Département d'Informatique de l'Université de Fribourg
(Suisse)

Une étude de l'évolutivité des modèles pour la reconnaissance de documents arabes dans un contexte interactif.

Thèse de doctorat

soumise à la Faculté des Sciences de l'Université de Fribourg (Suisse) pour l'obtention du grade de Doctor Scientiarium Informaticarum

Karim HADJAR

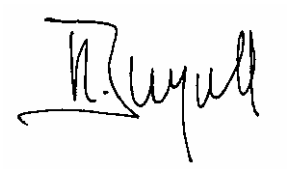
Thèse n°1513
Imprimerie St Paul, Fribourg
2006

Acceptée par la Faculté des Sciences de l'Université de Fribourg, sur la proposition des Professeurs :

- Rolf INGOLD, Université de Fribourg, Suisse ;
- Eric Trupin, Université de Rouen, France ;
- Beat HIRSBRUNNER, Université de Fribourg, Suisse.

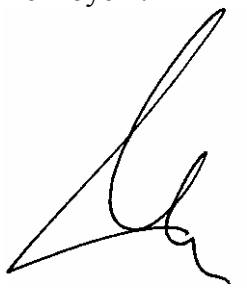
Fribourg, 13 avril 2006

Le Directeur de thèse :



Prof. Rolf INGOLD

Le Doyen :



Prof. Marco Celio

Remerciements

Arrivé au terme de ce doctorat, je tiens à exprimer mes remerciements envers toutes les personnes qui m'ont aidé de près ou de loin durant ces cinq ans d'études. Certaines contributions méritent d'être relevées tout spécialement :

- Rolf Ingold m'a offert un encadrement idéal, et m'a dirigé de manière fort judicieuse. Il a parfaitement su m'initier au monde scientifique, en me transmettant à la fois une formation des plus solides dans son domaine, une rigueur dans la démarche du chercheur, et le plaisir de s'ouvrir à d'autres disciplines.

- Eric Trupin m'a toujours impressionné par la fraîcheur de ses idées. Ses remarques détaillées et pertinentes sur la première version de ma thèse m'ont aidé à l'améliorer considérablement. Je lui suis très reconnaissant d'avoir participé à ma défense en tant qu'expert. Ce fut un honneur pour nous de recevoir l'appui d'un spécialiste si prestigieux en reconnaissance de documents.

- Béat Hirsbrunner a depuis longtemps manifesté de l'intérêt pour mes travaux jusqu'à accepter de joindre le jury de thèse. Je lui suis très reconnaissant d'avoir participé à ma défense en tant qu'expert. Venant d'une personnalité si compétente et sympathique, ses commentaires m'ont beaucoup touché.

- Asmaa El Hannani, m'a considérablement aidé dans la correction des chapitres de ma thèse. Ses commentaires m'ont beaucoup aidé.

- Maurizio Rigamonti m'a apporté une assistance considérable dans l'adaptation de *xmillum*.

- Mes collègues m'ont offert une ambiance de travail inoubliable. Merci donc à Oliver Hitz, Lyse Robadey, Jean-Luc Bloechle, Denis Lalanne, et toute l'équipe DIVA.

- Je remercie encore tous les membres de ma famille plus particulièrement mon père, décédé récemment, pour ses encouragements et pour tous les efforts consentis, ma mère pour son dévouement et ma femme qui n'a jamais douté de mes capacités. Son tendre soutien m'accompagnait en permanence.

A la mémoire de mon père,
A ma mère,
A ma femme,
A mon fils,
A ma famille.

Table des matières

Abstract

This thesis addresses the recognition of physical and logical structures of complex documents, rich in variability. More precisely, we studied the evolution of models within an interactive context where the system gradually integrates the knowledge induced by the corrections of the user.

We studied the features of the Arabic language and we designed a recognition system for this language. In a first stage, we adapted traditional segmentation methods that are generally used for documents using a Latin alphabet. We noted that the results obtained by these methods, can be improved by integrating knowledge related to the treated class of documents. For that purpose we recommend the intervention of a user. The idea is to transfer the expertise from the user towards the recognition system by converting its corrections into knowledge. Thus, in the second stage, we built two systems for performing respectively the physical recognition (*PLANET*) and logic (*LUNET*) by using an evolutiv model which adapts to all new class of documents.

The *PLANET* system uses several dedicated models; each one being associated a given class of documents. The task of these models is to learn the specific features of their class. The dedicated models are initialized with a general model, which is built in order to integrate general knowledge of a superclass of documents.

The *PLANET* and *LUNET* systems have been evaluated on the classes of documents which are well adapted to the problematic: three classes of newspapers in Arabic language (ANNAHAR, AL HAYAT et AL QUDS). After the interactive treatment of 10-15 pages, the recognition rate raised from 96.729% to 98.687% which corresponds to a reduction in the error rate of 59.859%. As for *LUNET*, the average recognition rate is 94% with a reduction in the error rate of 63.436%.

Thus, we estimate having shown the relevance of using evolutiv models for the recognition of the physical and logical structures, of complex documents. This type of approach is particularly advantageous for mid-sized applications; it is for instance the case of ground truth production, which is a tiresome and expensive operation. Thanks to *PLANET/LUNET* the process of building such ground truth is simplified.

Keywords

Document image analysis – Physical layout analysis – Logical structure analysis – Classes of documents – Documents with rich structures and variability – Arabic language – Artificial neural networks – Evolutiv models – Ground truth.

Résumé

Cette thèse aborde la reconnaissance de structures physiques et logiques de documents complexes, riches en variabilité. Plus particulièrement, nous avons étudié l'évolutivité des modèles dans un contexte interactif, où le système intègre progressivement les connaissances induites par les corrections de l'utilisateur.

Nous avons étudié les caractéristiques de la langue arabe et nous avons conçu un système de reconnaissance pour cette langue. Dans un premier temps, nous avons adapté des méthodes de segmentation classiques, généralement utilisées pour les documents utilisant un alphabet latin. Nous avons constaté que les résultats obtenus par ces méthodes, peuvent être améliorés en intégrant des connaissances relatives à la classe de documents traitée. Nous préconisons pour cela l'intervention de l'utilisateur. L'idée est de transférer l'expertise de l'utilisateur vers le système de reconnaissance en convertissant ses corrections en connaissances. Ainsi, dans un deuxième temps, nous avons construit deux systèmes de reconnaissance pour traiter respectivement la reconnaissance physique (*PLANET*) et logique (*LUNET*) en utilisant un modèle évolutif qui s'adapte à toute nouvelle classe de documents.

Le système *PLANET* utilise plusieurs modèles dédiés, chacun étant associé à une classe de documents donnée. La tâche de ces modèles est d'apprendre les caractéristiques propres à leur classe. Les modèles dédiés sont initialisés avec un modèle général qui est construit en vue d'avoir une connaissance générale de la superclasse de documents.

Les systèmes *PLANET* et *LUNET* ont été évalués sur les classes de documents bien adaptés à la problématique : les journaux en langue arabe (ANNAHAR, AL HAYAT et AL QUDS). Après le traitement interactif de 10-15 pages de documents, le taux de reconnaissance passe de 96.729% à 98.687% ce qui correspond à une diminution du taux d'erreurs de 59.859%. Quant à *LUNET*, le taux moyen de reconnaissance est de 94% avec une diminution du taux d'erreurs de 63.436%.

Ainsi, nous estimons avoir démontré la pertinence d'utiliser des modèles évolutifs pour la reconnaissance de structures physiques et logiques de documents complexes. Ce type d'approche est particulièrement avantageux pour les applications de reconnaissance de taille moyenne ; c'est notamment le cas de la création de fonds de vérité qui est une opération fastidieuse et coûteuse. Grâce à *PLANET / LUNET* le processus de construction de tels fonds est simplifié.

Mots-clés

Analyse d'images de documents – Reconnaissance structures physiques – Reconnaissance structures logiques – Classes de documents – Documents riches en structures et en variabilité – Langue arabe – Réseaux de neurones artificiels – Modèles évolutifs – Fonds de vérité.

Chapitre 1

Introduction

Depuis l'avènement de l'écriture, aux environs du III ème millénaire avant Jésus Christ (JC) en Mésopotamie, l'être humain n'a cessé d'améliorer ce moyen de communication. En effet, plusieurs civilisations ont apporté leur savoir faire dans le domaine de l'écriture. Il y a eu d'abord l'écriture des hiéroglyphes des pharaons, l'écriture chinoise, l'écriture grecque, et puis l'écriture arabe et romaine. Certaines de ces écritures ont disparu avec la destruction de leur civilisation alors que d'autres sont encore d'actualité. Malheureusement parmi les 3000 langues dénombrées dans le monde, seulement une centaine sont dotées d'un système d'écriture. Mais la percée majeure de l'écriture a atteint son apogée au 15ème siècle après JC avec l'invention de l'imprimerie par Gutenberg. Et c'est à partir de cette époque que l'écriture est entrée dans une nouvelle ère, à savoir l'ère du document imprimé.

La notion de document est très générale et il existe une panoplie de définitions. Une définition appropriée d'un document est la suivante : un document est le support physique pour conserver et transmettre de l'information. Selon le support choisi, un document peut être textuel, graphique, multimédia (sonore, vidéo). Le document imprimé a eu un essor en deux temps grâce à l'introduction de l'imprimerie et de l'informatique. En effet, de nos jours, avec l'apport de l'informatique, un grand nombre de documents actuels, qui existent de par le monde, est en format numérique. Ces documents sont soit conçus et réalisés, dés le départ, par ordinateur, soit numérisés au moyen d'un scanner. Il est à noter que le style des documents s'est enrichi de nouvelles typographies et de nouveaux designs grâce aux nouveaux logiciels permettant leur création. Ceci a permis de rendre les documents beaucoup plus attractifs et plus ergonomiques.

L'être humain arrive avec aisance à reconnaître n'importe quel document. Cette faculté est assez basique pour un être humain ; en revanche elle pose jusqu'à ce jour encore des problèmes pour l'ordinateur. L'analyse et la reconnaissance d'images de documents englobent un ensemble de techniques informatiques avec comme but la reconstitution du contenu du document sous la forme de documents structurés, selon une forme définie par l'application en question. Les documents structurés couvrent deux catégories de documents : les documents imprimés et les documents manuscrits. Parmi les documents imprimés nous distinguons les documents à structures simples et les documents à structures complexes. Dans cette thèse, nous nous intéressons aux documents à structures complexes.

La reconnaissance de documents s'applique à plusieurs langues écrites. La langue latine a reçu la plus grande attention de la part de chercheurs. En revanche, malgré le nombre de personnes qui parlent la langue arabe, peu de travaux de recherche sur la reconnaissance de documents ont été consacrés à cette langue.

Cette thèse à pour sujet l'évolutivité des modèles dans un contexte interactif pour la reconnaissance de structures physiques et logiques de documents riches en structures et en variabilité.

Dans ce chapitre nous présentons tout d'abord le document électronique. Ensuite, nous définissons la reconnaissance de documents. Puis nous décrivons les systèmes doués d'apprentissage et nous définissions les réseaux de neurones artificiels. Après, nous décrivons les caractéristiques de la langue arabe et les objectifs de cette thèse. Finalement, nous présentons l'organisation en chapitres.

1.1 Le document électronique

Dans cette section nous parlons du document électronique, des documents à structures complexes et de la production de documents avec les moyens électroniques.

1.1.1 Définition document électronique

Un document électronique est la représentation d'un document, sous la forme d'une structure de données stockée en mémoire ou sur un support informatique, transmissible entre ordinateurs. Dans un système informatique, cette structure de données est représentée dans un fichier sous forme d'une séquence d'octets. Un document électronique peut avoir plusieurs représentations, d'où la notion de format de fichiers.

La principale caractéristique d'un document électronique est sa facilité de modification. Tout document électronique peut être copié dans un système informatique, d'un support à un autre, sous forme d'un fichier. Tout document électronique est modifiable ; différentes opérations y sont applicables ; parmi celles-ci, nous citons l'impression, et l'édition. En ce qui concerne l'impression, elle consiste en la matérialisation du document électronique sous forme de papier. Par contre, les opérations d'édition (ajout, modification et suppression) modifient le contenu du document électronique. La suppression physique détruit de façon permanente le document électronique.

Une image synthétique peut être générée à partir d'un document électronique structuré. L'image numérisée est obtenue en numérisant le document papier. L'image synthétique et l'image numérisée représentent toutes les deux un aperçu du document électronique. Certes, elles sont toutes les deux des images électroniques ; cependant l'image numérisée est déformée et comporte du bruit alors que l'image synthétique en est dépourvue.

1.1.2 Documents à structures complexes

Les documents électroniques diffèrent entre eux du point de vue du contenu et du point de vue organisationnel. En effet, trois structures sont possibles : structures linéaires, structures hiérarchiques simples et structures complexes. Les premières sont représentées par exemple par les œuvres littéraires tels que les romans. Les deuxièmes sont représentées par les articles scientifiques ou les livres. Elles possèdent une organisation en chapitres, sections, articles et paragraphes que l'on peut représenter sous forme d'arbres. Les troisièmes sont représentées par les journaux, les magazines et les dépliants publicitaires. Elles possèdent une typographie riche et ne sont pas composées uniquement de texte mais d'une combinaison, selon une disposition variable, de textes, de graphiques et d'images.

De ce fait, nous pouvons définir les documents à structures complexes comme étant des documents possédant une structure de pages assez complexe et variable. Ce type de documents n'est pas régi par des règles claires et définies.

Les documents à structures complexes à l'instar des journaux sont construits de la manière suivante : l'éditeur en chef du journal et ses collaborateurs se réunissent ensemble pour décider de l'ensemble d'articles à insérer dans le journal. Une fois arrivé à consensus, le maquettiste et le graphiste interviennent pour le mettre en forme. Cette mise en forme est effectuée en utilisant un logiciel de PAO. Il incombe au maquettiste et au graphique de se soumettre aux exigences aussi bien du logiciel que de la structure de pages du journal. La variabilité de l'information à insérer dans le journal et les exigences du logiciel génèrent une variabilité entre l'édition du jour et celle du lendemain, connue sous la notion de variabilité intra-classes. Chaque éditeur de journal essaie de se distinguer de ses concurrents en donnant une empreinte à son journal par le biais d'une typographie et d’une représentation spécifique qui engendre une variabilité entre plusieurs éditeurs de journaux connue sous la notion de variabilité inter-classes.

La figure 1.1 illustre un exemple de documents à structures complexes et la variabilité de la structure physique intra-classes.

Figure 1.1 : Exemple de deux documents à structures complexes d'un même journal.

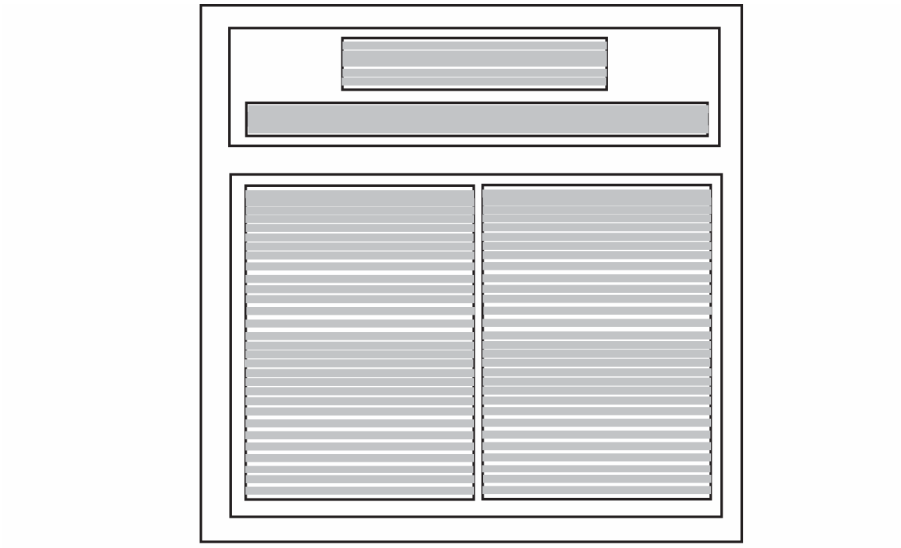
Afin de faciliter l'identification des différents documents à structures complexes, nous utilisons le principe des classes. En effet, nous regroupons dans une classe un ensemble de documents ayant des caractéristiques similaires. Par exemple, la classe "journaux" regroupe tous les journaux de différents éditeurs, la classe "magazines" regroupe tous les magazines, idem pour la classe "dépliants".

1.1.3 Production de documents avec les moyens électroniques

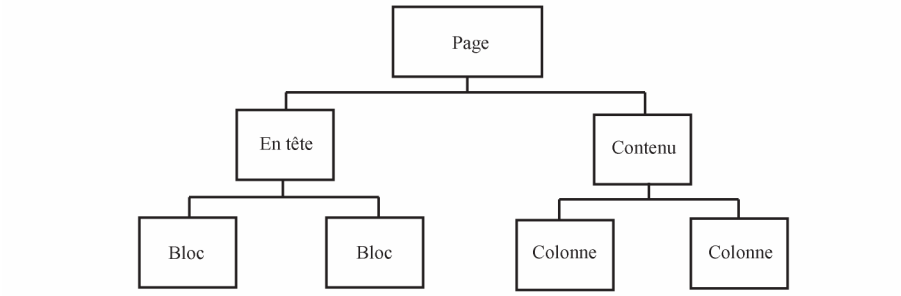
Dans l'édition professionnelle, le processus d'élaboration d'un document, depuis sa conception jusqu'à son impression, comporte plusieurs étapes faisant intervenir un certain nombre de personnes : l'auteur, le correcteur, le maquettiste et le typographe. Les systèmes de production de documents structurés retiennent essentiellement deux acteurs : l'auteur et le typographe.
Un document électronique commence par être saisi soit avec un logiciel de traitement de texte, soit avec un logiciel de publication assistée par ordinateur (PAO). Une fois cette étape achevée, le document est formaté. L'étape suivante consiste en la restitution du document au moyen d'un format de fichier permettant d'assurer une bonne qualité d'impression. La dernière étape se résume à l'impression du document électronique sous forme papier.

