining Initial Points for K-Means Clustering. *Pages 91–99 of : ICML*, vol. 98.

Brauer, Falk, Rieger, Robert, Mocan, Adrian, & Barczynski, Wojciech M. 2011. Enabling information extraction by inference of regular expressions from sample entities. *Pages 1285–1294 of : Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM.

Breiman, Leo, Friedman, Jerome H, Olshen, Richard A, & Stone, Charles J. 1984. Classification and regression trees. Wadsworth & Brooks. *Monterey, CA*.

Breslow, Leonard A, & Aha, David W. 1997. Simplifying decision trees : A survey. *The Knowledge Engineering Review*, **12**(1), 1–40.

Brown, Donald E, & Huntley, Christopher L. 1992. A practical application of simulated annealing to clustering. *Pattern Recognition*, **25**(4), 401–412.

Brucker, François, & Barthélemy, Jean-Pierre. 2007. *Eléments de classification : aspects combinatoires et algorithmiques*. Hermes sciences publications.

Bruha, Ivan. 2000. From machine learning to knowledge discovery : Survey of preprocessing and postprocessing. *Intelligent Data Analysis*, **4**(3, 4), 363–374.

Bunke, Horst. 2003. Recognition of cursive Roman handwriting : past, present and future. *Pages 448–459 of : Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*. IEEE.

Burges, Christopher JC. 1998. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, **2**(2), 121–167.

Caliński, Tadeusz, & Harabasz, Jerzy. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, **3**(1), 1–27.

Campos, Cassio P de, & Ji, Qiang. 2011. Efficient structure learning of Bayesian networks using constraints. *Journal of Machine Learning Research*, **12**(Mar), 663–689.

Carpineto, Claudio, & Romano, Giovanni. 1993. Galois : An order-theoretic approach to conceptual clustering. *Pages 33–40 of : Proceedings of ICML*, vol. 293.

Carpineto, Claudio, & Romano, Giovanni. 1996. A lattice conceptual clustering system and its application to browsing retrieval. *Machine learning*, **24**(2), 95–122.

Carpineto, Claudio, & Romano, Giovanni. 2004. *Concept data analysis : Theory and applications*. John Wiley & Sons.

Carpineto, Claudio, Romano, Giovanni, & d'Adamo, Paolo. 1999. Inferring dependencies from relations : a conceptual clustering approach. *Computational Intelligence*, **15**(4), 415–441.

Carpineto, Claudio, Romano, Giovanni, & Bordoni, Fondazione Ugo. 2004. Exploiting the potential of concept lattices for information retrieval with CREDO. *J. UCS*, **10**(8), 985–1013.

Caruana, Rich, Lawrence, Steve, & Giles, C Lee. 2001. Overfitting in neural nets : Backpropagation, conjugate gradient, and early stopping. *Pages 402–408 of : Advances in neural information processing systems*.

Cesarini, F., Francesconi, E., Gori, M., & Soda, G. 2003. Analysis and understanding of multi-class invoices. *International Journal on Document Analysis and Recognition*, **6**(2), 102–114.

Cesarini, Francesca, Gori, Marco, Marinai, Simone, & Soda, Giovanni. 1999. Structured document segmentation and representation by the modified XY tree. *Pages 563–566 of : Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on.* IEEE.

Cesarini, Francesca, Marinai, Simone, Sarti, L, & Soda, Giovanni. 2002. Trainable table location in document images. *Pages 236–240 of : Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 3. IEEE.

Chakraborty, Sanjay, & Nagwani, NK. 2011. Analysis and study of incremental k-means clustering algorithm. *High performance architecture and grid computing*, 338–341.

Chakraborty, Sanjay, Nagwani, NK, & Dey, Lopamudra. 2014. Performance comparison of incremental k-means and incremental dbscan algorithms. *arXiv preprint arXiv :1406.4751.*

Chen, Chi-hau. 2015. *Handbook of pattern recognition and computer vision.* World Scientific.

Chen, Datong, Odobez, Jean-Marc, & Bourlard, Herve. 2004. Text detection and recognition in images and video frames. *Pattern recognition*, **37**(3), 595–608.

Chen, Nawei, & Blostein, Dorothea. 2007. A survey of document image classification : problem statement, classifier architecture and performance evaluation. *International Journal on Document Analysis and Recognition*, **10**(1), 1–16.

Cheng, Jie, Grainer, G, Kelly, J, Bell, DA, & Lius, W. 2001. Learning bayesian networks from data : An information-theory based approach. *URL citeseer. ist. psu. edu/628344. html.*

Chickering, David Maxwell. 2002. Optimal structure identification with greedy search. *Journal of machine learning research*, **3**(Nov), 507–554.

Cho, Woo-Chul, & Richards, Debbie. 2004. Improvement of precision and recall for information retrieval in a narrow domain : reuse of concepts by formal concept analysis. *Pages 370–376 of : Proceedings of the 2004 IEEE/WIC/ACM international conference on web intelligence.* IEEE Computer Society.

Choi, Vicky, Huang, Yang, Lam, Vy, Potter, Dustin, Laubenbacher, Reinhard, & Duca, Karen. 2008. Using formal concept analysis for microarray data comparison. *Journal of bioinformatics and computational biology*, **6**(01), 65–75.

Chollet, Stéphanie, Lestideau, Vincent, Maurel, Yoann, Gandrille, Etienne, Lalanda, Philippe, & Raynaud, Olivier. 2012. Practical use of formal concept analysis in service-oriented computing. *Pages 61–76 of : International Conference on Formal Concept Analysis.* Springer.

Ciardiello, G, Scafuro, G, Degrandi, MT, Spada, MR, & Roccotelli, MP. 1988. An experimental system for office document handling and text recognition. *Pages 739–743 of : Proc 9th Int. Conf. on Pattern Recognition.*

Cimiano, Philipp, Hotho, Andreas, & Staab, Steffen. 2005. Learning concept hierarchies from text corpora using formal concept analysis. *J. Artif. Intell. Res.(JAIR)*, **24**(1), 305–339.

Cole, Richard, & Eklund, Peter. 2001. Browsing semi-structured web texts using formal concept analysis. *Conceptual Structures : Broadening the Base*, 319–332.

Cole, Richard, & Stumme, Gerd. 2000. CEM-a conceptual email manager. *Pages 438–452 of : ICCS*, vol. 1867. Springer.

Constantinou, Anthony Costa, Fenton, Norman, Marsh, William, & Radlinski, Lukasz. 2016. From complex questionnaire and interviewing data to intelligent Bayesian network models for medical decision support. *Artificial intelligence in medicine*, **67**, 75–93.

Conway, Alan. 1993. Page grammars and page parsing. a syntactic approach to document layout recognition. *Pages 761–764 of : Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*. IEEE.

Coüasnon, Bertrand. 2006. DMOS, a generic document recognition method : Application to table structure analysis in a general and in a specific way. *International Journal on Document Analysis and Recognition*, **8**(2), 111–122.

Cover, Thomas M, & Thomas, Joy A. 2006. Elements of information theory 2nd edition.

Cristianini, Nello, & Shawe-Taylor, John. 2000. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

Cuvelier, Etienne, & Aufaure, Marie-Aude. 2011. A buzz and e-reputation monitoring tool for twitter based on galois lattices. *Conceptual Structures for Discovering Knowledge*, 91–103.

Dagher, Issam, & Taleb, Catherine. 2014. Image denoising using fourth order wiener filter with wavelet quadtree decomposition. *Journal of Electrical and Computer Engineering*, **2014**.

Defays, Daniel. 1977. An efficient algorithm for a complete link method. *The Computer Journal*, **20**(4), 364–366.

Dempster, Arthur P, Laird, Nan M, & Rubin, Donald B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, 1–38.