Parallèlement, le document électronique passe par différentes formes : structure logique, structure physique, image et papier. La structure logique reflète le point de vue de l'auteur; elle permet de représenter l'organisation du document en entités telles que chapitres, sections, paragraphes, etc… Il est à noter que le niveau de structuration utilisé est fonction de l'application visée. La structure physique permet de représenter la structuration du document en vue de son impression ; de ce fait certains critères, comme

par exemple la découpe en page et la répartition des espaces, sont nécessaires. La figure 1.2 illustre un exemple de structure physique et de structure logique.



Structure Physique



Structure Logique

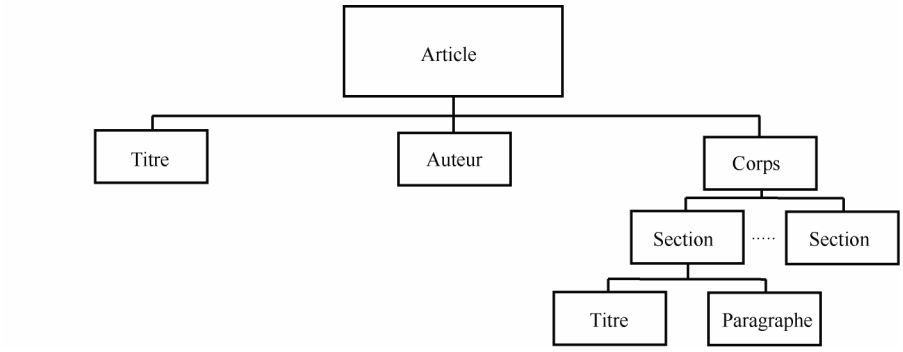


Figure 1.2 : Exemple de structures

L'image est obtenue par le biais d'une conversion de la structure physique. Finalement la représentation sur papier résulte de l'impression de l'image du document.

La figure 1.3 illustre les étapes de production et les différentes formes intermédiaires d'un document.

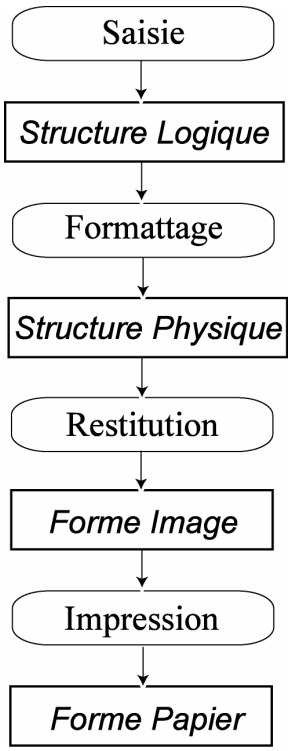


Figure 1.3 : Etapes de production et les différentes formes d'un document

Un document électronique peut être représenté au moyen de plusieurs formats de fichiers. Il existe plusieurs types de formats de fichiers : les formats structurés éditables, les formats d'échange et d'impression, et les images.

Les formats de fichiers structurés éditables se subdivisent en deux catégories : ceux avec balises et ceux WYSIWYG (What You See Is What You Get). Parmi les formats de fichiers structurés avec balises nous trouvons LATEX et HTML. Le premier est très répandu dans la communauté scientifique pour la rédaction des articles et des thèses. Le second est le langage de publication de pages web sur Internet. Pour les formats de fichier WYSIWYG, ils sont générés par des logiciels qui offrent une interface d'édition permettant de visualiser immédiatement le résultat final obtenu. Parmi ces logiciels nous citons Microsoft Word, Quark Xpress, Adobe Indesign, Adobe Framemaker et Adobe Pagemaker.

Les formats d'échange et d'impression quant à eux sont des formats non éditables. Parmi ces formats nous trouvons PS, EPS et PDF [1,2]. Ils ont été créés pour permettre la visualisation et l'impression sur toutes les plateformes d'une manière identique à l'original. Les fichiers en format PS et EPS sont, en réalité, des fichiers texte ASCII et ils représentent un programme de description de page. Cependant avec l'essor d'Internet,

PDF est devenu le format de diffusion de documents électroniques sur la toile car il est plus compact ; mais aussi le format le plus prisé pour les imprimeurs. Les fichiers en format PDF sont des fichiers binaires.

Les images permettent aussi de représenter un document électronique. Nous distinguons les images non compressées et les images compressées. Les premières sont très gourmandes en espace de stockage. Parmi ces formats, nous citons : BMP, PSD (format propriétaire d'Adobe) et TIFF [3] non compressé. Les deuxièmes ont été introduites dans le but d'accélérer leurs échanges sur la toile. En effet, les images compressées prennent moins de place en espace de stockage que celles non compressées et par conséquent leur transfert se trouve accélérer. Les formats d'images compressées comprennent ceux avec perte d'information et ceux sans perte d'information. Parmi les formats d'images compressés avec perte d'information nous citons le JPEG. Ce dernier est très utilisé dans le web mais aussi comme format de stockage dans les appareils photos numériques. En revanche il existe une multitude de formats d'images compressées sans perte d'information, tels que : GIF, PNG, et TIFF compressé.

Il arrive que certaines règles de structuration régissent un ensemble de documents. Ces règles sont communément appelées structures génériques. La structure générique définit le mode de construction des structures spécifiques. Chaque élément de la structure spécifique appartient à une classe générique. Les classes génériques sont définies par un ensemble de règles grammaticales.
Par exemple, la structure générique d'un mèl exprime certaines règles. Parmi celles-ci nous citons celle relative à l'ordre des éléments : un mél comprend l'adresse du destinataire, un sujet, un contenu, une signature et des pièces jointes. Le contenu est composé d'un ensemble de paragraphes suivi optionnellement d'une signature. Les pièces jointes sont facultatives.

1.2 Reconnaissance de documents

Dans cette section nous parlons des étapes de la reconnaissance, des structures de documents, des applications de la reconnaissance d'images de documents et de la reconnaissance de documents à structures complexes.

1.2.1 Etapes de la reconnaissance

Une fois un document électronique imprimé sous forme papier, l'auteur est confronté à l'archivage de cette forme électronique. Dans certains cas, l'archivage n'est pas bien assuré et, dans d'autres cas, les fichiers de cette forme électronique peuvent être détériorés ou manquants. Afin de remédier au problème de la détérioration de la forme électronique, les chercheurs ont recours à la reconnaissance. Celle-ci consiste à reconstituer le document électronique initial à partir de la forme papier. Cette reconstitution revient en

réalité à remonter les étapes de la production de documents de l'impression jusqu'à la saisie.

La numérisation ou l'acquisition d'images est le résultat de la conversion du document papier en une image numérisée. Ce processus est effectué soit par le biais d'un scanner soit d'une caméra. Le résultat de cette numérisation est une image. La qualité de l'image numérique obtenue dépend de plusieurs facteurs : la qualité du papier, la qualité du scanner ou de la caméra et le format d'image numérisée (compressé ou pas). En effet, s'il s'agit d'un document très ancien, le papier a de fortes chances d'avoir une couleur d'aspect jaunâtre. Ceci se reflète sur le résultat de la numérisation. Les trois points suivants diminuent la qualité : un scanner contenant de la poussière sur sa vitre, un scanner possédant une basse résolution et une caméra dont la mise au point est mal effectuée. Dans la chaîne de la qualité nous notons aussi le format de l'image numérisée, un format d'image compressé avec perte à l'instar de JPEG dégrade l'image numérisée obtenue.

Afin d'améliorer la qualité de l'image obtenue du document, plusieurs techniques sont mises en œuvre, les techniques de traitement d'image. Ces techniques permettent de rehausser la qualité de l'image du document et aussi de préparer le terrain pour les processus suivants. Les premières techniques que nous appliquons à l'image numérisée sont appelées prétraitement et consistent en un ensemble d'algorithmes (filtrage, redressement, lissage, squelettisation, binarisation) [27] dont l'objectif est de préparer le terrain à la reconnaissance. Le résultat de ce prétraitement est une image épurée dépourvue de bruit. Il est à noter que les images synthétiques sont aussi utilisées pour la reconnaissance et elles ne nécessitent ni une numérisation, ni un prétraitement contrairement aux images numérisées.

La reconnaissance de documents aura comme entrée une image numérisée épurée ou une image synthétique et elle est composée de deux étapes successives, l'une pour la reconnaissance de structures physiques (ou segmentation) et l'autre pour la reconnaissance de structures logiques.

La figure 1.4 illustre la reconnaissance de structures physiques et logiques.

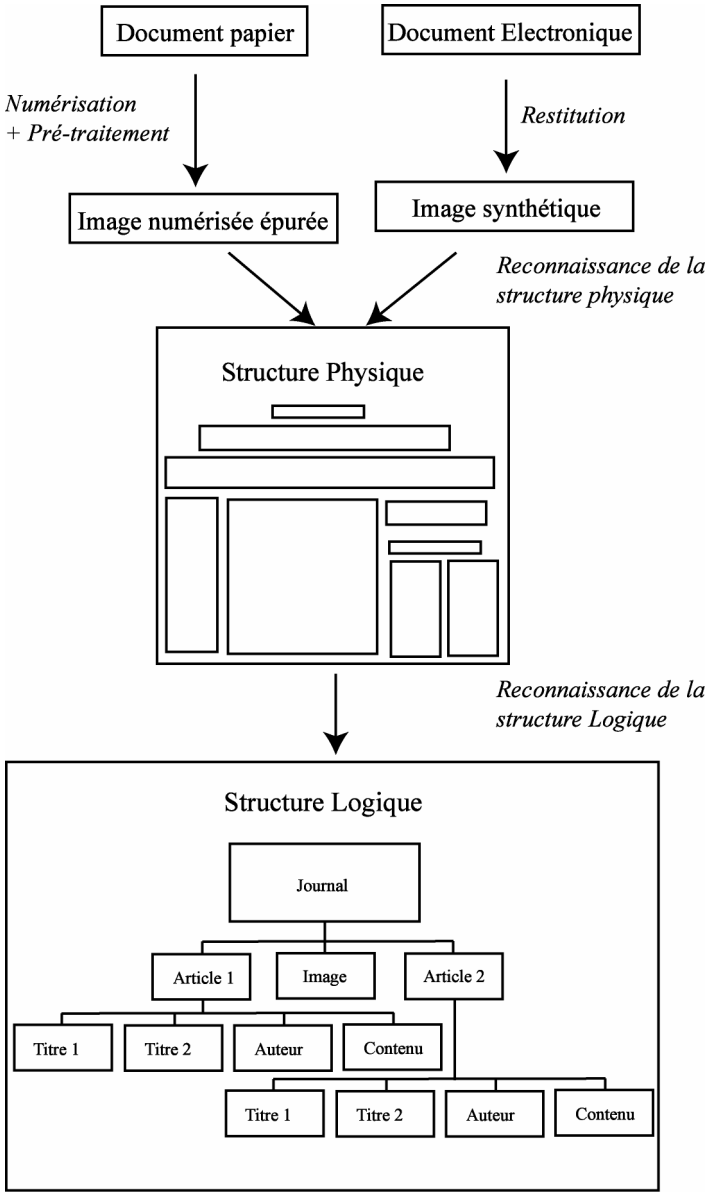


Figure 1.4 : Reconnaissance de structures physiques et logiques

La reconnaissance de structures physiques comprend la détection et la classification des différentes zones de l'image. Elle a pour objectif de délimiter toutes les régions d'intérêt de l'image. L'objectif de cette délimitation est de regrouper les régions en des zones homogènes : textes, graphiques et images. La finesse de cette décomposition de régions pour le cas du texte permet d'obtenir les lignes de texte, les mots et éventuellement les caractères.

Pour le cas d'un document composé d'une simple colonne, la reconnaissance de structures physiques comprend la segmentation en mots, en lignes de texte, et la fusion des lignes pour former des blocs. Si par contre le document est multi colonnes, une étape supplémentaire est nécessaire à savoir la segmentation du texte en colonnes et aussi la

détection des filets. De ce fait, la finesse de cette décomposition dépend du type de document mais aussi de l'application visée.

La reconnaissance de caractères a eu un grand essor de la part des chercheurs dès les années 1970 ; un grand nombre de travaux ont été effectués et le taux de reconnaissance s'est progressivement amélioré. La reconnaissance de caractères renferme aussi bien la reconnaissance de caractères imprimés que manuscrits. Si la reconnaissance de caractères imprimés est plus ou moins maîtrisée et donne de bons résultats, en revanche la reconnaissance des caractères manuscrits s'avère difficile et demeure un plein axe de recherche.

La dernière étape de la reconnaissance est la reconnaissance des structures logiques. Son objectif est de déterminer l'organisation logique des entités retrouvées au niveau de la reconnaissance de structures physiques, en effectuant un étiquetage. Les étiquettes utilisées dans cette étape sont dépendantes de l'application visée et peuvent correspondre à titre, auteur, paragraphe et article. La reconnaissance de structures logiques comprend aussi le recouvrement de l'ordre de lecture. Pour un document composé d'une seule colonne, cet ordre est de haut en bas et de gauche à droite. Cependant, cet ordre est fortement dépendant de la langue dans laquelle le document est écrit.

1.2.2 Structures de documents

Les documents électroniques sont sauvegardés dans des formats de fichiers différents. Chaque format de fichier permet de structurer le contenu du document selon des règles définies par l'éditeur du logiciel. Devant la richesse de formats de fichiers permettant la structuration de documents, nous nous trouvons devant un problème de conversion de l'information.

Plusieurs méthodes peuvent servir à minimiser les problèmes de conversion entre formats de documents. Parmi ces méthodes, nous citons la standardisation et les filtres. La première permet de faire de telle sorte que le format approuvé par l'organisme de standardisation devient un standard. Malheureusement certains éditeurs de logiciels n'adhèrent pas à ce format standardisé. La seconde permet de concevoir des filtres de conversion entre plusieurs formats de documents ; à ce sujet nous aurons des filtres composés de couples (document source, document destination). Néanmoins ces filtres sont conçus après l'apparition des nouvelles mises à jour des formats de documents.

Une solution à ces problèmes de conversion est le recours à l'image. Celle-ci est considérée comme étant le format pivot [15]. En effet, à partir de n’importe quel format de document et de n’importe quelle version de ce format on est capable de générer l’image du document. L'image est donc une représentation universelle de tous les formats de documents existants.

La reconnaissance utilise l’image en entrée et retourne comme résultat un fichier dans un format donné. L’image est très importante parce que le résultat de la reconnaissance

dépend étroitement de celle-ci. Les formats d'images numériques utilisés en reconnaissance sont : TIFF (Tagged Image File Format), JBIG, PNG et JPEG2000. Cependant, dans la communauté du document, le format d'image de prélédiction reste le format TIFF en raison de la compression sans perte mais aussi en raison de la stabilité de sa spécification qui remonte à 1992.

Le résultat de la reconnaissance de documents doit refléter aussi bien la structure physique que logique. Il est nécessaire que ces informations soient structurées. La représentation des documents sous forme structurée facilite l'édition et la mise à jour du contenu, le contrôle de la présentation et la recherche d'information.

Il existe un certain nombre de formats de représentation des résultats de reconnaissance. Il y a ceux qui sont conçus dans ce but alors que d'autres formats sont adaptés pour atteindre cet objectif. Dans le chapitre 3 nous passerons en revue ces formats.

1.2.3 Applications de la reconnaissance d'images de documents

Dans le monde entier, il existe un nombre faramineux de documents papiers. Ces derniers sont de différents types tels que les journaux, les revues, les livres, les encyclopédies, etc… Un grand volume de ces documents est conservé dans des bibliothèques nationales. Le reste est stocké soit dans les administrations, soit dans des musées, soit dans les bibliothèques universitaires, soit dans les entreprises et, à un degré moindre, dans nos bibliothèques personnelles.

Afin de préserver ces documents de tout genre de détérioration ou d'éventuelle décomposition, qui pourraient survenir, des techniques de conservation sont nécessaires. Le revers de la médaille, c'est que ces techniques sont assez coûteuses et que la pérennité de ces documents n'est pas assurée.

La numérisation permet de s'affranchir des problèmes posés par la conservation des documents papier. Le coût de stockage des documents électroniques est inférieur au coût de stockage des mêmes documents au format papier. Il est à noter que le coût de duplication des documents électroniques se trouvant dans un cédérom est bon marché et de cette façon nous assurons une conservation plus longue avec les documents électroniques par rapport aux documents au format papier. Néanmoins la numérisation des documents papiers nécessite des ressources matérielles et humaines qui engendrent des coûts à supporter.

La numérisation seule est insuffisante pour l'extraction d'information du document électronique ; en revanche, elle permet de préparer le terrain à la reconnaissance. La reconnaissance de documents permet l'extraction d'information et porte essentiellement sur les documents papier. Les travaux de recherche portant sur la reconnaissance de documents ont fait des grandes avancées dans ce domaine. Cependant, le domaine de la reconnaissance de documents n'est pas encore un problème résolu.

L’apport de la numérisation et de la reconnaissance de documents est indiscutable et a contribué à étoffer Internet. En effet, la reconnaissance d'images numérisées et d'images synthétiques est importante du point de vue de l'information extraite, puisqu'elle permet d'indexer un grand nombre de documents. Une vision réaliste serait qu'un jour, avec l'intégration des processeurs dans tous les équipements et la démocratisation de la communication sans fil, rechercher dans les documents par les mots clés en utilisant un moteur de recherche Internet, et ce depuis n'importe quel appareil intelligent, ne devienne intuitif.

Les applications de la reconnaissance de documents couvrent plusieurs domaines. En effet, nous trouvons des applications de reconnaissance de codes et adresses postales, de reconnaissance de chèques, de reconnaissance de formulaires, d'archivage de documents et de reconnaissance de factures médicales.

Les applications de reconnaissance de codes postaux sont très répandues dans les centres de tri postaux. Elles se basent sur la reconnaissance de l’écriture cursive manuscrite [92, 106]. Parmi ces applications, nous trouvons celles relatives à la reconnaissance manuscrite de chiffres [78, 103].

Une autre application utile, est la reconnaissance de formulaires [71, 86]. Les formulaires sont généralement constitués d’une partie fixe et d’une partie variable. La reconnaissance de formulaires repose sur la séparation entre ces deux parties.

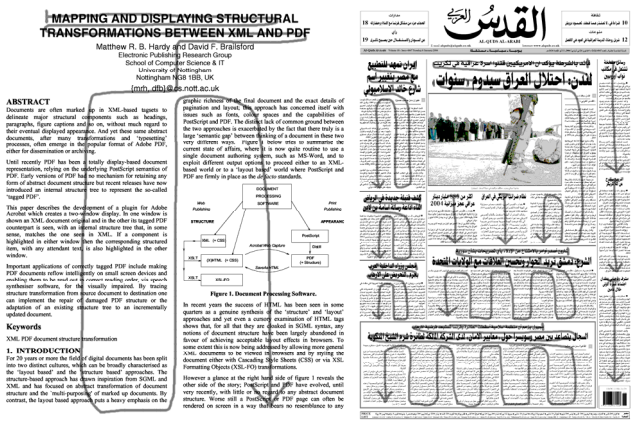
Les applications d’archivage de documents traitent assez bien les documents électroniques. Elles se focalisent sur deux axes : il y a celles qui stockent les documents dans leur formats d’origine et celles qui convertissent le document dans un autre format commun. Généralement le stockage de ces documents archivés, nécessite le recours aux bases de données.

Le taux de récupération des documents archivés à partir d'une base de données se trouve nettement amélioré si une extraction des structures et une indexation des données du document ont été effectuées. L'extraction des structures permet d'améliorer considérablement la pertinence de la recherche et aussi de faciliter la réédition de documents. Et il s'ensuit systématiquement une réduction de la surabondance de l'information dans la toile.

Comme application permettant l'extraction des structures, Dengel [28] a développé smartFIX, un système d'analyse et de compréhension de documents. Il permet le traitement des factures médicales. Ces dernières sont des documents composites ; elles renferment des tables, du texte et des annotations manuscrites. smartFIX a été développé dans le but de faciliter le traitement de ce genre de documents pour une institution d'assurance maladie. Il est à noter que cette dernière reçoit un nombre élevé de factures qui possèdent une grande variabilité d'un médecin à un autre et d'un laboratoire d'analyses médicales à un autre.

1.2.4 Reconnaissance de documents à structures complexes

La reconnaissance de documents à structures complexes est une tâche ardue. Une illustration de la complexité de structures physiques est la reconstitution de l'ordre de lecture. En effet, la reconstitution de l'ordre de lecture peut être définie pour une instance d'une classe de documents. En revanche, elle ne peut être généralisée pour toutes les classes de documents à structures complexes. Il est à noter que l'ordre de lecture peut être dans certains cas partiel. Dans ce cas, nous ne pouvons pas définir un ordre total des articles. La figure 1.5 illustre une reconstitution manuelle de l'ordre de lecture pour un document à structures simples et un document à structures complexes en langue arabe.



a) Structures simples b) Structures complexes

Figure 1.5 : Reconstitution de l'ordre de lecture pour un document à structures simples en langue latine (a) et pour un document à structures complexes en langue arabe (b).

La reconstitution de l'ordre de lecture fait partie de la reconnaissance de structures logiques. Comme nous l'avons mentionné, cette étape est simple pour les documents à structures simples mais reste difficile et complexe pour les documents à structures complexes.

1.3 Systèmes doués d'apprentissage

Les systèmes de reconnaissance de documents renferment les systèmes automatiques sans apprentissage et les systèmes automatiques ou semi-automatiques dotés d'apprentissage. Les systèmes de reconnaissance automatique de documents sont des systèmes autonomes. Il s'est avéré après de multiples expériences, que réaliser un système de reconnaissance automatique capable de reconnaître n'importe quel document est du domaine de l'irréalisable, vu la grande diversité des documents existant. La reconnaissance assistée est la voie à suivre dans ce cas. Cependant, quand les documents à reconnaître ne possèdent pas une grande variabilité, c'est la reconnaissance automatique qui est la plus adaptée.

Globalement les systèmes de reconnaissance de documents peuvent être des systèmes spécialisés ou génériques :

- Les systèmes spécialisés traitent un type de document particulier et ils possèdent un bon taux de reconnaissance,
- Les systèmes génériques traitent un large ensemble de documents. Cependant, l'amélioration du taux de reconnaissance est difficile et atteint rarement celui obtenu par les systèmes spécialisés, vu la large gamme de documents traités. Ces systèmes peuvent être statiques ou dynamiques. Les premiers sont des systèmes à base de règles ou à base d'apprentissage initial. Les seconds sont des systèmes à base d'apprentissage évolutif.

Les systèmes génériques statiques à base de règles sont plus sensibles au moindre changement de la structure du document. En effet, il suffit d'une petite modification de la structure de pages du document pour qu'une règle ne soit plus vérifiée. En revanche, les systèmes génériques à apprentissage initial sont plus prometteurs. Certes, l'apprentissage permet au système d'être plus robuste au moindre changement de la structure de pages du document. L'inconvénient de ces systèmes génériques à apprentissage unique ce sont les corrections répétitives de la part de l'utilisateur.

Les systèmes génériques d'apprentissage unique et évolutif reposent sur la présentation de plusieurs échantillons d'apprentissage. Ces échantillons doivent être le plus représentatif possible du modèle à traiter afin de permettre d'atteindre un bon taux de reconnaissance.

1.4 Caractéristiques de la langue arabe

La langue arabe a vu ses racines naître dans la péninsule Arabique et les premières traces de l'écriture arabe, telle qu'on la connaît ne remontent qu'au VIe siècle. Cette langue a vécu une expansion extrêmement rapide et a relié un immense empire recouvrant le Proche-Orient, l'ensemble de l'Afrique du nord, l'Espagne et la Sicile. L'expansion et le développement de cette langue sont intimement liés à la naissance et la diffusion de l'islam. Néanmoins cette langue ne s'est pas limitée au texte coranique, mais elle est devenue une langue de culture, de philosophie, de sciences et de techniques allant jusqu'à supplanter les autres langues locales lors de l'expansion arabo-musulmane.
La langue Arabe est une langue parlée par à peu près 300 millions de personnes. Elle est la langue officielle ou figure parmi les langues officielles de 19 pays. Il y a deux types d'écritures possibles pour la langue arabe :

- l'écriture classique qui correspond à l'écriture du Coran et la littérature classique,
- l'écriture universelle du monde arabe actuel.

En ce qui concerne l'arabe parlé, chaque pays ou région possède un arabe parlé dialectal. Il arrive que cet arabe parlé dialectal recouvre en fait plusieurs dialectes différents tous issus de l'arabe classique. L'arabe dialectal se subdivise en trois grands groupes :

- les dialectes arabiques : parlés dans la péninsule arabique,
- les dialectes maghrébins : algérien, marocain et tunisien,
- les dialectes proche-orientaux : égyptien et syro libano palestinien.

L'arabe appartient à la famille des langues sémitiques [42] comme l'hébreu et l'araméen, au sein desquelles essentiellement les consonnes sont représentées en écriture, néanmoins nous notons la présence de voyelles. L'alphabet arabe est composé de 28 lettres telles qu'elles sont illustrées dans la figure 1.6. Les mots sont écrits sur des lignes horizontales de la droite vers la gauche, par contre les chiffres sont écrits de la gauche vers la droite. La figure 1.7 illustre les chiffres utilisés dans la langue arabe.

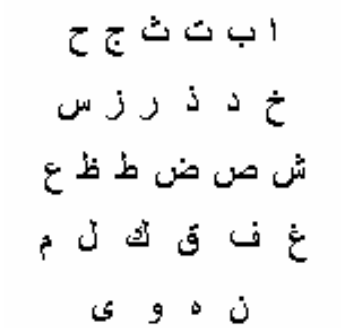


Figure 1.6 : Les 28 lettres de l'alphabet arabe.

Figure 1.7 : Les chiffres.

La lettre arabe change de forme selon sa position dans le mot [20]. Elle ne s'écrit donc pas de la même manière au début, au milieu ou en fin de mot. La figure 1.8 illustre les différentes lettres arabes ainsi que leurs formes au début, au milieu ou en fin de mot. Il est à noter la présence des points diacritiques rattachés aux lettres dont le nombre varie de un à trois points. Un nombre important de mots en arabes sont composés uniquement de consonnes. Néanmoins, nous notons la présence de voyelles au sein des mots. Les voyelles se subdivisent en deux catégories : voyelles longues et brèves.

Finale	Médiane	Initiale	Isolée	Finale	Médiane	Initiale	Isolé
ـا	ـا	ـا	ا	ـض	ـضـ	ـضـ	ض
ـب	ـبـ	ـبـ	ب	ـط	ـطـ	ـطـ	ط
ـت	ـتـ	ـتـ	ت	ـظ	ـظـ	ـظـ	ظ
ـث	ـثـ	ـثـ	ث	ـع	ـعـ	ـعـ	ع
ـج	ـجـ	ـجـ	ج	ـغ	ـغـ	ـغـ	غ
ـح	ـحـ	ـحـ	ح	ـف	ـفـ	ـفـ	ف
ـخ	ـخـ	ـخـ	خ	ـق	ـقـ	ـقـ	ق
ـد	ـد	ـد	د	ـك	ـكـ	ـكـ	ك
ـذ	ـذ	ـذ	ذ	ـل	ـلـ	ـلـ	ل
ـر	ـر	ـر	ر	ـم	ـمـ	ـمـ	م
ـز	ـز	ـز	ز	ـن	ـنـ	ـنـ	ن
ـس	ـسـ	ـسـ	س	ـه	ـهـ	ـهـ	ه
ـش	ـشـ	ـشـ	ش	ـو	ـو	ـو	و
ـص	ـصـ	ـصـ	ص	ـي	ـيـ	ـيـ	ي

Figure 1.8 : Les lettres et leurs formes dans un mot.

Les voyelles longues sont composées de trois lettres alif (ا), waaw (و) et yaa (ي). En revanche, les voyelles brèves sont facultatives. Nous distinguons une forme particulière d'une voyelle brève à savoir la double voyelle. La figure 1.9 illustre les types de voyelles de la langue arabe.

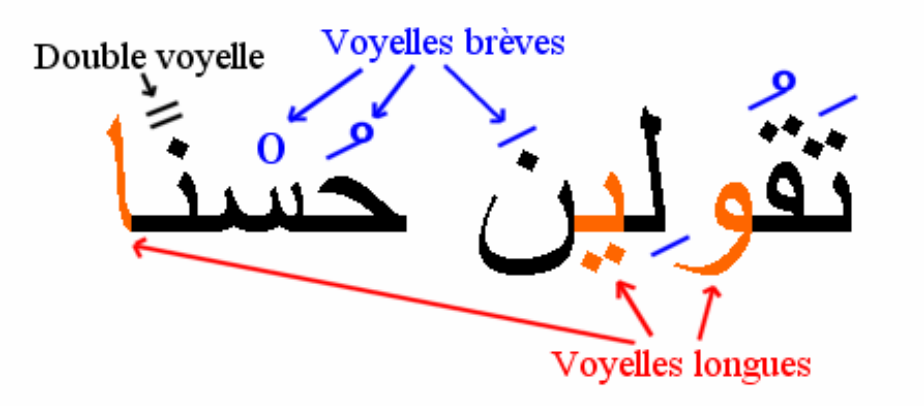


Figure 1.9 : Les voyelles de la langue arabe.

Les voyelles sont utilisées dans le but d'assurer qu'un texte est lu à haute voix sans erreurs de prononciation. Certains livres tels que le coran, la poésie et les livres d'apprentissage de la langue arabe utilisent les voyelles de manière systématique.

L'écriture arabe est curviligne, composée de consonnes, dans la grande majorité et de voyelles longues, liées entre elles par des ligatures. Les mots sont constitués d'un regroupement de lettres. Chaque lettre possède une représentation selon la position dans laquelle elle se trouve : initiale, médiane ou isolée. La composition du mot "bienvenue" en arabe repose sur le regroupement de cinq lettres ; la figure ci-dessous illustre ce mot.

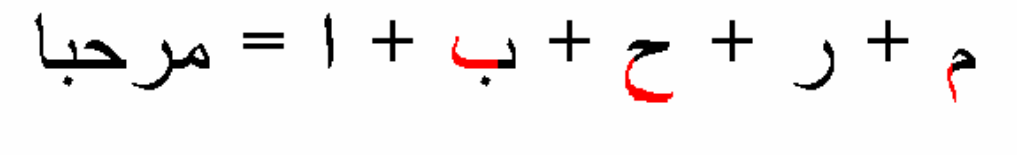


Figure 1.10 : La composition des mots en langue arabe.

Le mot "bienvenue" en arabe figurant dans la figure 1.10 est composé de deux pseudo-mots, dits aussi fraction d'un mot. En effet, un mot en arabe peut comprendre un ou plusieurs pseudo-mots composés d'un certain nombre de caractères différents.

Il y a un cas particulier dans la composition des mots en langue arabe et c'est le cas de la *chadda*. La présence d'une *chadda* au dessus d'une lettre indique que cette lettre est doublée. La figure 1.11 montre le dédoublement de lettre : *chadda*.

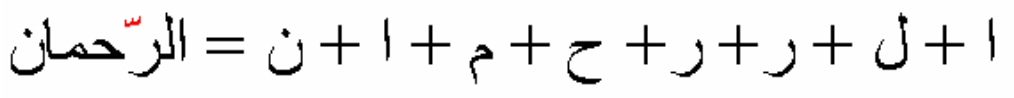


Figure 1.11 : Le dédoublement de lettre : *chadda*.

1.5 Objectifs de cette thèse

L'objectif de cette thèse est d'étudier l'évolutivité des modèles dans un contexte interactif pour la reconnaissance de structures physiques et logiques de documents riches en structures et en variabilité.

1.5.1 Documents à structures complexes

Les algorithmes de reconnaissance de structures physiques des documents à structures simples donnent de bons résultats. Cependant l'application de ces algorithmes aux documents à structures complexes est loin d'être satisfaisante. Ceci peut s'expliquer par la grande variabilité de structures physiques intra-classes et inter-classes. Néanmoins, il ne faut pas sous-estimer les méthodes simples, elles permettent de reconnaître les documents à structures complexes dans le cas général.

La langue arabe n'a pas eu un grand engouement au niveau des travaux de recherche pour la reconnaissance de documents. En effet, à notre connaissance, nous avons constaté qu'il n'y a pas eu de travail de recherche sur la reconnaissance de structures physiques des documents à structures complexes en langue arabe. De ce fait, nous avons étudié les spécificités de la reconnaissance de structures physiques, pour les documents complexes en langue arabe.

1.5.2 Choix en accord avec la philosophie *CIDRE*

Depuis 1994, les travaux du groupe de recherche DIVA (Document, Image and Voice Analysis) sont orientés par le projet *CIDRE*[1]. [15, 16, 17, 18, 19, 23, 35, 36, 37, 38, 39, 40, 41, 45, 46, 47, 82, 83, 84]. *CIDRE*, acronyme pour Cooperative & Interactive Document Reverse Engineering, est fondé sur une révision de toute la problématique en reconnaissance de documents qui s'organise selon quatre principes : réingénierie de documents, reconnaissance assistée, rôle de l'architecture logicielle et modélisation de classes de documents.

Les recommandations de *CIDRE* stipulent que :

- les outils et systèmes de reconnaissance de documents doivent être adaptatifs à tout changement de classes de documents. En ce qui concerne la reconnaissance assistée, l'idée projetée est de permettre à l'utilisateur de corriger les erreurs issues de la reconnaissance de documents et, ce, au cours des étapes de la reconnaissance de documents,
- l'architecture logicielle recommandée est une architecture modulaire et coopérative. En effet, il est préférable d'avoir de petits outils indépendants facilement intégrables et incluant une coopération homme machine. Enfin pour la modélisation de classes de documents, la recommandation stipule que les modèles de documents devraient être générés d'une manière évolutive.

1.5.3 Apprentissage évolutif

Nous rappelons que la philosophie CIDRE préconise une reconnaissance assistée et adaptative. C'est dans cette optique que plusieurs travaux de recherche, au sein de notre groupe de recherche DIVA, se sont attelés à respecter cette philosophie. Parmi ces travaux nous citons la méthode *2(CREM)* [83] qui est dotée d'un apprentissage évolutif et permet la reconnaissance aussi bien de structures physiques que logiques.