Demuth, Howard B, Beale, Mark H, De Jess, Orlando, & Hagan, Martin T. 2014. *Neural network design*. Martin Hagan.

Dengel, Andreas. 2003. Making documents work : challenges for document understanding. *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*

Dengel, Andreas, & Dubiel, Frank. 1996. Computer understanding of document structure. *International Journal of Imaging Systems and Technology*, **7**(4), 271–278.

Dharani, T, & Aroquiaraj, I Laurence. 2013. A survey on content based image retrieval. *Pages 485–490 of : Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on*. IEEE.

Diatta, Jean. 2003. Génération de la base de Guigues-Duquenne-Luxenburger pour les règles d'association par une approche utilisant des mesures de similarité multivoies. *Conférence d'Apprentissage*, 281–298.

Dietterich, Thomas G. 2000. An experimental comparison of three methods for constructing ensembles of decision trees : Bagging, boosting, and randomization. *Machine learning*, **40**(2), 139–157.

Dixit, Umesh D, & Shirdhonkar, MS. 2015. A Survey on Document Image Analysis and Retrieval System. *IJCI*, **4**(2), 259–270.

Doermann, David, Liang, Jian, & Li, Huiping. 2003. Progress in camera-based document image analysis. *Pages 606–616 of : Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*. IEEE.

Dong, Guozhu, Jiang, Chunyu, Pei, Jian, Li, Jinyan, & Wong, Limsoon. 2005. Mining succinct systems of minimal generators of formal concepts. *Pages 175–187 of : DASFAA*. Springer.

Doster, W. 1984. Different states of a document's content on its way from the gutenbergian world to the electronic world. *Pages 872–874 of : Proc. 7th Int. Conf. on Pattern Recognition*.

Ducrou, Jon, & Eklund, Peter. 2007. SearchSleuth : The Conceptual Neighbourhood of an Web Query. *Pages 249–259 of : Proc. CLA 2007*, vol. 331. Jean Diatta, Peter Eklund and Michel Liquiere, LIRMM & University of Montpellier II.

Egho, Elias, Jay, Nicolas, Raissi, Chedy, & Napoli, Amedeo. 2011. A FCA-based analysis of sequential care trajectories. *In : The Eighth International Conference on Concept Lattices and their Applications-CLA 2011*.

Eklund, Peter, Ducrou, Jon, & Brawn, Peter. 2004. Concept lattices for information visualization : Can novices read line-diagrams ? *Pages 57–73 of : International Conference on Formal Concept Analysis*. Springer.

El-Qawasmeh, E. 2003. A quadtree-based representation technique for indexing and retrieval of image databases. *Journal of Visual Communication and Image Representation*, **14**(3), 340 – 357.

Elomaa, Tapio. 1999. The biases of decision tree pruning strategies. *Pages 63–74 of : International Symposium on Intelligent Data Analysis*. Springer.

Elzinga, Paul, Wolff, Karl Erich, & Poelmans, Jonas. 2012. Analyzing chat conversations of pedophiles with temporal relational semantic systems. *Pages 242–249 of : Intelligence and Security Informatics Conference (EISIC), 2012 European*. IEEE.

Feelders, Ad, & Van der Gaag, Linda C. 2006. Learning Bayesian network parameters under order constraints. *International Journal of Approximate Reasoning*, **42**(1-2), 37–53.

Finkel, R.A., & Bentley, J.L. 1974. Quad trees : a data structure for retrieval on composite keys. *Acta Inform.*, **4**, 11–9.

Fletcher, Lloyd A., & Kasturi, Rangachar. 1988. A robust algorithm for text string separation from mixed text/graphics images. *IEEE transactions on pattern analysis and machine intelligence*, **10**(6), 910–918.

Föllmer, Hans. 1973. On entropy and information gain in random fields. *Probability Theory and Related Fields*, **26**(3), 207–217.

Forgy, Edward W. 1965. Cluster analysis of multivariate data : Efficiency vs. interpretability of classifications. *Biometrics*, **21**, 768–769.

Forrest, Stephanie, Hofmeyr, Steven A, & Somayaji, Anil. 1991. Emergent computation.

Fraley, Chris, & Raftery, Adrian E. 1998. How many clusters ? Which clustering method ? Answers via model-based cluster analysis. *The computer journal*, **41**(8), 578–588.

Friedman, Nir. 1998. The Bayesian structural EM algorithm. *Pages 129–138 of : Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc.

Friedman, Nir, Geiger, Dan, & Goldszmidt, Moises. 1997. Bayesian network classifiers. *Machine learning*, **29**(2-3), 131–163.

Fu, Huaiyu, Fu, Huaiguo, Njiwoua, Patrik, & Nguifo, Engelbert. 2004. A comparative study of fca-based supervised classification algorithms. *Concept Lattices*, 219–220.

Fujisawa, Hiromichi, & Nakano, Yasuaki. 1992. A top-down approach to the analysis of document images. *Pages 99–114 of : Structured Document Image Analysis*. Springer.

Gama, João, & Brazdil, Pavel. 1999. Linear tree. *Intelligent Data Analysis*, **3**(1), 1–22.

Ganter, Bernhard, & Kuznetsov, Sergei O. 2000. Formalizing hypotheses with concepts. *Pages 342–356 of : ICCS*. Springer.

Ganter, Bernhard, & Kuznetsov, Sergei O. 2003. Hypotheses and version spaces. *Pages 83–95 of : ICCS*, vol. 2746. Springer.

Ganter, Bernhard, & Wille, Rudolf. 1999. *Formal concept analysis : mathematical foundations*. Springer.

Gao, Hongxing, Rusinol, Marcal, Karatzas, Dimosthenis, Llados, Josep, Sato, Tomokazu, Iwamura, Masakazu, & Kise, Koichi. 2013. Key-region detection for document images - Application to administrative document retrieval. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 230–234.

Gehrke, Johannes, Ramakrishnan, Raghu, & Ganti, Venkatesh. 2000. RainForest—a framework for fast decision tree construction of large datasets. *Data Mining and Knowledge Discovery*, **4**(2), 127–162.

Geiger, Dan, Verma, Tom S, & Pearl, Judea. 2013. d-separation : From theorems to algorithms. *arXiv preprint arXiv :1304.1505*.

Gelman, Andrew, Carlin, John B, Stern, Hal S, Dunson, David B, Vehtari, Aki, & Rubin, Donald B. 2014. *Bayesian data analysis*. Vol. 2. CRC press Boca Raton, FL.

Ghosh, Joydeep. 2003. Scalable clustering. *The handbook of data mining, New Jersey : Lawrence Erlbaum Associates Publisher*, 247–277.

Glover, Fred, & Laguna, Manuel. 2013. Tabu Search - Part II. *Pages 3261–3362 of : Handbook of Combinatorial Optimization*. Springer.

Godin, Robert, & Mili, Hafedh. 1993. Building and maintaining analysis-level class hierarchies using galois lattices. *Pages 394–410 of : ACM SIGplan Notices*, vol. 28. ACM.

Godin, Robert, Pichet, C, & Gecsei, J. 1989. Design of a browsing interface for information retrieval. *Pages 32–39 of : ACM SIGIR Forum*, vol. 23. ACM.

Godin, Robert, Missaoui, Rokia, & April, Alain. 1993. Experimental comparison of navigation in a Galois lattice with conventional information retrieval methods. *International Journal of Man-Machine Studies*, **38**(5), 747–767.

Goebel, Michael, & Gruenwald, Le. 1999. A survey of data mining and knowledge discovery software tools. *ACM SIGKDD explorations newsletter*, **1**(1), 20–33.

Goel, Akash, & Sharma, Yogesh Kumar. 2014. Text Extraction of Vehicle Number Plate and Document Images Using Discrete Wavelet Transform in MATLAB. *IOSR Journal of Computer Engineering (IOSR-JCE)*, **16**(2), 2278–0661.

Goldstein, Jade, Mittal, Vibhu, Carbonell, Jaime, & Kantrowitz, Mark. 2000. Multi-document summarization by sentence extraction. *Pages 40–48 of : Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic summarization-Volume 4*. Association for Computational Linguistics.

Gorski, Nikolai, Anisimov, Valery, Augustin, Emmanuel, Baret, Olivier, & Maximov, Sergey. 2001. Industrial bank check processing : the A2iA CheckReader TM. *International Journal on Document Analysis and Recognition*, **3**(4), 196–206.

Grosicki, Emmanuèle, & El Abed, Haikal. 2009. ICDAR 2009 handwriting recognition competition. *Pages 1398–1402 of : Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE.

Guha, Sudipto, Rastogi, Rajeev, & Shim, Kyuseok. 1998. CURE : an efficient clustering algorithm for large databases. *Pages 73–84 of : ACM Sigmod Record*, vol. 27. ACM.

Guha, Sudipto, Rastogi, Rajeev, & Shim, Kyuseok. 2000. ROCK : A robust clustering algorithm for categorical attributes. *Information systems*, **25**(5), 345–366.

Guigues, Jean-Louis, & Duquenne, Vincent. 1986. Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Mathématiques et Sciences humaines*, **95**, 5–18.

Gupta, Anamika, Kumar, Naveen, & Bhatnagar, Vasudha. 2005. Incremental classification rules based on association rules using formal concept analysis. *Pages 11–20 of : MLDM*. Springer.

Hagan, Martin T, & Menhaj, Mohammad B. 1994. Training feedforward networks with the Marquardt algorithm. *IEEE transactions on Neural Networks*, **5**(6), 989–993.

Hamming, Richard W. 1950. Error detecting and error correcting codes. *Bell Labs Technical Journal*, **29**(2), 147–160.

Hamrouni, Tarek, Yahia, S Ben, & Slimani, Yahya. 2005a. Avoiding the itemset closure computation "pitfall". *Pages 46–59 of : CLA*, vol. 2005.

Hamrouni, Tarek, Yahia, Sadok Ben, & Slimani, Yahya. 2005b. Prince : An algorithm for generating rule bases without closure computations. *Pages 346–355 of : DaWaK*. Springer.

Hamrouni, Tarek, Yahia, Sadok Ben, & Nguifo, Engelbert Mephu. 2007. Towards a finer assessment of extraction contexts sparseness. *Pages 504–508 of : Database and Expert Systems Applications, 2007. DEXA'07. 18th International Workshop on*. IEEE.

Hamza, Hatem, Belaïd, Yolande, & Belaïd, Abdel. 2007. Case-Based Reasoning for Invoice Analysis and Recognition. 404–418.

Haralick, Robert M. 1994. Document image understanding : Geometric and logical layout. *Pages 385–390 of : CVPR*, vol. 94.

Hart, P. E., Nilsson, N. J., & Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, **4**(2), 100–107.

Hartigan, John A, & Hartigan, JA. 1975. *Clustering algorithms*. Vol. 209. Wiley New York.

Hartigan, John A, & Wong, Manchek A. 1979. Algorithm AS 136 : A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, **28**(1), 100–108.

Haykin, Simon S. 2001. *Neural networks : a comprehensive foundation*. Tsinghua University Press.

Heckerman, David, Meek, Christopher, & Cooper, Gregory. 2006. A Bayesian approach to causal discovery. *Innovations in Machine Learning*, 1–28.

Hennig, Christian, Meila, Marina, Murtagh, Fionn, & Rocci, Roberto. 2015. *Handbook of cluster analysis*. CRC Press.

Héroux, Pierre, Barbu, Eugen, Adam, Sébastien, & Trupin, Éric. 2007. Automatic ground-truth generation for document image analysis and understanding. *Pages 476–480 of : Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 1. IEEE.

Hsu, Chih-Wei, Chang, Chih-Chung, Lin, Chih-Jen, *et al.* 2003. A practical guide to support vector classification.

Huijuan, W., Yuan, Y., & Yuan, Q. 2011 (July). Application of Dijkstra algorithm in robot path-planning. *Pages 1067–1069 of : Mechanic Automation and Control Engineering (MACE), 2011 Second International Conference on*.

Hussain, Rafaqat, Gao, Hui, & Shaikh, Riaz Ahmed. 2016. Segmentation of connected characters in text-based CAPTCHAs for intelligent character recognition. *Multimedia Tools and Applications*, 1–15.

Inagaki, Kosaku, Kato, Toshikazu, Hiroshima, Tadashi, & Sakai, Toshiyuki. 1984. MACSYM : A hierarchical parallel image processing system for event-driven pattern understanding of documents. *Pattern Recognition*, **17**(1), 85–108.

Ingold, Rolf, & Armangil, Doga. 1991. A top-down document analysis method for logical structure recognition. *Pages 41–49 of : Proceedings of International Conference on Document Analysis and Recognition.*

Ishitani, Yasuto. 1999. Logical structure analysis of document images based on emergent computation. *Pages 189–192 of : Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on.* IEEE.

Iwaki, Osamu, Kida, Hirimi, & Arakawa, Hiroki. 1987. A segmentation method based on office document hierarchical structure. *Pages 759–763 of : Proceedings of the 1987 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2.

Izenman, Alan Julian. 2013. Linear discriminant analysis. *Pages 237–280 of : Modern multivariate statistical techniques.* Springer.

Jaakkola, Tommi S, & Jordan, Michael I. 2000. Bayesian parameter estimation via variational methods. *Statistics and Computing*, **10**(1), 25–37.

Jain, Anil K, Murty, M Narasimha, & Flynn, Patrick J. 1999. Data clustering : a review. *ACM computing surveys (CSUR)*, **31**(3), 264–323.

Jiang, Guoqian, Ogasawara, Katsuhiko, Endoh, Akira, & Sakurai, Tsunetaro. 2003. Context-based ontology building support in clinical domains using formal concept analysis. *International journal of medical informatics*, **71**(1), 71–81.

Johansson, Erik M, Dowla, Farid U, & Goodman, Dennis M. 1991. Backpropagation learning for multilayer feed-forward neural networks using the conjugate gradient method. *International Journal of Neural Systems*, **2**(04), 291–301.

Juneja, Komal, Verma, Akhilesh, Goel, Savita, & Goel, Swati. 2015. A survey on recent image indexing and retrieval techniques for low-level feature extraction in CBIR systems. *Pages 67–72 of : Computational Intelligence & Communication Technology (CICT), 2015 IEEE International Conference on.* IEEE.

Jung, Keechul, Kim, Kwang In, & Jain, Anil K. 2004. Text information extraction in images and video : a survey. *Pattern recognition*, **37**(5), 977–997.

Kanai, Junichi, Rice, Stephen V., Nartker, Thomas A., & Nagy, George. 1995. Automated evaluation of OCR zoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(1), 86–90.

Kandogan, Eser. 2001. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. *Pages 107–116 of : Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM.

Karypis, George, Han, Eui-Hong, & Kumar, Vipin. 1999. Chameleon : Hierarchical clustering using dynamic modeling. *Computer*, **32**(8), 68–75.

Kaufman, Leonard, & Rousseeuw, Peter J. 1990a. Finding groups in data : an introduction to cluster analysis. *Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, New York : Wiley.*

Kaufman, Leonard, & Rousseeuw, Peter J. 1990b. Partitioning around medoids (program pam). *Finding groups in data : an introduction to cluster analysis*, 68–125.