2(CREM) a été testée aussi bien pour la reconnaissance de structures physiques que logiques et, cela, pour une seule classe de documents à structures complexes : le journal Los Angeles Times. *2(CREM)* a montré la pertinence de l'ajout de l'apprentissage évolutif et ce en intégrant l'interactivité. Après avoir étudié en profondeur *2(CREM)* nous nous sommes rendu compte de la présence de certaines lacunes. Premièrement, au niveau

[1] Projet financé par le fond national Suisse pour la recherche scientifique, code 2000-059356.99-1

de l'interactivité : l'utilisateur corrige les erreurs de segmentation physique et logique en procédant seulement à un étiquetage et ne peut en aucun cas effectuer d'autres opérations comme la fusion ou le découpage d'entités mal segmentées. Deuxièmement, au niveau du choix des caractéristiques un mauvais choix d'une caractéristique supplémentaire, pendant la phase d'apprentissage, fait de telle sorte qu'elle soit non discriminante pour les classes.

La première expérience réussie avec *2(CREM)*, montre la pertinence de l'apprentissage évolutif et ce dans un environnement interactif. Nous allons continuer dans cette direction, à savoir évaluer les techniques de reconnaissance et développer deux systèmes de reconnaissance de structures physiques et logiques de classes de documents riches en structures et en variabilité. Dans notre cas, nous nous sommes intéressés aux journaux arabes. Les systèmes développés sont dotés d'un apprentissage évolutif qui leur permet de fournir une certaine autonomie et d'être efficace pour des applications de reconnaissance de taille moyenne. Ces systèmes sont aussi dotés d'une interactivité plus élaborée avec l'utilisateur, en lui fournissant un outil de correction adéquat de structures physiques, comme la fusion ou le découpage d'entités mal segmentées et un outil d'étiquetage logique des entités. Et d'un choix de caractéristiques plus pertinentes et indépendantes de la classe de documents.

Nous voulons que les deux systèmes permettent entre autres la construction de fonds de vérité.
Les systèmes développés utilisent l'interface graphique xmillum [47]. Celle-ci permet la visualisation, la correction et l'étiquetage, d'une manière interactive, des résultats de la reconnaissance de structures physique et logiques de documents.

1.6 Organisation en chapitres

Le contenu de cette thèse est organisé en sept chapitres.

Dans le chapitre 2 nous allons discuter de l'état de l'art en reconnaissance de documents. Ce chapitre est subdivisé en deux sections, la première traite la reconnaissance de structures physiques alors que la deuxième traite la reconnaissance de structures logiques. Nous détaillerons les travaux de recherche relatifs à la reconnaissance des documents à structures complexes, aux systèmes dotés d'apprentissage et à la construction de fonds de vérité.

Le chapitre 3 décrit l'architecture générale ainsi que le format de représentation des résultats intermédiaires. Nous parlerons des différents formats de représentation des résultats de reconnaissance utilisés dans les travaux de recherche et nous décrirons les formats de représentation des résultats intermédiaires.

Dans le chapitre 4, nous décrivons notre adaptation à la langue arabe des méthodes simples traditionnellement utilisées pour la reconnaissance de documents à structures simples en langue latine. Cette adaptation est faite pour prendre en considération les

caractéristiques de la langue arabe et, ce, dans un environnement composé de documents à structures complexes. Nous y décrivons en détail les tâches suivantes : l'extraction des filets, l'extraction des cadres, la séparation texte image, l'extraction des lignes de texte et la fusion des lignes de texte en blocs.

Ceci nous amène au chapitre 5 dans lequel nous présentons *PLANET* [2] en détail. Nous montrons comment *PLANET* s'est doté de l'apprentissage évolutif pour la reconnaissance de structures physiques de plusieurs classes de documents en langue arabe. Ensuite, nous décrivons en détail l'évaluation de *PLANET* avec la tâche de la fusion des lignes de texte en blocs ainsi que les résultats obtenus.

Dans le chapitre 6, nous décrivons *LUNET* [3], un système de reconnaissance de structures logiques doté d'apprentissage évolutif. Nous décrivons en détail la procédure d'évaluation de *LUNET* avec l'étiquetage des blocs de texte et les résultats obtenus.

Finalement, le chapitre 7 conclut cette thèse en rappelant le travail scientifique qui a été réalisé et les extensions futures qui pourraient être entreprises dans des travaux ultérieurs.

[2] pour Physical Layout Analysis of classes of documents using artificial neural NETs.
[3] pour Logical layout analysis of classes of documents Using artificial neural NETs.

Chapitre 2

État de l'art

L'homme a réussi à matérialiser son moyen de communication naturel la parole, en un support physique à savoir le document. Ce dernier a permis à l'homme de transmettre son savoir et ses pensées de génération en génération. Certes, le document a permis l'essor des sciences dans toutes les disciplines. Vu son importance, le document a subi de nombreuses améliorations au fil des années mais son rôle de communication est resté inchangé.
L'avènement de l'ère de l'information n'a pas ralenti ni réduit le nombre de documents circulant partout dans le monde. En effet, l'informatique a cherché à récupérer les centaines de millions de documents au format papier, en les numérisant, et à en extraire leurs contenus. Et c'est à ce moment que la reconnaissance d'images de documents a vu le jour.
Les performances des méthodes de reconnaissance d'images de documents s'améliorent de jours en jours. En effet, plusieurs problèmes ont été surmontés et la recherche est abondante, cependant nous sommes encore loin de la perfection. Le taux de reconnaissance atteint rarement la barre fatidique des 100%.
La plupart des travaux de reconnaissance d'images de documents ont été appliqués sur des documents en langues latines. Et bien que trois cents millions de personnes parlent la langue arabe, celle-ci n'a pas attiré beaucoup de chercheurs. La plupart des travaux de recherche qui existent, se sont focalisés sur la reconnaissance de l'écriture arabe imprimée et manuscrite.

En raison de l'absence dans la littérature des travaux de recherche décrivant des méthodes de reconnaissance de structures physiques et logiques de documents en langue arabe riches en structures et en variabilité, nous allons passer en revue, dans ce chapitre, l'état de l'art pour la reconnaissance de structures physiques et logiques de documents en langue latine, et l'état de l'art pour la reconnaissance optique de caractères de documents arabes pour nous inspirer de ces travaux.

2.1 Reconnaissance de structures physiques

Dans cette section nous allons passer en revue l'état de l'art pour la reconnaissance de structures physiques. A cet effet, nous passerons en revue l'état de l'art concernant :

- les méthodes de reconnaissance de structures physiques ascendantes, descendantes et mixtes,
- la segmentation des documents à structures complexes,

- les méthodes intégrant des techniques d'apprentissage,
- la construction de fonds de vérité,
- la mesure de performances des algorithmes de reconnaissance de structures physiques.

2.1.1 Introduction

La structure physique d'un document comporte des entités qui diffèrent d'un document à un autre. En revanche une entité reste commune à tous les documents ; le texte. Les autres entités que nous retrouvons sont les images et les graphiques. En effet, ces dernières sont généralement insérées au sein des documents soit pour étoffer le document, soit pour expliquer une partie du texte.

Il est à rappeler que le but de la reconnaissance est de faire de la réingénierie. Il en va de même pour la reconnaissance de structures physiques d'une image de document. Cette réingénierie de documents permet de retrouver les entités constituant le document. La reconnaissance de structures physiques comprend deux étapes ; la détection et la classification des différentes zones de l'image. La détection comprend la segmentation qui consiste en la découpe, en zones de l'image, du document par des formes géométriques en 2D (rectangles ou polygones). Dans l'étape de classification les formes géométriques, obtenues dans l'étape précédente, sont étiquetées en tant que texte, images et graphiques. La décomposition en texte peut être affinée en mots, en lignes et en blocs de texte. En effet, c'est à l'étape de segmentation que nous décidons cet affinement.

Dans la littérature, plusieurs articles se sont intéressés à l'état de l'art pour la reconnaissance de structures physiques. Certains de ces articles présentent uniquement l'état de l'art pour les méthodes de reconnaissance de structures physiques [67, 89]. D'autres, présentent conjointement l'état de l'art des méthodes de la reconnaissance de structures physiques avec celles relatives aux structures logiques [24, 43, 51, 65, 91]. Il est à noter que Mao [65] et Jain [51] dressent un tableau comparatif des algorithmes d'analyse de structures physiques de documents et résument les avantages et les limites de certaines approches.

Les méthodes de la reconnaissance de structures physiques, décrites dans la littérature sont des méthodes automatiques et peuvent être classées en deux grandes classes : les méthodes descendantes et les méthodes ascendantes [89]. Les méthodes descendantes commencent par le niveau le plus élevé à savoir la page et descendent d'un niveau à un autre jusqu'à arriver au niveau des composantes connexes ou au niveau pixel. Par contre les méthodes ascendantes reposent sur des fusions successives de composantes connexes du plus bas niveau vers le niveau le plus élevé. En effet, la dernière fusion permet la reconstitution de la page une fois les fusions du bas niveau ont été effectuées. Actuellement, une troisième classe figure parmi les méthodes descendantes et ascendantes : les méthodes mixtes. Celles-ci combinent les deux approches : descendantes et ascendantes pour la reconnaissance de structures physiques. La figure 2.1 illustre l'approche descendante et ascendante.

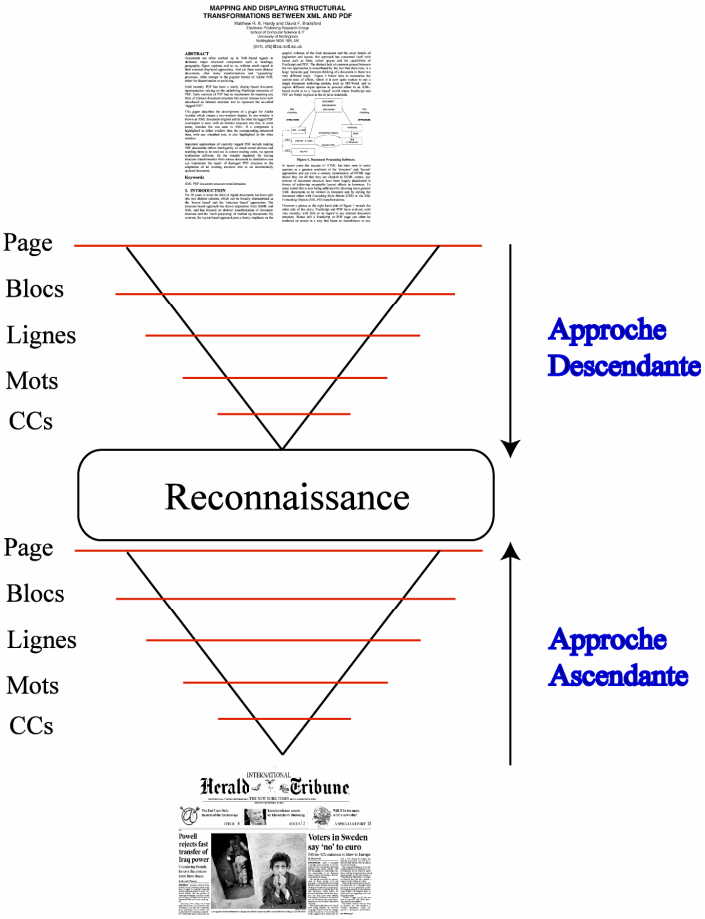


Figure 2.1 : Approche descendante et ascendante

2.1.2 Méthodes descendantes

Les méthodes descendantes commencent par le niveau le plus élevé à savoir la page et descend d'un niveau à un autre jusqu'à arriver au niveau des composantes connexes ou au niveau pixel. Un exemple d'algorithme utilisant la stratégie descendante est le fameux algorithme de découpage X-Y [69]. Ce dernier est plus approprié aux structures de type Manhattan. Une structure de page de type Manhattan est celle dont les régions de la page sont tous des rectangles et que les rectangles possèdent la même orientation. En réalité ce nom fait référence à la disposition des bâtiments de quartier de Manhattan à New york. La figure 2.2 illustre une structure de page de type Manhattan.

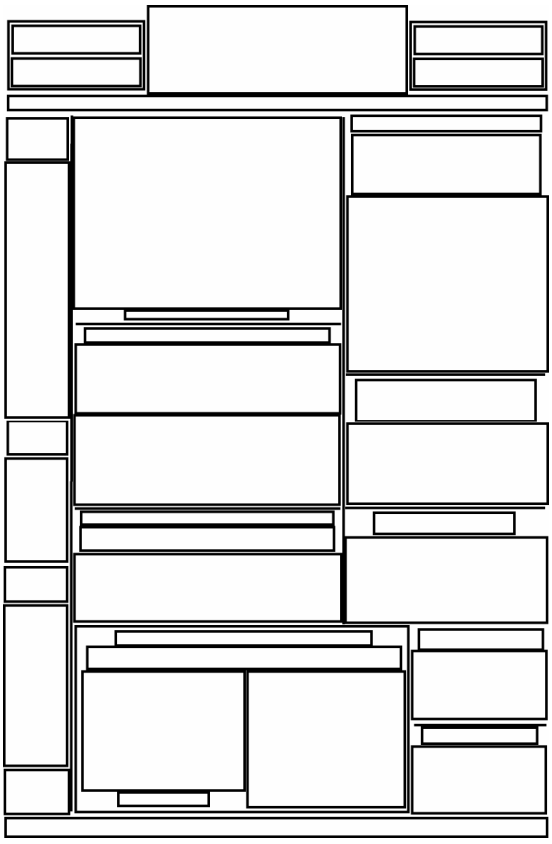


Figure 2.2 : Structure de page de type Manhattan

Dans les sous-sections suivantes nous allons passer en revue l'état de l'art en ce qui concerne les méthodes descendantes utilisant :

- l'algorithme de découpage X-Y,
- l'algorithme de lissage RLSA,
- l'analyse du fond blanc de l'image.

2.1.2.1 Méthodes utilisant l'algorithme de découpage X-Y

L'algorithme de découpage X-Y fait partie des méthodes descendantes. Il utilise les méthodes de profils de projection et a été introduit par Nagy [69]. L'hypothèse de base repose sur le fait que les éléments structurés de la page sont généralement présentés dans des blocs rectangulaires. Mais aussi sur le fait que les blocs peuvent être divisés en groupes de telle sorte que les blocs qui sont adjacents l'un à l'autre, dans un groupe, ont une dimension en commun. Le document est successivement divisé en de petits blocs rectangulaires en faisant une alternance de découpages horizontaux et verticaux le long des espaces blancs. Ces espaces blancs sont trouvés en utilisant un seuil de profil de projection. Le résultat d'une telle segmentation peut être représenté dans un arbre X-Y, dans lequel la racine correspond à la page toute entière et les feuilles représentent les blocs de la page et chaque niveau de l'arbre représente alternativement les résultats de la segmentation horizontale ou verticale. Une version améliorée a été introduite de nouveau

par Nagy [68] en proposant une alternance de découpages en utilisant les deux stratégies de segmentation ascendante et descendante.

Akindele [4] présente une amélioration par rapport à l'algorithme de découpage X-Y. En effet, un ensemble de règles topologiques ont été introduites pour résoudre le problème de la globalité de l'algorithme de découpage X-Y et pour permettre aux blocs d'être segmentés aussi bien en rectangles qu'en polygones à travers la méthode de suivi de segments.

Cesarini [25] a proposé une version modifiée de l'algorithme de découpage X-Y pour la segmentation des factures et des documents techniques comportant des tables. En effet, la modification porte sur le stockage des lignes de découpages lors de la sortie de l'arbre d'exploration et aussi sur la description des relations d'adjacence pour les inter espacements au moyen de liens appropriés.

2.1.2.2 Méthodes utilisant l'algorithme de lissage RLSA

D'autres méthodes descendantes utilisent l'algorithme de lissage RLSA (Run Length Smearing Algorithm) [95]. Wang [99] propose une méthode basée sur l'utilisation conjointe de l'algorithme RLSA et de l'algorithme de découpage X-Y récursif pour segmenter l'image en blocs. Cette segmentation est suivie par une étape d'analyse de la texture pour classer les blocs résultants. Cette méthode a été appliquée aux journaux et s'avère sensible à la rotation de l'image du document.

Yamashita [104] décrit un système de reconnaissance de documents qui facilite la conversion des documents de la forme imprimée vers la forme électronique. Les documents utilisés sont en Kanji. L'approche utilisée est basée sur l'algorithme de lissage RLSA avec un seuil adaptatif. De ce fait, le moindre changement dans l'espacement entre les mots et au niveau de la taille des fontes affecte peu le résultat de la segmentation.

2.1.2.3 Méthodes utilisant l'analyse du fond de l'image

Les travaux de recherche utilisant les méthodes descendantes s'orientent vers l'analyse du fond de l'image et plus exactement les zones blanches de celle-ci. En effet, Spitz [88] a été le premier à utiliser cette approche. La méthode recherche les flux blancs dans les deux directions verticale et horizontale, et les exploite comme délimiteur générique de structures.

Pavlidis [76] utilise une approche similaire à celle proposée par Spitz [88] tout en présentant une amélioration au niveau de la performance de l'algorithme. D'abord, le profil de projection vertical est calculé et la plus longue plage de valeurs d'intervalles blancs est recherchée. Puis les colonnes d'intervalle sont converties en des colonnes de blocs, tout en fusionnant les petits blocs en des blocs plus grands. Finalement, les blocs sont étiquetés en texte ou non texte en utilisant des caractéristiques. Parmi celles-ci nous

trouvons le ratio de la moyenne des longueurs de l'intervalle noir sur la moyenne des longueurs de l'intervalle blanc, le ratio du nombre d'intervalles noirs sur une certaine longueur et le nombre total d'intervalles.