Kaytoue, Mehdi, Kuznetsov, Sergei O, Napoli, Amedeo, & Duplessis, Sébastien. 2011. Mining gene expression data with pattern structures in formal concept analysis. *Information Sciences*, **181**(10), 1989–2001.

Khakzad, Nima. 2015. Application of dynamic Bayesian network to risk analysis of domino effects in chemical infrastructures. *Reliability Engineering & System Safety*, **138**, 263–272.

Kim, Jongwoo, Le, Daniel X, & Thoma, George R. 2001. Automated labeling in document images. *Pages 111–122 of : Document Recognition and Retrieval.*

Kira, Kenji, & Rendell, Larry A. 1992. The feature selection problem : Traditional methods and a new algorithm. *Pages 129–134 of : AAAI*, vol. 2.

Klein, Bertin, & Dengel, Andreas R. 2003. Problem-adaptable document analysis and understanding for high-volume applications. *Document Analysis and Recognition*, **6**(3), 167–180.

Koester, Bjoern. 2006. Conceptual knowledge retrieval with fooca : Improving web search engine results with contexts and concept hierarchies. *Pages 176–190 of : Industrial Conference on Data Mining*, vol. 4065. Springer.

Koester, Bjoern, *et al.* 2005. Conceptual Knowledge Processing with Google. *Pages 178–183 of : LWA.*

Kohonen, Teuvo. 2013. Essentials of the self-organizing map. *Neural networks*, **37**, 52–65.

Kotsiantis, Sotiris B, Zaharakis, I, & Pintelas, P. 2007. *Supervised machine learning : A review of classification techniques.*

Kreich, Joachim, Luhn, Achim, & Maderlechner, Gerd. 1991. An experimental environment for model based document analysis. *Pages 50–58 of : Proceedings of International Conference on Document Analysis and Recognition.*

Krishnamoorthy, Mukkai, Nagy, George, Seth, Sharad, & Viswanathan, Mahesh. 1993. Syntactic segmentation and labeling of digitized pages from technical journals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**(7), 737–747.

Kumar, R Dinesh, & Vijayabhasker, R. 2016. Offline Sanskirthandwritten Character Recognition Framework Based on Multi Layerfeed Forward Network with Intelligent Character Recognition. *Asian Journal of Information Technology*, **15**(11), 1678–1685.

Kuznetsov, Sergei O. 2004a. Complexity of learning in concept lattices from positive and negative examples. *Discrete Applied Mathematics*, **142**(1), 111–125.

Kuznetsov, Sergei O. 2004b. Machine learning and formal concept analysis. *Pages 287–312 of : International Conference on Formal Concept Analysis*. Springer.

Kuznetsov, Sergei O, & Makhalova, Tatyana P. 2015. Concept Interestingness Measures : a Comparative Study. *Pages 59–72 of : CLA*, vol. 1466.

Kuznetsov, Sergei O, & Obiedkov, Sergei A. 2002. Comparing performance of algorithms for generating concept lattices. *Journal of Experimental & Theoretical Artificial Intelligence*, **14**(2-3), 189–216.

Lakhal, Lotfi, & Stumme, Gerd. 2005. Efficient mining of association rules based on formal concept analysis. *Formal concept analysis*, **3626**, 180–195.

Langley, Russell. 1971. *Practical statistics simply explained*. Courier Corporation.

Larsen, Bjornar, & Aone, Chinatsu. 1999. Fast and effective text mining using linear-time document clustering. *Pages 16–22 of : Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

Lee, Chang Ha, & Kanungo, Tapas. 2003. The architecture of trueviz : A ground-truth/metadata editing and visualizing toolkit. *Pattern recognition*, **36**(3), 811–825.

Lee, Ming Che, Chang, Jia Wei, & Hsieh, Tung Cheng. 2014. A grammar-based semantic similarity algorithm for natural language sentences. *The Scientific World Journal*, **2014**.

Lee, S-W, Park, J-S, & Tang, Y. 1993. Performance evaluation of nonlinear shape normalization methods for the recognition of large-set handwritten characters. *Pages 402–407 of : Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*. IEEE.

Leonard, James, & Kramer, MA. 1990. Improvement of the backpropagation algorithm for training neural networks. *Computers & Chemical Engineering*, **14**(3), 337–341.

Levine, Mark S. 1977. *Canonical analysis and factor comparison*. Vol. 6. Sage.

Li, Jingquan, Hnatow, Justin, Meier, Timothy, & Deloge, Stephen Patrick. 2014 (dec # " 16"). *System and method for document processing*. US Patent 8,910,870.

Lienhart, Rainer, & Wernicke, Axel. 2002. Localizing and segmenting text in images and videos. *IEEE Transactions on circuits and systems for video technology*, **12**(4), 256–268.

Lim, Tjen-Sien, Loh, Wei-Yin, & Shih, Yu-Shan. 2000. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine learning*, **40**(3), 203–228.

Lindig, Christian. 2000. Fast concept analysis. *Working with Conceptual Structures-Contributions to ICCS*, **2000**, 152–161.

Liquière, Michel, & Mephu Nguifo, E. 1990. LEGAL : LEarning with GAlois lattice. *Journées Franaises sur l'Apprentissage, Lannion*, 93–113.

Liu, Cheng-Lin, Yin, Fei, Wang, Da-Han, & Wang, Qiu-Feng. 2013. Online and offline handwritten Chinese character recognition : benchmarking on new databases. *Pattern Recognition*, **46**(1), 155–162.

Liu, Xiaoqing, & Samarabandu, Jagath. 2006. Multiscale edge-based text extraction from complex images. *Pages 1721–1724 of : Multimedia and Expo, 2006 IEEE International Conference on*. IEEE.

Lounkine, Eugen, Auer, Jens, & Bajorath, Jurgen. 2008. Formal concept analysis for the identification of molecular fragment combinations specific for active and highly potent compounds. *Journal of medicinal chemistry*, **51**(17), 5342–5348.

Luxenburger, Michael. 1991. Implications partielles dans un contexte. *Mathématiques, informatique et sciences humaines*, **29**(113), 35–55.

MacQueen, James, *et al.* 1967. Some methods for classification and analysis of multivariate observations. *Pages 281–297 of : Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1. Oakland, CA, USA.

Madden, M. 2003. The performance of Bayesian network classifiers constructed using different techniques. *Pages 59–70 of : Proceedings of European conference on machine learning, workshop on probabilistic graphical models for classification*.

Maddouri, Mondher. 2005. A formal concept analysis approach to discover association rules from data. *Pages 10–21 of : CLA*.

Maddouri, Mondher, & Kaabi, Fatma. 2006. On statistical measures for selecting pertinent formal concepts to discover production rules from data. *Pages 780–784 of : Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*. IEEE.

Malerba, Donato, Esposito, Floriana, Lisi, Francesca A., & Altamura, Oronzo. 2001. Automated discovery of dependencies between logical components in document image understanding. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, **2001-January**, 174–178.

Manolopoulos, Y., Papadopoulos, A. N., & Vassilakopoulos, M. 2005. *Spatial databases : technologies, techniques and trends*. Igi Global.

Mao, Jianchang, & Jain, Anil K. 1996. A self-organizing network for hyperellipsoidal clustering (HEC). *Ieee transactions on neural networks*, **7**(1), 16–29.

Mao, Song, Rosenfeld, Azriel, & Kanungo, Tapas. 2003. Document structure analysis algorithms : a literature survey. *Pages 197–207 of : Electronic Imaging 2003*. International Society for Optics and Photonics.

Margner, Volker, & El Abed, Haikal. 2011. ICDAR 2011-Arabic handwriting recognition competition. *Pages 1444–1448 of : Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE.

Masters, Timothy. 1995. *Advanced algorithms for neural networks : a C++ sourcebook.* John Wiley & Sons, Inc.

Masuda, Isao, Hagita, N, Akiyama, T, Takahashi, T, & Naito, S. 1985. Approach to smart document reader system. *Pages 550–557 of : Proc. CVPR*, vol. 85.

Matas, Jiri, & Kittler, Josef. 1995. Spatial and feature space clustering : Applications in image analysis. *Pages 162–173 of : Computer Analysis of Images and Patterns.* Springer.

Matas, Jiri, Chum, Ondrej, Urban, Martin, & Pajdla, Tomás. 2004. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, **22**(10), 761–767.

McCulloch, Warren S, & Pitts, Walter. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, **5**(4), 115–133.

Meddouri, Nida, & Maddouri, Mondher. 2009. Boosting Formal Concepts to Discover Classification Rules. *Pages 501–510 of : IEA/AIE.* Springer.

Medvet, Eric, Bartoli, Alberto, & Davanzo, Giorgio. 2011. A probabilistic approach to printed document understanding. *International journal on document analysis and recognition*, **14**(4), 335–347.

Messai, Nizar, Bouaud, Jacques, Aufaure, Marie-Aude, Zelek, Laurent, & Séroussi, Brigitte. 2011. Using formal concept analysis to discover patterns of non-compliance with clinical practice guidelines : a case study in the management of breast cancer. *Pages 119–128 of : Conference on Artificial Intelligence in Medicine in Europe.* Springer.

Milligan, Glenn W. 1981. A monte carlo study of thirty internal criterion measures for cluster analysis. *Psychometrika*, **46**(2), 187–199.

Milligan, Glenn W, & Cooper, Martha C. 1985. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, **50**(2), 159–179.

Minaee, Shervin, Yu, Haoping, & Wang, Yao. 2014. A Robust Regression Approach for Background/Foreground Segmentation. *arXiv preprint arXiv :1412.5126.*

Mitra, Sushmita, Pal, Sankar K, & Mitra, Pabitra. 2002. Data mining in soft computing framework : a survey. *IEEE transactions on neural networks*, **13**(1), 3–14.

Molloy, Ian, Chen, Hong, Li, Tiancheng, Wang, Qihua, Li, Ninghui, Bertino, Elisa, Calo, Seraphin, & Lobo, Jorge. 2008. Mining roles with semantic meanings. *Pages 21–30 of : Proceedings of the 13th ACM symposium on Access control models and technologies.* ACM.

Moon, Todd K. 1996. The expectation-maximization algorithm. *IEEE Signal processing magazine*, **13**(6), 47–60.

Motameny, Susanne, Versmold, Beatrix, & Schmutzler, Rita. 2008. Formal concept analysis for the identification of combinatorial biomarkers in breast cancer. *Formal Concept Analysis*, 229–240.

Mufti, Ghazi Bel, Bertrand, Patrice, & El Moubarki, Lassad. 2012. Decomposition of the Rand index in order to assess both the stability and the number of clusters of a partition.

Murthy, Sreerama K. 1998. Automatic construction of decision trees from data : A multi-disciplinary survey. *Data mining and knowledge discovery*, **2**(4), 345–389.

Nagy, George, Seth, Sharad, & Viswanathan, Mahesh. 1992. A prototype document image analysis system for technical journals. *Computer*, **25**(7), 10–22.

Naïm, Patrick, Wuillemin, Pierre-Henri, Leray, Philippe, Pourret, Olivier, & Becker, Anna. 2011. *Réseaux bayésiens*. Editions Eyrolles.

Narendra, Kumpati S, & Parthasarathy, Kannan. 1991. Gradient methods for the optimization of dynamics systems containing neural networks. *IEEE Transactions on Neural networks*, **2**(2), 252–262.

Nehmé, Kamal, Valtchev, Petko, Rouane, Mohamed H, & Godin, Robert. 2005. On computing the minimal generator family for concept lattices and icebergs. *Pages 192–207 of : International Conference on Formal Concept Analysis*. Springer.

Ng, Raymond T., & Han, Jiawei. 2002. CLARANS : A method for clustering objects for spatial data mining. *IEEE transactions on knowledge and data engineering*, **14**(5), 1003–1016.

Nguifo, Engelbert Mephu, & Njiwoua, Patrick. 2001. IGLUE : A lattice-based constructive induction system. *Intelligent data analysis*, **5**(1), 73–91.

Nguifo, Engelbert Mephu, Tsopzé, Norbert, & Tindo, Gilbert. 2008. M-CLANN : Multiclass concept lattice-based artificial neural network for supervised classification. *Pages 812–821 of : International Conference on Artificial Neural Networks*. Springer.

Niculescu, Radu Stefan, Mitchell, Tom M, & Rao, R Bharat. 2006. Bayesian network learning with parameter constraints. *Journal of Machine Learning Research*, **7**(Jul), 1357–1383.

Niyogi, Debashish, & Srihari, Sargur N. 1995. Knowledge-based derivation of document logical structure. *Pages 472–475 of : Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, vol. 1. IEEE.

Njiwoua, P, & Mephu Nguifo, E. 1999. Améliorer l'apprentissage à partir d'instances grâce à l'induction de concepts : le système CIBLe. *Revue d'intelligence artificielle*, **13**(2), 413–440.

O'Gorman, Lawrence. 1993. The document spectrum for page layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**(11), 1162–1173.

O'Gorman, Lawrence, & Kasturi, Rangachar. 1995. *Document image analysis*. Vol. 39. IEEE Computer Society Press Los Alamitos.

Oliver, Jonathan J, Baxter, Rohan A, & Wallace, Chris S. 1996. Unsupervised learning using MML. *Pages 364–372 of : ICML*.

Oosthuizen, GD. 1996. The application of concept lattice to machine learning. *Dept. Comput. Sci., Univ. Pretoria, Pretoria, South Africa, Tech. Rep. CSTR*, **94**(01).

Oria, V., Özsu, M. T., & Iglinski, P. 2001. Querying Images in the DISIMA DBMS. *Pages 89–98 of : Multimedia Information Systems.*

Outrata, Jan. 2010. Boolean factor analysis for data preprocessing in machine learning. *Pages 899–902 of : Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on.* IEEE.

Pakhira, Malay K, Bandyopadhyay, Sanghamitra, & Maulik, Ujjwal. 2004. Validity index for crisp and fuzzy clusters. *Pattern recognition*, **37**(3), 487–501.

Paredes, Roberto, Kavallieratou, Ergina, & Lins, Rafael Dueire. 2010. ICFHR 2010 contest : Quantitative evaluation of binarization algorithms. *Pages 733–736 of : Frontiers in Handwriting Recognition (ICFHR), 2010 International Conference on.* IEEE.

Pasquier, Nicolas, Bastide, Yves, Taouil, Rafik, & Lakhal, Lotfi. 1999. Closed sets based discovery of small covers for association rules. *Pages 361–381 of : BDA'1999 international conference on Advanced Databases.*

Patel, Chirag, Patel, Atul, & Patel, Dharmendra. 2012. Optical character recognition by open source OCR tool tesseract : A case study. *International Journal of Computer Applications*, **55**(10).

Pearl, Judea. 1982. *Reverend Bayes on inference engines : A distributed hierarchical approach.* Cognitive Systems Laboratory, School of Engineering and Applied Science, University of California, Los Angeles.

Pearl, Judea. 1984. Heuristics : intelligent search strategies for computer problem solving.

Pearl, Judea. 1986. Fusion, propagation, and structuring in belief networks. *Artificial intelligence*, **29**(3), 241–288.

Pearl, Judea. 1988. *Probabilistic inference in intelligent systems.*

Pearl, Judea, & Verma, Thomas S. 1995. A theory of inferred causation. *Studies in Logic and the Foundations of Mathematics*, **134**, 789–811.