L'approche proposée par Baird [13] est basée sur la constitution de l'ensemble des rectangles blancs appelés couverture. L'algorithme accepte en entrée un ensemble de rectangles noirs représentant les rectangles englobants des composants connexes et en sortie l'ensemble de rectangles blancs maximaux, résultant de l'union des petits rectangles blancs. Un rectangle blanc maximal est un rectangle qui ne peut pas être étendu tout en restant blanc partout. L'algorithme de l'union des rectangles blancs repose sur une heuristique et donne en sortie les rectangles blancs maximaux. L'heuristique repose sur une règle pour l'arrêt de la fusion des rectangles blancs.

Antanacopoulos [11] présente la méthode de pavage. Cette méthode consiste en la recherche de la plus longue plage de valeurs de pixels dans le sens vertical pour noircir les zones. En effet, il s'agit d'un lissage vertical. Des rectangles de différentes tailles, sont utilisés pour couvrir le fond de l'image. L'extraction intervient en considérant les bords des rectangles coïncidant avec les bords des zones noircies. L'inconvénient de cette méthode est qu'elle classe mal les fontes ayant une taille élevée.

Antanacopoulos [8] a améliorée sa propre méthode [11]. Celle-ci ressemble à celle proposée par Pavlidis [76]. Parmi les avantages de cette méthode, c'est la faculté de traiter les documents ayant subit des rotations et dont les régions ne sont pas rectangulaires. Le processus est le suivant : scanner chaque ligne de l'image du document en recherchant la plus longue plage de valeurs de pixels blancs. Ensuite, ces plages de valeurs de pixels blancs sont fusionnées pour former des carreaux blancs d'une manière séquentielle de haut en bas. Un carreau blanc est représenté par un rectangle dont les dimensions verticales et horizontales sont ajustées pour englober la plus longue plage de valeurs de pixels blancs dans la direction horizontale.

Bruel [21] présente deux algorithmes pour la résolution des problèmes relatifs à l'analyse géométrique de structures de documents. Le premier algorithme est utilisé soit pour l'analyse des espaces blancs, soit pour l'analyse de la structure du fond de l'image en termes de couverture rectangulaire. Cet algorithme est simple à implémenter et ne requiert aucune structure géométrique de données. Le deuxième algorithme est utilisé pour la détection de lignes de texte en présence d'obstacles. Ces deux algorithmes entrent dans une nouvelle approche pour la segmentation de documents.

2.1.3 Méthodes ascendantes

Les méthodes ascendantes commencent par le niveau le plus bas et remontent d'un niveau à un autre jusqu'à compléter la page. En effet, elles se basent sur l'analyse des composantes connexes. Ces dernières sont obtenues en scannant une image pixel par pixel et en regroupant les pixels en des composants en se basant sur la connexité des pixels qui peut être en 4 voisins ou en 8 voisins.

Le principe des méthodes ascendantes est le suivant : elles commencent par fusionner du plus bas niveau, en formant les mots à partir des composantes connexes, et puis remontent à un niveau supérieur en fusionnant les mots en lignes, les lignes en blocs, etc… jusqu'à ce que la page soit complètement reconstituée.

Les méthodes ascendantes sont typiquement des variantes de la méthode des composantes connexes. L'inconvénient de la méthode des composantes connexes est qu'elle est sensible à l'interligne, à l'espacement entre caractères et à la résolution de la numérisation.

Cependant elle n'est pas limitée aux blocs de forme rectangulaire contrairement aux méthodes descendantes.

Dans les sous-sections suivantes nous allons passer en revue l'état de l'art des méthodes ascendantes utilisant : les composantes connexes, le filtrage à base de fenêtres, la technique docstrum et les diagrammes de Voronoi.

2.1.3.1 Méthodes utilisant les composantes connexes

Fisher [31] a combiné l'algorithme de lissage avec l'extraction des composantes connexes. Les composantes connexes et leurs rectangles englobants constituent les blocs de la structure physique du document. Un ensemble de caractéristiques des composantes connexes est utilisé. Cette approche permet d'identifier les zones textes et non textes mais reste cependant sensible à la rotation de l'image du document.

Saitoh [85] procède par échantillon de 8x4 pixels sur toute l'image, puis extrait les composantes connexes. Ces dernières sont classées en texte, bruit, table, diagramme, image bitonale, ou filet, en utilisant les attributs des blocs tels que la hauteur du bloc, le ratio hauteur/largeur et la présence ou non de filets. Les blocs sont divisés selon les critères suivants : la distance verticale entre les lignes et la hauteur des lignes dans les blocs. L'échantillon de test de la méthode est composé de documents en langue Japonaise.

Drivas [29] présente une méthode de segmentation de pages d'images de documents. Celle-ci comporte un ensemble d'algorithmes. Le premier algorithme permet de déterminer l'angle de rotation, le deuxième permet la segmentation et le troisième permet l'étiquetage des blocs obtenus en texte et en image. L'algorithme de segmentation extrait les composantes connexes ensuite applique la fusion de ces derniers. L'approche de fusion repose sur la recherche des plus proches composantes connexes et le regroupement des composantes connexes ayant une même dimension. L'échantillon de test comporte 30 documents extraits de revues, de cartes de visites et de rapports techniques.

2.1.3.2 Méthodes utilisant le filtrage à base de fenêtres

Les méthodes ascendantes, utilisant le filtrage à base de fenêtres, reposent sur un balayage d'une fenêtre d'une certaine taille sur toute l'image du document. Lebourgeois [58] utilise un filtre de 8 x 3 pixels. L'image échantillonnée est dilatée par un élément de structure horizontale pour rassembler les caractères adjacents l'un vers l'autre. Il est à noter que chaque composante connexe est caractérisée par son rectangle englobant et par la moyenne des longueurs de plages de valeurs de pixels noirs. Ensuite, si la composante connexe est à l'intérieur de l'intervalle celle-ci sera classée en une zone de texte, sinon elle sera classée en zone non texte. Les composantes connexes classées en zone texte sont fusionnées verticalement en blocs selon des règles prenant en considération l'alignement.

2.1.3.3 Méthodes utilisant la technique docstrum

O'Gorman [73] introduit la technique "docstrum" qui est une technique d'analyse de structures physiques de page, basée sur la combinaison de l'analyse ascendante et du clustering qui fait intervenir le calcul des k plus proches voisins pour chaque composante connexe de la page. Chaque paire de voisins les plus proches possède un angle et une distance associée. En regroupant les composants à travers les caractéristiques citées précédemment, les régions géométriques de structures physiques de la page peuvent être déterminées. La méthode proposée est indépendante du changement de l'orientation de la page mais aussi de l'espacement intertexte. Cependant, la valeur du k est dépendante de la structure de la page.

2.1.3.4 Méthodes utilisant les diagrammes de Voronoi

Kise [56] présente une méthode de segmentation de pages basée sur la surface approximée des diagrammes de Voronoi. La méthode repose sur les étapes suivantes : au début, un point du diagramme de Voronoi est construit à partir de l'ensemble des pixels noirs sur les contours des composantes connexes. Ensuite, une surface est obtenue en éliminant du point du diagramme Voronoi tous les arêtes générées à partir d'une paire de points sur la même composante connexe. Une caractéristique distinctive de cette méthode est qu'elle s'applique sur des images de documents possédant une structure de type Manhattan et ayant subie une rotation. Il est à noter que cette méthode est efficace pour l'extraction des zones de texte et possède un taux de reconnaissance comparable à celui obtenu par les méthodes basées sur l'analyse des composantes connexes.

2.1.4 Méthodes mixtes

Les méthodes descendantes et ascendantes ont leurs limites. En effet, ces méthodes purement descendantes ou ascendantes donnent de bons résultats en présence de classes spécifiques de documents et peinent en présence d'autres types de classes. Devant ces

insuffisances mutuelles de deux types de méthodes, il y a eu naissance d'un nouveau type de méthode ; les méthodes mixtes. Les méthodes mixtes résultent de la combinaison des méthodes descendantes et ascendantes ou de l'utilisation conjointe d'une de ces dernières avec une autre méthode comme par exemple l'analyse syntaxique.

Dans les sous-sections suivantes nous allons passer en revue l'état de l'art en ce qui concerne les méthodes mixtes utilisant l'analyse syntaxique de documents et d'autres comprenant une combinaison de méthodes.

2.1.4.1 Analyse syntaxique des documents

Nous rappelons que l'analyse syntaxique tient ses origines dans la compilation des programmes informatiques écrits dans un langage de programmation à coté de l'analyse lexicale. Cette technique d'analyse a été approchée pour la reconnaissance de structures physiques de documents ; c'est une méthode guidée par un modèle. En effet, Nagy [70] propose un prototype d'un système d'analyse d'images de documents pour les revues techniques. Ce système utilise l'analyse syntaxique de documents en vue d'une génération d'une grammaire par type de documents. Il est à noter que les langages de programmation possèdent un ensemble de règles fixes. De manière analogue chaque publication d'une revue possède une structure prédéterminée de conventions. Ces conventions, exprimées dans une grammaire, spécifient la taille, la position, l'espacement et l'ordre d'apparition des blocs qui correspondent aux entités logiques dans la page. L'algorithme de base pour la segmentation repose sur le découpage en X-Y, ensuite intervient l'analyse syntaxique pour l'extraction des blocs. Viswanathan [94] utilise une approche similaire.

Krishnamoorthy [57] propose une méthode permettant la combinaison de la segmentation d'une image avec l'étiquetage. Cette méthode est différente des autres méthodes d'analyse syntaxique des documents en deux points. Le premier point c'est que les grammaires utilisées forment une hiérarchie. Le deuxième point c'est la combinaison de formules syntaxiques avec la méthode de recherche du "branch and bound".

2.1.4.2 Autres méthodes mixtes

Les autres méthodes mixtes englobent les différentes combinaisons possibles d'utilisations de méthodes descendantes, ascendantes et autres. La première méthode mixte a été introduite par Baird [14] qui se base sur l'analyse de la texture. L'approche adoptée par Esposito [30] consiste en l'utilisation de l'algorithme de lissage RLSA avec une méthode ascendante pour classer les blocs selon leurs contenus en utilisant un arbre de décision. En tout il y a cinq classes : le texte, les lignes horizontales et les lignes verticales, les images et les graphiques. La classification est basée sur l'évaluation des dix caractéristiques pour chaque bloc. Le reste des méthodes utilisent le principe de découpage et de fusion [12, 38, 62, 75].

Pavlidis [75] utilise une approche basée sur le découpage et la fusion. La méthode utilisée permet de distinguer entre les régions bitonales et non bitonales tout en permettant la séparation entre texte et graphiques. Cette séparation est possible en se basant sur les observations de la corrélation croisée de chaque ligne balayée dans un bloc avec la ligne de dessous. Le critère de densité de pixel noir est utilisé pour la séparation de texte et graphique.

Azokly [12] adopte la méthode basée sur le découpage et la fusion. L'algorithme de découpage est hiérarchique et il est basé sur l'analyse des rectangles blancs qui constituent le fond de l'image. La fusion s'effectue à travers des règles. Celles-ci décrivent les structures à reconnaître.

La méthode développée par Liu [62] est basée sur une approche de découpage et de fusion. En effet, le processus de découpage repose sur la séparation en des zones non homogènes et la fusion de ces derniers en des zones homogènes. Un seuil adaptatif est utilisé et qui permet de calculer les bordures de segmentation. C'est l'algorithme de découpage X-Y qui est utilisé comme méthode descendante. En revanche, la méthode ascendante comporte l'utilisation d'un seuil adaptatif pour calculer les bordures de segmentation.

Nous avons proposé [38] une approche similaire à celle proposée par Lui [62]. Nous nous sommes inspirés de l'algorithme de découpage X-Y de Nagy pour le découpage de l'image du document. Ce découpage est effectué après avoir extrait les filets horizontaux et verticaux à partir d'une méthode ascendante. L'image découpée en petites régions est fusionnée pour former des régions plus grandes.

2.1.5 Segmentation des documents à structures complexes

Les premières méthodes de reconnaissance de structures physiques ont été appliquées à des documents à structures simples. Des travaux récents montrent un engouement pour la reconnaissance de documents à structures complexes. La principale difficulté inhérente à ce type de documents est la variabilité de structures physiques intra-classes et inter-classes.

La première expérience a été menée pas Wang [99]. Celle-ci repose sur l'utilisation de l'algorithme RLSA avec l'algorithme de découpage X-Y récursif pour segmenter l'image en blocs. Ensuite, intervient une étape d'analyse de la texture pour classer les blocs résultants.
Govindaraju [34] utilise une technique similaire à celle de Wang [99] pour la classification mais il fusionne les composantes connexes en de grandes zones pour obtenir les blocs.

Gatos [33] présente un ensemble d'algorithmes intégrés pour la segmentation de pages de journaux numérisés ainsi que l'identification des articles. La méthode utilisée est une

méthode mixte. Elle repose sur l'utilisation conjointe d'une technique ascendante (les composantes connexes pour l'extraction de blocs), avec la technique descendante (analyse du fond de l'image).

Un concours de segmentation des documents à structures complexes est instauré au sein de la conférence ICDAR 2001 et ce dernier est devenu une tradition. Trois équipes ont participé à ce concours [38, 61, 66] et une évaluation quantitative a été effectuée par Gatos [32]. Les méthodes proposées par les trois équipes sont détaillées dans les paragraphes suivants.

Lui [61] utilise une méthode ascendante pour la reconnaissance des différentes entités : les filets, les images, les graphiques et les textes. Pour la fusion des lignes de texte en blocs, les composantes connexes voisines sont prises en considération et seulement la paire de composantes connexes la plus valable est choisie pour la fusion. Une fois la fusion effectuée, les composantes connexes de petites tailles sont supprimées, et ensuite intervient l'étiquetage du texte en titre et la séparation graphique image à partir d'un graphe.

Mitchell [66] utilise une méthode ascendante. En effet, l'image est initialement segmentée, ensuite les régions rectangulaires qui contiennent le plus de pixels du premier plan sont regroupées. Les patterns sont constitués à partir de ces régions, ils sont plus grands et moins nombreux que les composantes connexes ; cependant ils garantissent la segmentation de composants séparés par plus que trois pixels. Les caractéristiques utilisées lors de la classification de l'entité sont : la taille, la forme et la plage de valeurs de pixels. Enfin, les patterns sont regroupés pour former les lignes et les blocs.

Pour ce concours, nous avons proposé une technique de segmentation de pages de journaux basée sur le découpage et la fusion de zones (split and merge) [38]. C'est une méthode mixte utilisant le principe de découpage et de fusion. En effet, les étapes suivantes sont réalisées :

- extraction de l'image,
- extraction des filets horizontaux et verticaux,
- découpage de l'image du journal en de petites zones à partir des filets verticaux et horizontaux extraits et fusion de ces petites zones pour former des régions plus grandes,
- extraction des lignes de texte,
- étiquetage des blocs en zones de texte et en zones de titre.

2.1.6 Techniques d'apprentissage

Les méthodes automatiques de reconnaissance de structures physiques de documents possèdent des limites. Les systèmes actuels font intervenir l'utilisateur pour corriger ou valider les résultats de reconnaissance. En effet, ces systèmes renferment des méthodes guidées par des modèles. La construction de ces modèles peut se faire soit manuellement, soit par apprentissage. La méthode manuelle est fastidieuse et généralement c'est la

méthode avec apprentissage qui est privilégiée afin de faire éviter à l'utilisateur de corriger toujours les mêmes erreurs.

L'adjonction des techniques d'apprentissage aux méthodes de reconnaissance des structures permet d'améliorer les taux de reconnaissance. Ces taux peuvent être constamment améliorés si les méthodes de reconnaissance incluent un apprentissage incrémental.

Les techniques d'apprentissage nécessitent généralement beaucoup de données de référence communément appelées fonds de vérité. Celles-ci incluent des modèles à base d'arbres de grammaires, des modèles à bases de règles, des modèles stochastiques, des méthodes des patterns et des méthodes à base de réseaux de neurones artificiels. Dans les sous-sections suivantes nous passerons en revue ces différents modèles et méthodes.

2.1.6.1 Modèles à base de grammaires d'arbres

Les modèles à base d'arbres de grammaires ont été utilisés pour l'apprentissage. En effet, Akindele [5] propose une méthode basée sur l'inférence d'arbres de grammaires avec une combinaison des constructeurs génériques de la famille ODA (Office Document Architecture) qui est un standard d'échange et de balisage de documents électroniques tel que SGML [44]. La méthode construit une structure physique spécifique pour chaque échantillon et invite l'utilisateur pour assigner les étiquettes logiques aux composants. A partir de cette structure logique étiquetée, le modèle générique de la classe en cours de traitement est généré et modifié en utilisant une méthode d'inférence d'arbres de grammaires.

2.1.6.2 Modèles à base de règles

Les modèles à base de règles s'inspirent des méthodes issues de l'intelligence artificielle. Malerba [63] décrit le système WISDOM qui comprend un modèle à base de règles. En effet, WISDOM génère des échantillons d'apprentissage décrivant comment l'utilisateur a corrigé les erreurs de segmentation. Les actions de correction que l'utilisateur peut effectuer sont : le découpage horizontal et vertical, et la fusion. Ces échantillons d'apprentissage sont regroupés sous forme de règles qui permettent au système d'inférer une règle.

2.1.6.3 Modèles stochastiques

Brugger [23] propose un modèle stochastique de n-grams généralisés. Ce modèle permet en combinant les méthodes structurelles avec des données statistiques, et au moyen d'une heuristique, de déterminer l'étiquetage qui donne la meilleure structure arborescente. Ce modèle est pourvu d'une méthode d'apprentissage incrémentale. Une fois que l'utilisateur valide les résultats, le système met à jour les fréquences relatives des bi-grams et tri-

grams. Cette technique à base de modèle stochastique a été surtout utilisée pour la reconnaissance de structures logiques.

2.1.6.4 Méthode des patterns

La méthode des patterns repose sur la comparaison de configurations d'entités avec des patterns. Cette méthode a été proposée par Robadey [83]. Le principe repose sur le fait que chaque entité physique est caractérisée à l'aide d'une configuration comprenant des propriétés physiques (taille, typographie) et aussi de ses voisins. Chaque classe est représentée par un sélecteur de caractéristiques pertinentes et un ensemble de patterns. La reconnaissance est effectuée en comparant la configuration d'une entité avec les patterns de chaque classe. La méthode des patterns est pourvue d'un apprentissage incrémental et a été appliquée aux documents à structures complexes pour la reconnaissance de structures physiques et logiques.

2.1.6.5 Méthode à base des réseaux de neurones artificiels

La méthode à base des réseaux de neurones artificiels fait intervenir la faculté d'apprentissage des réseaux de neurones artificiels pour la reconnaissance de structures.
Les méthodes de reconnaissance de structures physiques reposent essentiellement sur les méthodes descendantes, ascendantes et mixtes. Cependant, à part celles-ci il y a une autre méthode qui est issue de la reconnaissance d'image et elle repose sur la classification de pixels. La reconnaissance de structures physiques par la méthode de classification de pixels est composée de deux étapes : la classification de pixels et l'extraction de régions.

Les réseaux de neurones artificiels (RNA) ont été utilisés aussi bien dans l'étape de classification de pixels que dans l'étape de l'extraction de régions. En effet, Jain [50, 53] utilise les RNA pour la classification de pixels alors que Strothopoulos [90], Cesarini [26] et Andersen [7] utilisent les RNA pour l'extraction de régions.

a) Classification de pixels

La méthode de segmentation développée par Jain [50] est basée sur l'analyse de textures pour la classification de pixels. L'apprentissage est effectué par un réseau de neurones artificiels composés de trois couches et possède en entrée les valeurs d'un masque de pixels et en sortie la classification selon ces trois classes : le texte, le graphique et l'image de fond. Cette méthode a l'inconvénient d'être très gourmande en temps de calcul.

Jain [53] a amélioré sa méthode décrite précédemment. En effet, le nombre de classes à reconnaître a augmenté. Deux classes ont été ajoutées à savoir l'image bitonale et les filets. La classification s'opère à travers l'entraînement d'un réseau de neurones artificiels par un ensemble de masques pour la distinction de trois principales classes de texture dans la segmentation à savoir :

- les images bitonales,

- le fond de l'image,
- le texte et les filets.

Le paradigme du RNA utilisé est le Perceptron multicouches. Seulement un sous-ensemble de pixels de la fenêtre d'entrée sont utilisés en entrée au RNA pour réduire le nombre de connections et améliorer ainsi la faculté de généralisation. Après l'étiquetage des pixels, les régions sont extraites à travers un lissage de l'image. Cette méthode permet de gérer des documents avec plusieurs langues. En revanche l'évaluation n'a été effectuée qu'avec des documents en langue anglaise et chinoise. Malgré les améliorations introduites, cette méthode demeure très gourmande en temps de calcul.

b) Extraction de régions

La méthode d'extraction de régions proposée par Strouthopoulos [90] repose sur l'appariement de contenu d'un type de document mixte en deux zones : texte et non texte. Chaque région, extraite par l'algorithme à base de RLSA, est identifiée par un réseau de neurones artificiels auto organisé avec des caractéristiques locales. Ce réseau de neurones auto organisé permet de réduire la dimension des données pour la technique de visualisation intitulée : cartes auto organisées (SOM). Plusieurs masques sont utilisés pour extraire les patterns textuels de la zone. Les caractéristiques utilisées sont les occurrences dans la zone de l'ensemble donné de masques et d'autres valeurs qui expriment les relations entre ces masques. En revanche, quand le calcul des caractéristiques locales est coûteux, un nombre restreint de blocs à l'intérieur de chaque zone peut être analysé. La sortie du RNA est la classification de la zone en texte ou non texte.

Cesarini [26] se base sur l'algorithme de découpage récursif modifié (MXY). L'arbre X-Y modifié est construit sur la base des régions extraites par un logiciel commercial de reconnaissance de caractères (OCR). La technique de reconnaissance de caractères (OCR) permet, comme son nom l'indique, de reconnaître les caractères d'un document imprimé ou manuscrit. Les logiciels d'OCR peuvent reconnaître les caractères écrits dans une multitude de polices de caractères, en revanche des problèmes persistent pour la reconnaissance de caractères manuscrits. Un vecteur de caractéristiques de taille fixe est introduit au RNA entraîné, pour la classification des images de documents. Chaque caractéristique de ce vecteur décrit les occurrences de certaines entités dans l'arbre correspondant au document. Le RNA utilisé est un Perceptron multi-couches.

Andernsen [7] utilise la méthode d'extraction de régions. La méthode de segmentation utilise une version modifiée de l'algorithme de découpage récursif X-Y. Un réseau de neurones artificiels est utilisé pour classer les régions sur-segmentées. En effet, les neurones d'entrée du RNA sont constitués des caractéristiques extraites des régions sur-segmentées. Parmi les caractéristiques choisies nous trouvons : la surface, la densité, l'hauteur, le nombre de grandes composantes connexes, la superposition,… Plusieurs anciens journaux ont été utilisés dans le protocole d'apprentissage et d'évaluation, et les résultats affichés ne montrent que la mesure de performance pour la séparation des régions textes et non textes.

Dans la section suivante nous verrons des travaux de recherche concernant la construction de fonds de vérité et les mesures de performances des algorithmes de reconnaissance de structures physiques.

2.1.7 Fonds de vérité et mesure de performances

Devant la multitude, la diversité de documents et le nombre de méthodes de reconnaissance de structures physiques proposées dans la littérature une mesure de performances s'impose. La préparation de ces fonds de vérité est cruciale pour pouvoir effectuer cette mesure de performances Or la préparation des ce fonds de vérité nécessite un temps conséquent. Plusieurs travaux de recherche ont étudié la construction des fonds de vérité [10, 80] et d'autres ont étudié la mesure de performances des algorithmes de reconnaissance de structures physiques [9, 32, 60, 97, 98].

2.1.7.1 Fonds de vérité

La construction d'un fonds de vérité est une tache ardue, coûteuse et qui ne peut être automatisée [80]. Antonacopoulos [10] décrit un outil qui génère des fonds de vérité pour la mesure de performance des algorithmes d'extraction de structures physiques. Le système de fonds de vérité développé permet l'édition des résultats de segmentation en permettant les opérations suivantes : la fusion, la séparation et l'altération de la forme. Il permet aussi la spécification du type et de la fonction de chaque région afin de permettre l'évaluation de la classification de la page de document et l'étiquetage logique. Une région peut être un bloc de texte ou un bloc graphique (une image, un graphique et des filets). L'algorithme de segmentation de page de document utilisé est basé sur l'analyse des zones blanches. Cet algorithme sur-segmente les régions.

2.1.7.2 Mesure de performances

Dans l'analyse de performance des algorithmes de segmentation, la première approche est basée sur la comparaison des caractères fonds de vérité avec le résultat d'un OCR appliqué à l'image segmentée du document.
Il est à noter que les méthodes qui effectuent des comparaisons de régions peuvent être divisées en deux catégories ; celles basées sur une description au niveau pixel et celles basées sur la géométrie. La première est lente et nécessite deux instances réduites de l'image alors que la deuxième utilise le rectangle comme figure géométrique pour la comparaison.

Cependant Antonacopoulos [9] utilise le polygone comme approche de représentation des données de segmentation. Cette méthode utilise le système d'adressage tesseral pour représenter les régions par des polygones. Dans la représentation d'adressage tesseral les deux coordonnées X et Y sont compressées dans un seul entier. Si l'entier est codifié sur 32 bits alors les premiers 16 bits significatifs sont alloués à X et le reste des 16 bits

restants sont alloués à Y. L'outil développé entre dans le contexte de la mesure de performance des algorithmes de segmentation.

Liang [60] présente une mesure de performance des algorithmes de reconnaissance de structures physiques. Cette mesure permet, entre autres, de mesurer le taux de correspondance entre les entités détectées et ceux figurant au sein du fonds de vérité.
Un formalisme mathématique est décrit pour permettre cette mesure.

Gatos [32] présente la méthode d'évaluation utilisée dans le cadre du premier concours de segmentation d'images de documents. La méthode de mesure des performances est basée sur le nombre d'occurrences de similarités entre les entités extraites par les algorithmes, présentés par les équipes participant à ce concours, et celles du fonds de vérité. Les entités à extraire sont :

- les textes,
- les titres,
- les titres en vidéo inverse,
- les filets horizontaux et verticaux,
- les images et les graphiques.

Afin de classer les concurrents une mesure de distance a été utilisée. Celle-ci combine les valeurs moyennes du taux de détection et la valeur de précision de la reconnaissance. Les documents utilisés lors de ce concours sont des images de documents scannées issues d'anciennes unes de journaux en Grec et en Anglais.

Wang [98] présente une version améliorée de la classification de zones et de la mesure de performances. Une version optimisée de l'arbre de décision binaire est utilisée pour l'estimation de la probabilité maximale de la classe du contenu de la zone, pour un ensemble donné. L'algorithme de Viterbi est utilisé pour la recherche d'une solution optimale pour une zone de séquence dans un autre ensemble.

Un autre travail de recherche mené par Wang [97] mesure les performances mais cette fois-ci les zones sont représentées sous forme d’un vecteur de caractéristiques. Le nombre de caractéristiques utilisées est de vingt-cinq. Parmi ces caractéristiques nous retrouvons la moyenne et la variance de la plage de valeurs. En effet, pour chaque zone, la plage de valeurs et les caractéristiques spatiales sont calculées pour chaque ligne dans deux différentes directions : l’horizontale et la diagonale. Un modèle probabiliste est utilisé pour la classification de chaque zone en se basant sur le vecteur de caractéristiques. Dans le processus de classification un arbre de décision est utilisé. Les différentes classes de zones sont au nombre de neuf :

- texte avec une taille inférieure à dix-huit points,
- texte avec une taille supérieure à dix-neuf points,
- math,
- table,
- image bitonale,
- cartes,
- filets,
- logo,

- autres.

L'ensemble d'entraînement et de test est composé d'images issues du fonds de vérité du cédérom de l'université de Washington [79, 80]. Il est à noter que ce fonds de vérité n'utilise que des documents à structures simples.

2.2 Reconnaissance de structures logiques

Une fois l'étape de la reconnaissance de structures physiques effectuée, l'étape qui s'ensuit est l'étape de la reconnaissance de structures logiques. Les travaux de recherche dans le domaine de la reconnaissance de structures logiques sont moins généraux que ceux pour la reconnaissance de structures physiques. En effet, la structure logique est fortement dépendante de l'application à traiter.

La reconnaissance de structures logiques repose essentiellement sur l'élaboration d'une correspondance entre les entités physiques, extraites lors de l'étape de l'extraction de structures physiques d'un document, et un ensemble d'entités logiques qui expriment généralement la sémantique du document. Cette mise en correspondance entre entités physiques et logiques est communément appelée étiquetage logique. Celle-ci consiste en l'attribution d'une étiquette à un bloc physique. C'est l'application à traiter qui détermine le type et le nombre d'étiquettes. Par exemple pour le cas d'une table de matières d'actes de conférence, ces étiquettes correspondent par exemple, aux sections, titres, et noms d'auteurs.

Plusieurs techniques ont été utilisées pour l'extraction de structures logiques. Il y a des méthodes d'extraction de structures logiques sont à base de règles.

Ingold [48] utilise une méthode d'analyse descendante pour la reconnaissance de structures logique d'un document. Pour chaque classe de documents, une description formelle, comprenant des règles de composition et des règles de présentation, est effectuée. A partir de la description de documents, une série d'automates est construite qui permet d'effectuer l'analyse. L'expérimentation a été effectuée sur les textes juridiques.

Niyogi [72] présente un système nommé DeLoS à base de règles pour l'extraction de structures logiques. Cette approche repose sur l'utilisation des heuristiques appliquées sur la structure physique pour inférer les classes et les étiquettes des blocs dans une image de documents, et sur la combinaison de ces blocs pour la création d'unités logiques. Le nombre de règles utilisées par DeLoS s'élève à 160 règles. DeLoS permet aussi de restituer l'ordre de lecture. La classe de documents utilisée par DeLoS se limite aux journaux.

Tsujimoto [93] utilise une méthode basée sur la transformation de l'arbre de structures géométriques pour la reconnaissance de structures logiques du document. Cette méthode

permet le déplacement de nœuds au sein de l'arbre et l'annotation de chaque nœud avec l'étiquette appropriée tout en se référant à un ensemble spécifiques de règles.

Certaines méthodes d'extraction de structures logiques reposent sur les modèles stochastiques [23] et sur les modèles syntaxiques [49]

La méthode proposée par Brugger [23] repose sur un modèle stochastique de n-grams généralisés. L'étiquetage est effectué par le biais d'une heuristique, obtenue en combinant les méthodes structurelles avec des données statistiques.
Hu [49] utilise un modèle décrit par une grammaire. Des règles de production de la grammaire hors-contexte sont utilisées pour représenter la structure du document logique. Le cheminement des étapes du processus est dicté par un algorithme de programmation dynamique. L'auteur a recours à la logique floue pour résoudre l'incertitude.

D'autres méthodes sont dotées d'apprentissage. En effet, plusieurs techniques d'apprentissage ont été utilisées : les réseaux Bayesien [87], les réseaux de neurones artificiels flous [74], la mise en correspondance de graphes [59], la méthode des patterns [83] (voir section 2.1.6.4) et le modèle stochastique de n-grams généralisés [23] (voir section 2.1.6.3).

Souafi [87] propose une modèle probabiliste représenté par les réseaux Bayesien pour l'étiquetage logique de documents. Ce modèle utilise un classifieur de réseaux Bayesien pour représenter les relations entre l'ensemble des attributs et la classe d'étiquetage correspondante. Le modèle décrit utilise un apprentissage supervisé. L'expérimentation a été effectuée sur les tables de matières de plusieurs magazines commerciaux et scientifiques.

Palermo [74] propose un système à base de réseaux de neurones artificiels flous pour l'étiquetage logique d'images de documents. Les caractéristiques utilisées, au sein du RNA flou, sont de deux types : géométrique (la taille, la position, le nombre de mots et de lignes dans un bloc) et contextuel (une seule information contextuelle est introduite pour déterminer si le bloc précédent a été étiqueté ou pas). L'ensemble de documents utilisé pour l'apprentissage et le test du RNA flou repose sur des lettres commerciales et des articles scientifiques.

Liang [59] propose un système doté d'un apprentissage adaptatif de modèles de classes de documents. Le système développé utilise un modèle pour l'étiquetage logique des images de documents en utilisant la technique de mise en correspondance de graphes et, améliore d'une manière adaptative ce modèle avec la rétroaction des erreurs. Le graphe utilisé est entièrement connecté, chaque nœud correspond à un bloc segmenté d'une page de documents. Les attributs de chaque nœud sont la position, la taille du rectangle englobant et la taille de la fonte de caractères. Les arrêtes entre une paire de nœuds reflètent la relation spatiale entre deux blocs. Un logiciel de reconnaissance optique de caractères est utilisé à la suite de la reconnaissance de structures physiques. Les documents utilisés dans l'expérimentation sont des articles extraits d'actes de conférences.

2.3 Reconnaissance optique de caractères de documents arabes

D'après la recherche que nous avons effectuée, nous avons trouvé qu'un seul travail de recherche relatif à la reconnaissance de structures physiques de documents en langue arabe celui de Haoula [42]. En revanche, la grande attention a été portée pour la reconnaissance optique de l'écriture arabe aussi bien imprimée que manuscrite.

Dans cette section, nous passerons en revue l'unique travail de recherche que nous avons trouvé et certains travaux relatifs à la reconnaissance optique de l'écriture arabe. La segmentation dans ces travaux, se limite généralement à la segmentation en lignes de texte et à la séparation des pseudo-mots.

Haoula [42] décrit les différentes méthodes utilisées pour la transformation du document de la forme papier vers sa forme logique. L'expérimentation a été effectuée sur le journal officiel de la république Tunisienne en langue arabe ; qui est un journal à deux colonnes dont la structure de page est similaire à celle des articles scientifiques. Les différentes méthodes se résument ainsi ; dans une première phase, l'extraction de structures physiques est effectuée. Celle-ci est une méthode ascendante qui se base sur le regroupement de composantes connexes. L'extraction effectuée permet d'avoir une segmentation en mots, en lignes et en blocs. La deuxième phase traite de la reconnaissance des fontes arabes. En revanche la troisième phase, traite de la reconnaissance de structures logiques. Il est à noter qu'aucune évaluation n'est donnée pour la reconnaissance de structures physiques.