PEHRO, Duda, & Stork, DG. 2001. Pattern classification. *D Wiley-Interscience Publication.*

Perea, Ignacio, & López, Damián. 2004. Syntactic modeling and recognition of document images. *Structural, Syntactic, and Statistical Pattern Recognition*, 416–424.

Phan, Trung Quy, Shivakumara, P, & Tan, CL. 2009. A Gradient Difference Based Technique for Video Text Detection. *Pages 156–160 of : Proceeding of 2009 International Conference on Document Analysis and Recognition (ICDAR).*

Phillips, Ihsin T, Chen, Su, & Haralick, Robert M. 1993. CD-ROM document database standard. *Pages 478–483 of : Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on.* IEEE.

Poelmans, Jonas, Elzinga, Paul, Viaene, Stijn, & Dedene, Guido. 2010. Formal concept analysis in knowledge discovery : a survey. *Conceptual Structures : From Information to Intelligence*, 139–153.

Poelmans, Jonas, Ignatov, Dmitry I, Viaene, Stijn, Dedene, Guido, Kuznetsov, Sergei O, *et al.* 2012. Text Mining Scientific Papers : A Survey on FCA-Based Information Retrieval Research. *Pages 273–287 of : ICDM*, vol. 7377. Springer.

Poelmans, Jonas, Ignatov, Dmitry I, Kuznetsov, Sergei O, & Dedene, Guido. 2013a. Formal concept analysis in knowledge processing : A survey on applications. *Expert systems with applications*, **40**(16), 6538–6560.

Poelmans, Jonas, Kuznetsov, Sergei O, Ignatov, Dmitry I, & Dedene, Guido. 2013b. Formal concept analysis in knowledge processing : A survey on models and techniques. *Expert systems with applications*, **40**(16), 6601–6623.

Pratikakis, Ioannis, Gatos, Basilis, & Ntirogiannis, Konstantinos. 2013. ICDAR 2013 document image binarization contest (DIBCO 2013). *Pages 1471–1476 of : Document Analysis and Recognition (ICDAR), 2013 12th International Conference on.* IEEE.

Priss, Uta. 1997. A graphical interface for document retrieval based on formal concept analysis. *Pages 66–70 of : Proceedings of the 8th Midwest Artificial Intelligence and Cognitive Science Conference.*

Priss, Uta. 2006. Formal concept analysis in information science. *Arist*, **40**(1), 521–543.

Quinlan, J Ross. 1993. C4. 5 : Programming for machine learning. *Morgan Kauffmann*, **38**.

Quinlan, J Ross et al. 1979. *Discovering rules by induction from large collections of examples.* Expert systems in the micro electronic age. Edinburgh University Press.

Rai, Pradeep, & Singh, Shubha. 2010. A survey of clustering techniques. *International Journal of Computer Applications*, **7**(12), 1–5.

Raj, Hrishav, & Ghosh, Rajib. 2014. Devanagari text extraction from natural scene images. *Pages 513–517 of : Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on.* IEEE.

Raju, S Sabari, Pati, Peeta Basa, & Ramakrishnan, AG. 2004. Gabor filter based block energy analysis for text extraction from digital document images. *Pages 233–243 of : Document Image Analysis for Libraries, 2004. Proceedings. First International Workshop on.* IEEE.

Ren, Chao, An, Ning, Wang, Jianzhou, Li, Lian, Hu, Bin, & Shang, Duo. 2014. Optimal parameters selection for BP neural network based on particle swarm optimization : A case study of wind speed forecasting. *Knowledge-Based Systems*, **56**, 226–239.

Richards, Debbie, & Malik, Usama. 2003. Multi-level knowledge discovery from rule bases. *Applied Artificial Intelligence*, **17**(3), 181–205.

Ricordeau, Marc. 2003. Q-concept-learning : generalization with concept lattice representation in reinforcement learning. *Pages 316–323 of : Tools with Artificial Intelligence, 2003. Proceedings. 15th IEEE International Conference on.* IEEE.

Ricordeau, Marc, & Liquiere, Michel. 2007. Policies Generalization in Reinforcement Learning using Galois Partitions Lattices. *In : CLA*.

Rissanen, Jorma. 1978. Modeling by shortest data description. *Automatica*, **14**(5), 465–471.

Rissanen, Jorma. 1998. *Stochastic complexity in statistical inquiry*. Vol. 15. World scientific.

Robnik-Šikonja, Marko, & Kononenko, Igor. 2003. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine learning*, **53**(1-2), 23–69.

Rock, Tammo, & Wille, Rudolf. 2000. Ein toscana-erkundungssystem zur literatursuche. *Pages 239–253 of : Begriffliche Wissensverarbeitung*. Springer.

Rokach, Lior, & Maimon, Oded. 2014. *Data mining with decision trees : theory and applications*. World scientific.

Rosenblatt, Frank. 1957. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory.

Rouane-Hacene, Mohamed, Toussaint, Yannick, & Valtchev, Petko. 2009. Mining safety signals in spontaneous reports database using concept analysis. *Pages 285–294 of : Conference on Artificial Intelligence in Medicine in Europe*. Springer.

Rousseeuw, Peter J. 1987. Silhouettes : a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, **20**, 53–65.

Rudolph, Sebastian. 2007. Using FCA for encoding closure operators into neural networks. *Conceptual Structures : Knowledge Architectures for Smart Applications*, 321–332.

Ruggieri, Salvatore. 2002. Efficient C4. 5 [classification algorithm]. *IEEE transactions on knowledge and data engineering*, **14**(2), 438–444.

Rumelhart, David E, Hinton, Geoffrey E, & Williams, Ronald J. 1985. *Learning internal representations by error propagation*. Tech. rept. DTIC Document.

Sahami, Mehran. 1995. Learning classification rules using lattices. *Machine learning : ECML-95*, 343–346.

Saitta, Lorenza, & Neri, Filippo. 1998. Learning in the "real world". *Machine learning*, **30**(2), 133–163.

Scholkopf, Bernhard, & Smola, Alexander J. 2001. *Learning with kernels : support vector machines, regularization, optimization, and beyond*. MIT press.

Schwarz, Gideon, *et al.* 1978. Estimating the dimension of a model. *The annals of statistics*, **6**(2), 461–464.

Scowen, Roger S. 1998. *Extended BNF-a generic base standard*. Tech. rept. Technical report, ISO/IEC 14977. http ://www. cl. cam. ac. uk/mgk25/iso-14977. pdf.

Seeri, Shivananda V, Pujari, JD, & Hiremath, PS. 2015. Multilingual text localization in natural scene images using wavelet based edge features and fuzzy classification. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, **4**(1), 210–218.

Seeri, Shivananda V, Pujari, JD, & Hiremath, PS. 2016. Text Localization and Character Extraction in Natural Scene Images using Contourlet Transform and SVM Classifier. *International Journal of Image, Graphics and Signal Processing*, **8**(5), 36.

Shekar, BH, Smitha, ML, & Shivakumara, P. 2014. Discrete wavelet transform and gradient difference based approach for text localization in videos. *Pages 280–284 of : Signal and Image Processing (ICSIP), 2014 Fifth International Conference on*. IEEE.

Sibson, Robin. 1973. SLINK : an optimally efficient algorithm for the single-link cluster method. *The computer journal*, **16**(1), 30–34.

Skorsky, Martin. 1997. Graphische Darstellung eines Thesaurus. *DGD-Reihe DOK*, 119–125.

Sobottka, Karin, Bunke, Horst, & Kronenberg, Heino. 1999. Identification of text on colored book and journal covers. *Pages 57–62 of : Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on*. IEEE.