Amin [6] présente l'état de l'art pour la reconnaissance optique du caractère arabe d'une manière hors-ligne. Deux approches de segmentation en mots sont présentées, la segmentation implicite et la segmentation explicite. Ensuite, les méthodes utilisant ces deux approches de segmentation sont détaillées. Les méthodes décrites utilisent soit les modèles de Markov cachés, soit les réseaux de neurones artificiels. Dans la première à savoir la segmentation implicite, les mots sont segmentés directement en lettres, par contre pour la segmentation explicite, les mots sont d'abord segmentés en pseudo-caractères puis reconnus individuellement.

Pechwitz [77] utilise une approche basée sur les modèles de Markov cachés pour la reconnaissance de mots arabes manuscrits en utilisant la base de données IFN/ENIT[4]. La segmentation en mots a été effectuée en utilisant l'analyse en composantes connexes et l'extraction de la représentation du contour de l'image. L'approche a été appliquée sur des images de documents contenants le nom de villes et villages de la Tunisie.

Ben Amara [20] dresse les problèmes actuels rencontrés en reconnaissance de l'écriture arabe mais aussi les perspectives de développement. Elle décrit brièvement les méthodes utilisées pour la segmentation en lignes de texte, en pseudo-mots et en caractères. Pour la

[4] Base de données de mots arabes manuscrits. URL: http://www.ifnenit.com

segmentation en lignes de texte et en pseudo-mots, c'est le profil de projection vertical qui est utilisé. En revanche la segmentation en caractères est difficile et non encore résolu.

2.4 Conclusion

Malgré les avancées effectuées par la reconnaissance de documents, certains problèmes restent encore ouverts comme par exemple la reconnaissance de documents à structures complexes. Néanmoins, les méthodes de reconnaissance de structures physiques pour les documents à structures simples donnent de bons résultats et ces méthodes sont arrivées à maturité. En revanche, dans l'optique de rendre les méthodes de reconnaissance de structures physiques pour les documents à structures complexes plus performantes, plusieurs travaux de recherche s'orientent pour traiter ce genre de documents.

Ces méthodes sont dépendantes et sont sensibles au moindre changement de la classe de documents. En effet, il suffit d'une modification mineure au niveau de la classe de documents pour que les performances chutent d'une manière drastique. D'une manière générale la méthode de reconnaissance de structures physiques doit être adaptative en présence soit d'une modification de la même classe de documents, soit d'une autre classe de documents. Néanmoins, la construction manuelle des fonds de vérité de ces classes de documents s'avère une tâche longue et ardue. Afin de palier la difficulté de la création de ces fonds de vérité, l'adjonction de l'interactivité et de l'apprentissage au sein de la méthode s'avère indispensable.

D'après notre étude sur l'état de l'art des méthodes de reconnaissance de structures physiques, nous avons trouvé qu'un seul travail traitant les documents en langue arabe à structures simples. Par contre, à notre connaissance nous n'avons pas eu à faire à des travaux de recherche décrivant des méthodes de reconnaissance de structures physiques de documents en langue arabe riches en structures et en variabilité. Ainsi, dans notre thèse nous avons conçu deux systèmes de reconnaissance de structures physiques et logiques de documents riches en structures et en variabilité en langue arabe. Ces systèmes sont dotés d'un apprentissage évolutif.

Chapitre 3

Format de représentation des résultats intermédiaires.

A l'heure actuelle, il existe un nombre élevé de formats de fichiers. Or devant cette multitude de formats, la conversion entre ces formats de fichier devient une tâche ardue. Afin de faciliter cette tâche, nous devons utiliser le plus possible des formats standardisés mais aussi des formats permettant une plus grande flexibilité. De ce fait, le bon choix du format de représentation des résultats de reconnaissance est important.

En reconnaissance de documents, le bon choix du format de représentations des résultats de reconnaissance est essentiel pour permettre la constitution des fonds de vérité. Si en plus ce choix s'oriente vers un format ouvert, alors l'échange des résultats de reconnaissance entre les chercheurs sera facilité.

Pour représenter les résultats intermédiaires, notre choix s'est porté sur XML qui est considéré comme un format standard et ouvert. Ce choix est motivé par la facilité d'échange des résultats et de réutilisabilité de ce format. Dans les sous-sections suivantes, nous présentons l'architecture globale, les différents formats de représentation des résultats de reconnaissance, puis nous décrivons XML, ses avantages par rapport à d'autres formats et la façon dont nous avons appliqué XML pour la représentation des résultats intermédiaires. Ensuite, nous décrivons les formats de représentation de données du réseau de neurones artificiels utilisés pour l'apprentissage de structures physiques et logiques.

3.1 Architecture globale

L'architecture globale de la thèse repose sur un système de reconnaissance de structures physiques des documents arabes à structures complexes, un système de reconnaissance de structures physiques de toutes les classes de documents doté d'apprentissage évolutif : *PLANET* et un système de reconnaissance de structures logiques de toutes les classes de documents doté d'apprentissage évolutif : *LUNET*. Ces trois systèmes utilisent *xmillum* qui a été développé dans notre groupe de recherche DIVA par Oliver Hitz.
xmillum est un "framework" pour créer des applications assistées de reconnaissance de documents [45, 47] et il travaille avec des données au format XML. *xmillum* ne vise pas à être une application à part entière, mais plutôt à être un cadre pour créer rapidement des petites applications qui résolvent des tâches très spécifiques de la reconnaissance de document. *xmillum* permet l'utilisation des transformations XSLT.

La figure 3.1 illustre le schéma global de l'architecture. Dans cette figure, nous constatons que les résultats du système de reconnaissance de structures physiques, de *PLANET* et de *LUNET* sont des fichiers XML. Chaque fichier est défini dans un contexte approprié :

1. Le système de reconnaissance s'applique sur des images de documents de toutes les classes de documents et génère un fichier XML représentant le résultat de la reconnaissance de structures physiques. Ce fichier comprend le résultat de l'extraction des filets, l'extraction des cadres, la séparation texte image, l'extraction des lignes de texte, et la fusion des lignes de texte en blocs,
2. Le système de reconnaissance des structures physiques doté d'apprentissage évolutif : *PLANET*, permet d'améliorer le taux de reconnaissance obtenu par le système de reconnaissance de structures physiques, en lui ajoutant l'apprentissage des classes de documents. Le fichier résultat de *PLANET* en XML représente la reconnaissance de structures physiques par les réseaux de neurones artificiels.
3. Le système de reconnaissance des structures logiques doté d'apprentissage évolutif : *LUNET*, permet entre autres la validation de l'architecture proposée par *PLANET*. *LUNET* accepte en entrée le fichier de reconnaissance de *PLANET* et génère en sortie un fichier XML représentant le résultat de reconnaissance de structures logiques par les réseaux de neurones artificiels.

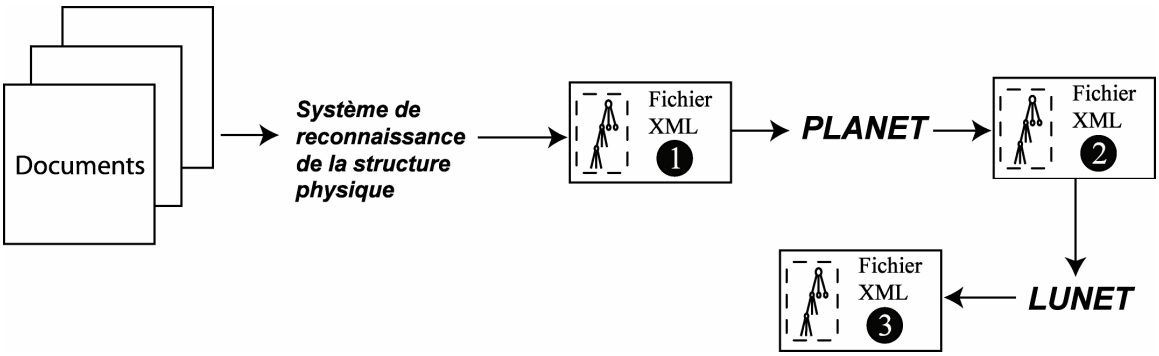


Figure 3.1 : Architecture globale

3.1.1 PLANET

PLANET, notre système de reconnaissance de structures physiques de toutes les classes de documents, basé sur les réseaux de neurones artificiels (voir chapitre 5) repose sur la correction interactive des résultats de reconnaissance et sur l'apprentissage. Le flux d'information véhiculé par notre système de reconnaissance de structures physiques est illustré dans la figure 3.2.

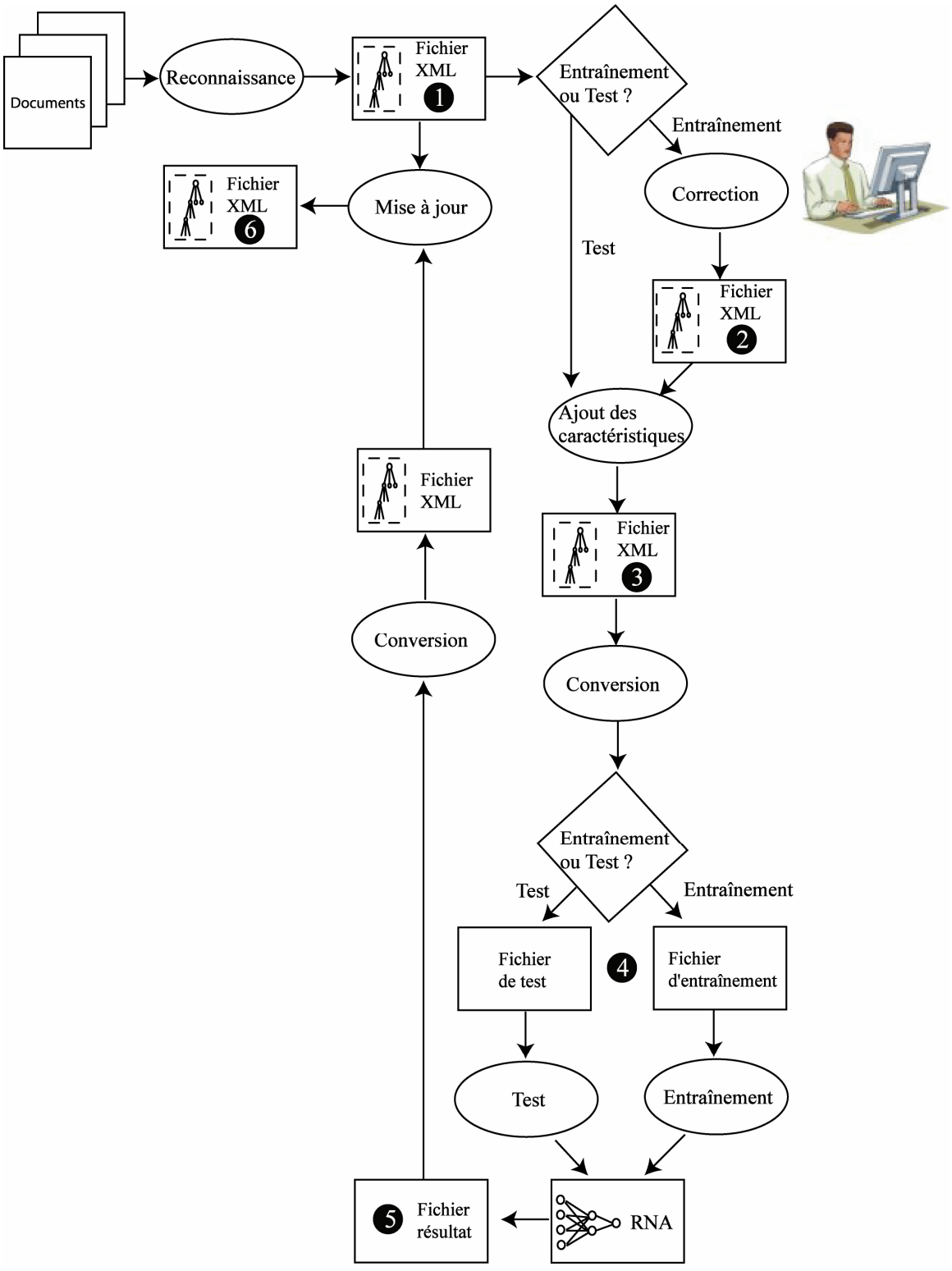


Figure 3.2 : Flux d'information de *PLANET*

La figure 3.1 montre qu'il existe plusieurs fichiers représentant des résultats intermédiaires :

1- Le résultat de la reconnaissance de structures physiques des documents complexes en langue arabe est représenté dans un fichier XML. Ensuite, l'utilisateur chercheur choisit entre l'entraînement et le test du réseau de neurones artificiels. Si son choix s'oriente vers l'entraînement alors c'est l'étape de correction interactive des résultats de reconnaissance qui est exécutée (2), sinon c'est l'étape de test qui est choisie. Dans les deux cas, une étape d'ajout de caractéristiques est nécessaire (3),
2- L'utilisateur chercheur a recours à *xmillum* avec des plugins permettant la fusion et le découpage, que nous avons développés, pour la correction des résultats de la reconnaissance. Ces plugins génèrent en sortie un fichier XML contenant la liste des actions de corrections faites par l'utilisateur chercheur,
3- Des caractéristiques sont ajoutées soit aux actions de corrections faites par l'utilisateur chercheur s'il s'agit d'un entraînement, soit aux entités s'il s'agit d'un test. Ces caractéristiques sont utiles aussi bien pour la construction du fichier d'entraînement que de test du réseau de neurones artificiels. Le fichier des caractéristiques obtenu est un fichier XML,
4- Le simulateur de réseaux de neurones que nous avons choisi possède un format texte propriétaire pour le fichier de test et le fichier d'entraînement. Le fichier des caractéristiques est converti selon le choix de l'utilisateur chercheur soit en fichier test, soit en fichier d'entraînement,
5- Le résultat de la reconnaissance du réseau de neurones artificiels obtenu est un fichier dans un format propriétaire qui est converti en XML,
6- Finalement, le fichier résultat de reconnaissance (1) est mis à jour avec le fichier des résultats obtenus par le réseau de neurones artificiels.

Tous ces formats de fichiers sont détaillés dans les sections 3.4.1 et 3.5.

3.1.2 LUNET

LUNET, le système de reconnaissance de structures logiques de toutes les classes de documents, basée sur les réseaux de neurones artificiels (voir chapitre 6) repose sur une architecture semblable à celle de *PLANET*. *LUNET* a été conçu entre autres pour valider l'architecture de *PLANET* en héritant de celle-ci l'interaction et l'apprentissage. Le flux d'information véhiculé par le système de reconnaissance de structures logiques, et de l'apprentissage est illustré dans la figure 3.3.

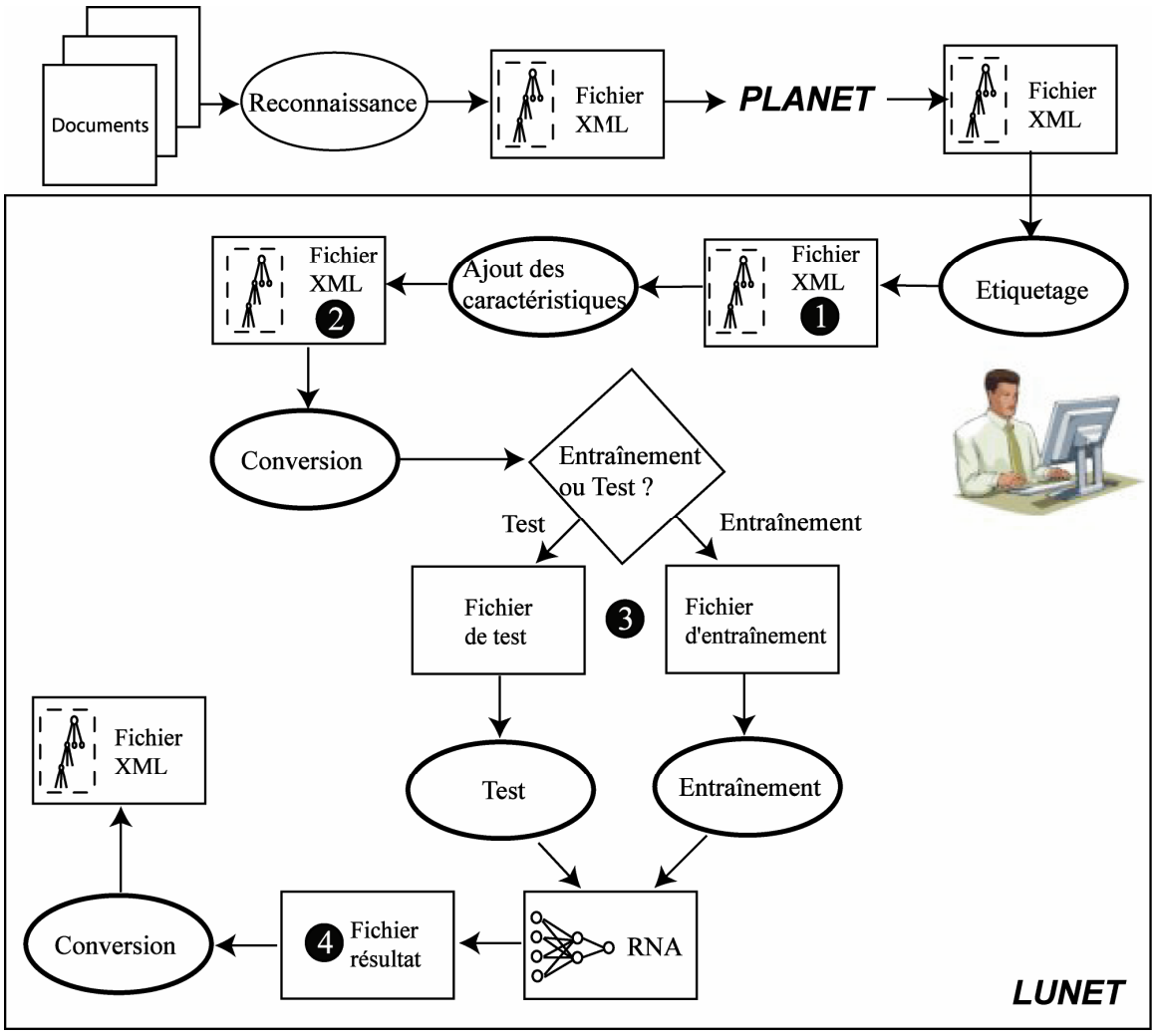


Figure 3.3 : Flux d'information de *LUNET*

La figure 3.3 montre qu'il existe un certain nombre de fichiers de résultat intermédiaires :

1- L'utilisateur chercheur utilise *xmillum* avec une feuille de style spécifique, que nous avons développée, pour l'étiquetage des blocs de texte. Ces blocs sont extraits du fichier de reconnaissance de structures physiques de *PLANET*. L'étiquetage aboutit à un fichier XML contenant la liste des blocs de texte étiquetés,
2- Un ensemble de caractéristiques est ajouté à chaque bloc de texte étiqueté. Le fichier obtenu est un fichier XML,
3- Le fichier des caractéristiques est converti selon le choix de l'utilisateur chercheur soit en tant que fichier test, soit en tant que fichier d'entraînement,
4- Le résultat de la reconnaissance de structures logiques du réseau de neurones artificiels est un fichier. Ce dernier est converti en XML,

Tous ces formats de fichiers sont détaillés dans les sections 3.4.2 et 3.5.