Sokal, Robert R. 1958. A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, **38**, 1409–1438.

Soon, Keanhuat, & Kuhn, Werner. 2004. Formalizing user actions for ontologies. *Geographic Information Science*, 299–312.

Sørensen, Thorvald. 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biol. Skr.*, **5**, 1–34.

Specht, Donald F. 1991. A general regression neural network. *IEEE transactions on neural networks*, **2**(6), 568–576.

Stumme, Gerd, Taouil, Rafik, Bastide, Yves, Pasquier, Nicolas, & Lakhal, Lotfi. 2001. Intelligent structuring and reducing of association rules with formal concept analysis. *Pages 335–350 of : Annual Conference on Artificial Intelligence*. Springer.

Stumpfe, Dagmar, Lounkine, Eugen, & Bajorath, Jürgen. 2011. Molecular test systems for computational selectivity studies and systematic analysis of compound selectivity profiles. *Chemoinformatics and Computational Chemical Biology*, 503–515.

Sug, Hyontai. 2009. An effective sampling method for decision trees considering comprehensibility and accuracy. *W. Trans. on Comp*, **8**(4), 631–640.

Sumathi, CP, & Devi, G Gayathri. 2014. Automatic text extraction from complex colored images using gamma correction method. *Journal of Computer Science*, **10**(4), 705.

Sumathi, CP, Santhanam, T, & Devi, G Gayathri. 2012a. A survey on various approaches of text extraction in images. *International Journal of Computer Science and Engineering Survey*, **3**(4), 27.

Sumathi, CP, Santhanam, T, & Priya, N. 2012b. Techniques and challenges of automatic text extraction in complex images : a survey. *Journal of Theoretical and Applied Information Technology*, **35**(2), 225–235.

Summers, Kristen. 1995. Near-wordless document structure classification. *Pages 462–465 of : Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, vol. 1. IEEE.

Sung, Myung-Chul, Jun, Bongjin, Cho, Hojin, & Kim, Daijin. 2015. Scene text detection with robust character candidate extraction method. *Pages 426–430 of : Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE.

Tang, Yuan Y., Lee, Seong Whan, & Suen, Ching Y. 1996. Automatic document processing : A survey. *Pattern Recognition*, **29**(12), 1931–1952.

Tang, Yuan Yan, Ma, Hong, Mao, Xiaogang, Liu, Dan, & Suen, Ching Y. 1995. A new approach to document analysis based on modified fractal signature. *Pages 567–570 of : Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, vol. 2. IEEE.

Tateisi, Yuka, & Itoh, Nohuyasu. 1994. Using stochastic syntactic analysis for extracting a logical structure from a document image. *Pages 391–394 of : Pattern Recognition, 1994. Vol. 2-Conference B : Computer Vision & Image Processing., Proceedings of the 12th IAPR International. Conference on*, vol. 2. IEEE.

Tehsin, Samabia, Masood, Asif, & Kausar, Sumaira. 2014. Survey of Region-Based Text Extraction Techniques for Efficient Indexing of Image/Video Retrieval. *International Journal of Image, Graphics and Signal Processing*, **6**(12), 53.

Tekaya, Sondess Ben, Yahia, Sadok Ben, & Slimani, Yahya. 2005. GenAll Algorithm : Decorating Galois lattice with minimal generators. *Pages 166–178 of : CLA*.

Tibshirani, Robert. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.

Tibshirani, Robert, Walther, Guenther, & Hastie, Trevor. 2001. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, **63**(2), 411–423.

Tilley, Thomas. 2004. Tool support for FCA. *Pages 104–111 of : ICFCA*, vol. 2961. Springer.

Tilley, Thomas, & Eklund, Peter. 2007. Citation analysis using Formal Concept Analysis : A case study in software engineering. *Pages 545–550 of : Database and Expert Systems Applications, 2007. DEXA'07. 18th International Workshop on*. IEEE.

Trupin, Éric. 2005. 01-La reconnaissance d'images de documents : Un panorama.

Tsopzé, Norbert, Nguifo, Engelbert Mephu, & Tindo, Gilbert. 2007. CLANN : Concept Lattice-based Artificial Neural Network for Supervised Classification. *In : CLA*, vol. 331.

Tsujimoto, Suichi, & Asada, Haruo. 1990. Understanding multi-articled documents. *Pages 551–556 of : Pattern Recognition, 1990. Proceedings., 10th International Conference on*, vol. 1. IEEE.

Valtchev, Petko, Missaoui, Rokia, & Godin, Robert. 2004. Formal concept analysis for knowledge discovery and data mining : The new challenges. *Pages 352–371 of : ICFCA*, vol. 2961. Springer.

Valtchev, Petko, Missaoui, Rokia, & Godin, Robert. 2008. A framework for incremental generation of closed itemsets. *Discrete Applied Mathematics*, **156**(6), 924–949.

Valveny, Ernest, Terrades, Oriol Ramos, Mas, Joan, & Rossiñol, Marçal. 2013. *Interactive Document Retrieval and Classification.*

van De Vel, Marcel LJ. 1993. *Theory of convex structures.* Vol. 50. Elsevier.

Van Der Merwe, Dean, Obiedkov, Sergei, & Kourie, Derrick. 2004. Addintent : A new incremental algorithm for constructing concept lattices. *Concept Lattices*, 205–206.

Vendramin, Lucas, Campello, Ricardo JGB, & Hruschka, Eduardo R. 2010. Relative clustering validity criteria : A comparative overview. *Statistical analysis and data mining : the ASA data science journal*, **3**(4), 209–235.

Villerd, Jean, Toussaint, Yannick, & Lillo-Le Louët, Agnès. 2010. Adverse drug reaction mining in pharmacovigilance data using formal concept analysis. *Pages 386–401 of : Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer.

Visani, Muriel, Bertet, Karell, & Ogier, J-M. 2011. Navigala : An original symbol classifier based on navigation through a galois lattice. *International Journal of Pattern Recognition and Artificial Intelligence*, **25**(04), 449–473.

Voorhees, Ellen M. 1986. Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. *Information Processing & Management*, **22**(6), 465–476.

Wahl, Friedrich M, Wong, Kwan Y, & Casey, Richard G. 1982. Block segmentation and text extraction in mixed text/image documents. *Computer graphics and image processing*, **20**(4), 375–390.

Wang, Shiguo. 2010. A comprehensive survey of data mining-based accounting-fraud detection research. *Pages 50–53 of : Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, vol. 1. IEEE.

Ward Jr, Joe H. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, **58**(301), 236–244.

Weinberger, Kilian Q, Blitzer, John, & Saul, Lawrence K. 2006. Distance metric learning for large margin nearest neighbor classification. *Pages 1473–1480 of : Advances in neural information processing systems.*

Wermelinger, Michel, Yu, Yijun, & Strohmaier, Markus. 2009. Using formal concept analysis to construct and visualise hierarchies of socio-technical relations. *Pages 327–330 of : Software Engineering-Companion Volume, 2009. ICSE-Companion 2009. 31st International Conference on.* IEEE.

Wille, Rudolf. 1982. Restructuring lattice theory : an approach based on hierarchies of concepts. *Pages 445–470 of : Ordered sets*. Springer.

Witten, Ian H, Frank, Eibe, Hall, Mark A, & Pal, Christopher J. 2016. *Data Mining : Practical machine learning tools and techniques*. Morgan Kaufmann.

Wold, Svante, Esbensen, Kim, & Geladi, Paul. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems*, **2**(1-3), 37–52.

Wollbold, Johannes, Guthke, Reinhard, & Canter, Bernhard. 2008. Constructing a knowledge base for gene regulatory dynamics by formal concept analysis methods. *Lecture Notes in Computer Science*, **5147**, 230–244.

Wong, Kwan Y., Casey, Richard G., & Wahl, Friedrich M. 1982. Document analysis system. *IBM journal of research and development*, **26**(6), 647–656.

Wu, Jui-Chen, Hsieh, Jun-Wei, & Chen, Yung-Sheng. 2008. Morphology-based text line extraction. *Machine Vision and Applications*, **19**(3), 195–207.

Xie, Zhipeng, Hsu, Wynne, Liu, Zongtian, & Lee, Mong Li. 2002. Concept lattice based composite classifiers for high predictability. *Journal of Experimental & Theoretical Artificial Intelligence*, **14**(2-3), 143–156.

Xu, Rui, & Wunsch, Donald. 2005. Survey of clustering algorithms. *IEEE Transactions on neural networks*, **16**(3), 645–678.

Xu, Wei, Li, Wenjie, Wu, Mingli, Li, Wei, & Yuan, Chunfa. 2006. Deriving event relevance from the ontology constructed with formal concept analysis. *Computational Linguistics and Intelligent Text Processing*, 480–489.

Yamashita, A, Amano, T, Takahashi, I, & Toyokawa, K. 1991. A model based layout understanding method for the document recognition system. *Pages 130–138 of : Proceedings of International Conference on Document Analysis and Recognition*.

Ye, Qixiang, Jiao, Jianbin, Huang, Jun, & Yu, Hua. 2007. Text detection and restoration in natural scene images. *Journal of Visual Communication and Image Representation*, **18**(6), 504–513.

Ye, Xiangyun, Cheriet, Mohamed, & Suen, Ching Y. 2001. Stroke-model-based character extraction from gray-level document images. *IEEE Transactions on Image Processing*, **10**(8), 1152–1161.

Yıldız, Olcay Taner, & Dikmen, Onur. 2007. Parallel univariate decision trees. *Pattern Recognition Letters*, **28**(7), 825–832.

Yin, Xu-Cheng, Yin, Xuwang, Huang, Kaizhu, & Hao, Hong-Wei. 2014. Robust text detection in natural scene images. *IEEE transactions on pattern analysis and machine intelligence*, **36**(5), 970–983.

Yuan, Yuan, Kim, Il-Koo, Zheng, Xiaozhen, Liu, Lingzhi, Cao, Xiaoran, Lee, Sunil, Cheon, Min-Su, Lee, Tammy, He, Yun, & Park, Jeong-Hoon. 2012. Quadtree based nonsquare block structure for inter frame coding in high efficiency video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, **22**(12), 1707–1719.

Zaki, Mohammed Javeed, & Ogihara, Mitsunori. 1998. Theoretical foundations of association rules. *Pages 71–78 of : 3rd ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery.*

Zhang, Bin. 2001. Generalized k-harmonic means–dynamic weighting of data in unsupervised learning. *Pages 1–13 of : Proceedings of the 2001 SIAM International Conference on Data Mining.* SIAM.

Zhang, Honggang, Zhao, Kaili, Song, Yi-Zhe, & Guo, Jun. 2013. Text extraction from natural scene image : A survey. *Neurocomputing*, **122**, 310–323.

Zhang, Jing, & Kasturi, Rangachar. 2008. Extraction of text objects in video documents : Recent progress. *Pages 5–17 of : Document Analysis Systems, 2008. DAS'08. The Eighth IAPR International Workshop on.* IEEE.

Zheng, Zijian. 1998. Constructing conjunctions using systematic search on decision trees. *Knowledge-Based Systems*, **10**(7), 421–430.

Zheng, Zijian. 2000. Constructing X-of-N attributes for decision tree learning. *Machine learning*, **40**(1), 35–75.

Zhou, Baoyao, Hui, Siu Cheung, & Chang, Kuiyu. 2004. A formal concept analysis approach for web usage mining. *Pages 437–441 of : International Conference on Intelligent Information Processing.* Springer.

Zhou, Yun, Fenton, Norman, & Neil, Martin. 2014. Bayesian network approach to multinomial parameter learning using data and expert judgments. *International Journal of Approximate Reasoning*, **55**(5), 1252–1268.

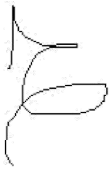
POLE RECHERCHE
Ecoles **D**octorales

LETTRE D'ENGAGEMENT DE NON-PLAGIAT

Je, soussigné(e) Cynthia Pitou ,

en ma qualité de doctorant(e) de l'Université de La Réunion, déclare être conscient(e) que le plagiat est un acte délictueux passible de sanctions disciplinaires. Aussi, dans le respect de la propriété intellectuelle et du droit d'auteur, je m'engage à systématiquement citer mes sources, quelle qu'en soit la forme (textes, images, audiovisuel, internet), dans le cadre de la rédaction de ma thèse et de toute autre production scientifique, sachant que l'établissement est susceptible de soumettre le texte de ma thèse à un logiciel anti-plagiat.

Fait à Sainte Clotilde le : 22/08/2017

Signature : 

Extrait du Règlement intérieur de l'Université de La Réunion
(validé par le Conseil d'Administration en date du 11 décembre 2014)

Article 9. Protection de la propriété intellectuelle – Faux et usage de faux, contrefaçon, plagiat

L'utilisation des ressources informatiques de l'Université implique le respect de ses droits de propriété intellectuelle ainsi que ceux de ses partenaires et plus généralement, de tous tiers titulaires de tels droits.
En conséquence, chaque utilisateur doit :
- utiliser les logiciels dans les conditions de licences souscrites ;
- ne pas reproduire, copier, diffuser, modifier ou utiliser des logiciels, bases de données, pages Web, textes, images, photographies ou autres créations protégées par le droit d'auteur ou un droit privatif, sans avoir obtenu préalablement l'autorisation des titulaires de ces droits.

La contrefaçon et le faux
Conformément aux dispositions du code de la propriété intellectuelle, toute représentation ou reproduction intégrale ou partielle d'une œuvre de l'esprit faite sans le consentement de son auteur est illicite et constitue un délit pénal.
L'article 444-1 du code pénal dispose : « Constitue un faux toute altération frauduleuse de la vérité, de nature à causer un préjudice et accomplie par quelque moyen que ce soit, dans un écrit ou tout autre support d'expression de la pensée qui a pour objet ou qui peut avoir pour effet d'établir la preuve d'un droit ou d'un fait ayant des conséquences juridiques ».
L'article L335_3 du code de la propriété intellectuelle précise que : « Est également un délit de contrefaçon toute reproduction, représentation ou diffusion, par quelque moyen que ce soit, d'une œuvre de l'esprit en violation des droits de l'auteur, tels qu'ils sont définis et réglementés par la loi. Est également un délit de contrefaçon la violation de l'un des droits de l'auteur d'un logiciel (…) ».

Le plagiat est constitué par la copie, totale ou partielle d'un travail réalisé par autrui, lorsque la source empruntée n'est pas citée, quel que soit le moyen utilisé. Le plagiat constitue une violation du droit d'auteur (au sens des articles L 335-2 et L 335-3 du code de la propriété intellectuelle). Il peut être assimilé à un délit de contrefaçon. C'est aussi une faute disciplinaire, susceptible d'entraîner une sanction.
Les sources et les références utilisées dans le cadre des travaux (préparations, devoirs, mémoires, thèses, rapports de stage…) doivent être clairement citées. Des citations intégrales peuvent figurer dans les documents rendus, si elles sont assorties de leur référence (nom d'auteur, publication, date, éditeur…) et identifiées comme telles par des guillemets ou des italiques.

Les délits de contrefaçon, de plagiat et d'usage de faux peuvent donner lieu à une sanction disciplinaire indépendante de la mise en œuvre de poursuites pénales.