3.2 Les différents formats de représentation des résultats de reconnaissance

Parmi les différents formats de représentation des résultats de reconnaissance nous citons DAFS (Document Attribute Format Specification) [81]. Ce dernier a été construit en vue d'un format d'échange de résultats de reconnaissance d'images de documents et il a été le plus utilisé avant le développement de XML. DAFS permet : la subdivision du document en des entités bien définies, la délimitation et l'étiquetage. Le laboratoire "Intelligent Systems Laboratory" de l'université de Washington à Seattle a adopté DAFS comme format de représentation pour le fonds de vérité de la base de données d'images de documents. Bapst [15] pour sa part a utilisé DAFS pour la gestion des données dans le cadre du projet CIDRE. Liang [60] et Wang [97] ont utilisé ce format pour la représentation des leurs résultats de reconnaissance.

A part DAFS, plusieurs formats moins bien connus ont été utilisés comme format de représentation des résultats de segmentation. Comme par exemple le format qui a été utilisé dans le concours de segmentation de journaux à ICDAR en 2001 [32]. Ce format a été utilisé pour décrire les fonds de vérité des unes de journaux ainsi que les résultats de segmentation des participants à ce concours.

3.3 XML

XML (eXtensible Markup Language) a été développé en 1996 par un groupe de travail sous les auspices du consortium W3C. L'idée de base, lors du développement d'XML, était de fournir la puissance et la flexibilité de SGML [44]. Néanmoins, le but essentiel de XML est de spécifier un sous-ensemble de SGML tout en éliminant la lourdeur de ce dernier.

XML est un format non propriétaire, donc public. La spécification de la première version a été acceptée par le W3C en 1998 alors que la dernière version en date est la 1.1 et celle-ci est apparue en 2004. XML a été adopté par la grande majorité des éditeurs de logiciels Microsoft, IBM, Oracle, ... dans leurs produits pour représenter leurs formats propriétaires dans un format universellement reconnu.

De nos jours, XML est supporté par presque tous les navigateurs Web (Internet Explorer, Netscape, Mozilla et Opera). Dans la majorité des langages de programmation (Java, C#, C++, Visual Basic, …), il existe des APIs permettant d'analyser (DOM, SAX, JDOM, JAXP) et de générer des fichiers XML.

Le langage XML propose diverses fonctionnalités dont :

- la DTD (Document Type Definition) qui représente l'ensemble des règles et des propriétés que doit suivre un document XML. Ces règles définissent généralement

le nom et le contenu de chaque balise et le contexte dans lequel elles doivent exister,
- le XML Schemas qui permet d'enrichir et de typer la description de la structure et du contenu d'un document XML,
- le langage de transformation de données tel que XSLT (eXtensible Stylesheet Language Transformations),
- le langage d'interrogation de bases de données tel que XQuery,
- le langage XPath qui permet de localiser avec précision une partie donnée d'un document XML,
- le langage XLink qui permet la mise en place des liens vers les contenus de documents.

L'utilisation de XML en tant que format de représentation des résultats de reconnaissance permet facilement d'échanger ces résultats et favorise la création des fonds de vérité.

Un fichier XML est visualisable sur presque tous les navigateurs Internet et sa modification est possible par le biais d'un éditeur de texte mais en revanche elle n'est pas simple. Des éditeurs XML spécialisés permettent de faciliter la modification des fichiers XML.

3.4 Utilisation de XML

Nous avons utilisé XML dans *PLANET* et dans *LUNET*.

3.4.1 PLANET

Nous avons adopté XML à différents niveaux :
- le format de représentation des résultats de la reconnaissance,
- le format de correction des résultats de reconnaissance,
- le format des caractéristiques des entités.

3.4.1.1 Format de résultat de la reconnaissance

Vu que la reconnaissance de structures physiques de chaque image de document est représentée dans un fichier XML, nous avons écrit une DTD (Document Type Definition) pour la validation des fichiers XML obtenus. Il est à noter que la structuration des résultats de reconnaissance en XML peut être faite de différentes manières. Le listage 3.1 présente la DTD que nous avons définie. Les éléments XML threads, images, texts, frames, blocks représentent respectivement les filets, les images les lignes de texte, les cadres, et les blocs de texte.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT segmentation (Threads, Images, Texts, Frames, Blocks)>
<!ATTLIST segmentation
  image CDATA #REQUIRED
```

```
>
<!ELEMENT Threads (Thread+)>
<!ELEMENT Thread EMPTY>
<!ATTLIST Thread
  x  CDATA #REQUIRED
  y  CDATA #REQUIRED
  w  CDATA #REQUIRED
  h  CDATA #REQUIRED
>
<!ELEMENT Images (Image+)>
<!ELEMENT Image EMPTY>
<!ATTLIST Image
  x  CDATA #REQUIRED
  y  CDATA #REQUIRED
  w  CDATA #REQUIRED
  h  CDATA #REQUIRED
>
<!ELEMENT Texts (Text+)>
<!ELEMENT Text EMPTY>
<!ATTLIST Text
  x  CDATA #REQUIRED
  y  CDATA #REQUIRED
  w  CDATA #REQUIRED
  h  CDATA #REQUIRED
>
<!ELEMENT Frames (Frame+)>
<!ELEMENT Frame EMPTY>
<!ATTLIST Frame
  x  CDATA #REQUIRED
  y  CDATA #REQUIRED
  w  CDATA #REQUIRED
  h  CDATA #REQUIRED
>
<!ELEMENT Blocks (Block+)>
<!ELEMENT Block EMPTY>
<!ATTLIST Block
  x  CDATA #REQUIRED
  y  CDATA #REQUIRED
  w  CDATA #REQUIRED
  h  CDATA #REQUIRED>
```

Listage 3.1 : DTD validant les résultats de la reconnaissance en XML

Un résultat de reconnaissance en format XML est présenté dans le listage 3.2.

```
<?xml version="1.0" encoding="UTF-8"?>
<segmentation image="AlHayat_01_10_2003.tif">
 <Threads>
  <Thread x="221" y="1048" w="4160" h="5" />
  <Thread x="221" y="1148" w="4160" h="5" />
  <Thread x="2300" y="1184" w="5" h="3951" />
  <Thread x="221" y="3434" w="2054" h="5" />
  <Thread x="2327" y="4018" w="2054" h="5" />
  ...
 </Threads>
 <Images>
  <Image x="1809" y="337" w="1251" h="607" />
  <Image x="2323" y="1854" w="1562" h="2136" />
  <Image x="2358" y="1880" w="1135" h="1105" />
  ...
 </Images>
```

```
<Texts>
 <Text x="1809" y="337" w="1251" h="607" />
 <Text x="3878" y="361" w="501" h="63" />
 <Text x="2130" y="365" w="359" h="52" />
 <Text x="3003" y="427" w="63" h="487" />
 <Text x="922" y="445" w="375" h="367" />
 <Text x="2122" y="446" w="400" h="169" />
 <Text x="1547" y="447" w="237" h="126" />
 ...
</Texts>
<Frames>
 <Frame x="215" y="2617" w="2057" h="782" />
</Frames>
<Blocks>
 <Block x="215" y="2617" w="2057" h="782" />
 <Block x="3878" y="314" w="501" h="114" />
 <Block x="2130" y="365" w="359" h="52" />
 <Block x="229" y="346" w="490" h="90" />
 <Block x="3003" y="427" w="63" h="487" />
 <Block x="2122" y="446" w="400" h="169" />
 ...
</Blocks>
</segmentation>
```

Listage 3.2 : Résultat de reconnaissance en format XML

Pour que l'utilisateur chercheur puisse faire une bonne interprétation des résultats de reconnaissance, il doit disposer d'un outil graphique lui permettant de visualiser conjointement les résultats de la reconnaissance et l'image du document.

Avec *xmillum*, nous avons la possibilité de visualiser les résultats de segmentation par couches. Celles-ci permettent d'isoler les différents types d'entités (les cadres, les filets, les lignes de texte, les images et les blocs de texte) les unes des autres. Une couche supplémentaire est ajoutée en arrière plan permettant la visualisation de l'image du document.

L'affectation des entités de reconnaissance aux couches, se fait à l'intérieur d'une feuille de style XSLT dans laquelle nous spécifions comment les résultats seront affichés. Dans cette feuille de style nous spécifions pour chaque élément XML la forme géométrique, la couleur et la transparence. Par exemple, les éléments blocs issus de la fusion des lignes de texte seront affichés dans des rectangles, avec une couleur en rouge et avec une transparence de 20%. Le listage 3.3 montre un exemplaire de la feuille de style XSLT qui permet de visualiser chaque entité de reconnaissance dans un style bien défini. En effet, nous définissons le style "red-style" en spécifiant la couleur, le degré de transparence et la couleur de remplissage. Ce style est associé à une classe java, représenté par l'objet *xmillum* "red-block", pour la gestion des événements et d'affichage de l'entité bloc de texte.

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"  xmlns:tmp="tmp"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:template match="segmentation">
   <xmi:document>
```

```
        <!-- Style for solid, Red, transparent blocks -->
        <xmi:style name="red-style">
           <param name="foreground" value="red"/>
           <param name="transparency" value="0.2"/>
           <param name="fill" value="true"/>
        </xmi:style>

        <!-- Rectangular red blocks -->
        <xmi:object name="red-block" class="iiuf.xmillum.displayable.Block">
           <param name="style" value="red-style"/>
        </xmi:object>

      .....
       <xmi:object name="image" class="iiuf.xmillum.displayable.Image"/>
         <!-- First layer: the background image -->
         <xmi:layer name="Background_Image">
               <image src="{@image}"/>
         </xmi:layer>
         <!-- Second layer: the different blocks -->
         <xmi:layer name="Blocks">
               <xsl:for-each select="Blocks/Block">
                     <red-block x="{@x}" y="{@y}" w="{@w}" h="{@h}" />
               </xsl:for-each>
         </xmi:layer>
         .....
    </xmi:document>
  </xsl:template>
</xsl:stylesheet>
```

Listage 3.3 : Exemplaire d'une feuille de style XSLT permettant la visualisation des résultats de reconnaissance en XML

La figure 3.4 illustre la visualisation des résultats de reconnaissance sous forme de couches. Dans cette figure l'utilisateur chercheur a décidé de ne visualiser que deux couches : la couche de l'image de fond représentant l'image du document et la couche bloc représentant la fusion des lignes de texte en blocs ; les autres couches sont désactivées.

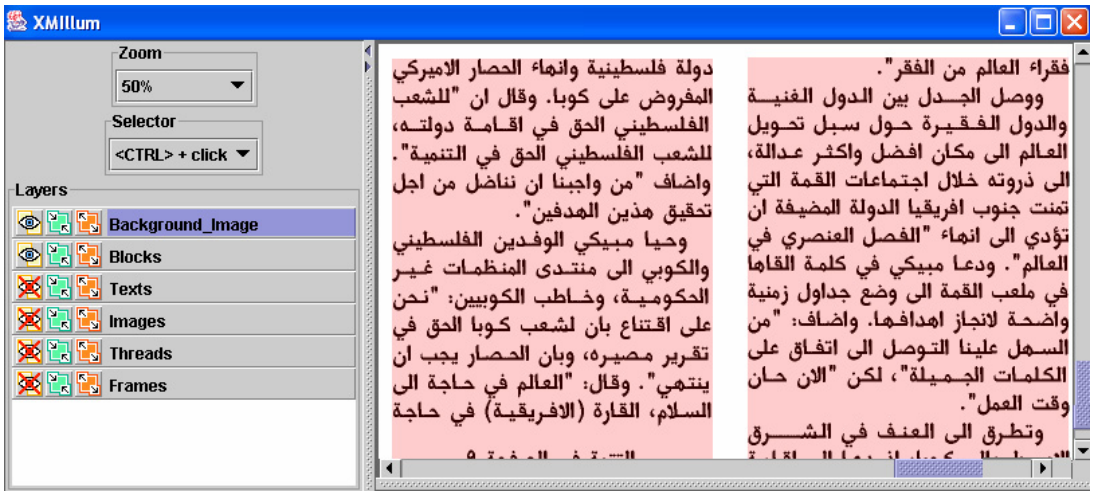


Figure 3.4 : Visualisation des résultats de reconnaissance sous formes de couches avec *xmillum*

3.4.1.2 Format de correction des résultats de reconnaissance

La visualisation des résultats de reconnaissance avec *xmillum* permet de déceler les erreurs de segmentation. Il est important de fournir à l'utilisateur chercheur un outil permettant la correction des erreurs de reconnaissance et ce, dans un environnement interactif. Dans cette optique nous avons développé un module de correction des résultats de reconnaissance.

Le processus de correction est le suivant : l'utilisateur chercheur corrige d'une manière interactive les erreurs de reconnaissance, quand il termine cette tâche, il invoque la sauvegarde et le module de correction se charge de générer la liste d'opérations que l'utilisateur chercheur a effectué dans un fichier XML. Il est à noter, que la correction des erreurs de reconnaissance sert à obtenir :

- des résultats corrects pour la suite de l'analyse,
- des données validées qui peuvent servir pour l'apprentissage

Or, pour permettre cette correction interactive des résultats de segmentation, il faut penser aux opérations que l'utilisateur chercheur peut exécuter. Les types d'erreurs récurrents lors de la reconnaissance de structures physiques d'une image de document sont la sous-segmentation ou la sur-segmentation.

Pour corriger une erreur de sur-segmentation l'utilisateur chercheur doit disposer d'une opération lui permettant de fusionner les entités sur-segmentées. Cette fusion peut s'opérer sur les entités sur-segmentées horizontalement ou verticalement. En revanche pour corriger une erreur de sous-segmentation, l'opération appropriée que l'utilisateur chercheur peut exécuter est le découpage. Ce découpage permet de séparer les entités sous-segmentées. La position de découpage est définie par une ligne séparatrice qui peut être orientée, soit horizontalement ou verticalement. Dans *xmillum* ces deux opérations sont implémentées sous forme de plugins

L'utilisateur chercheur peut donc corriger les résultats de segmentation au moyen de deux différentes opérations : découpage et fusion. La figure 3.5 illustre une capture d'écran, des deux opérations dans notre outil de correction des résultats de reconnaissance.

(a) fusion verticale

(b) découpage horizontal et vertical

Figure 3.5 : Correction des erreurs de reconnaissance : (a) fusion, (b) découpage

La démarche de correction recommandée pour l'utilisateur chercheur c'est de corriger uniquement les erreurs de reconnaissance et de laisser inchangées les entités correctement reconnues. Une fois le processus de correction interactif achevé, l'utilisateur chercheur sélectionne la sauvegarde à partir du menu contextuel et à ce moment un fichier XML est généré contenant les opérations de fusions et de découpages ainsi que les entités correctes. Le fichier XML constitue en quelque sorte le fond de vérité de l'image de document puisqu'il comprend les entités correctes.

Pour permettre à l'utilisateur chercheur la correction interactive des résultats de reconnaissance sous *xmillum*, nous avons adapté une nouvelle feuille de style XSLT qui comprend des références à deux classes java, "`BlockFusion`" et "`BlockSplit`", permettant d'effectuer respectivement la fusion et le découpage de blocs. Au niveau interactivité, l'utilisateur chercheur choisit l'opération à effectuer par le biais d'un menu contextuel

relatif au bloc de texte sélectionné. Ce menu comprend les opérations de fusion, de découpage horizontal et vertical ; il comporte aussi la possibilité de sauvegarder le résultat et de garder l'entité inchangée. Cette feuille de style est présentée dans le listage 3.4.

```
<?xml version="1.0"?>
<!--  Stylesheet  showing  Objects  extracted  from  a  complex  structured
documents -->
<xsl:stylesheet version="1.0"  xmlns:tmp="tmp"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="segmentation">
    <xmi:document>

      <xmi:handler name="merge" class="iiuf.xmillum.handlers.BlockFusion">
      </xmi:handler>
      <xmi:handler name="split" class="iiuf.xmillum.handlers.BlockSplit">
      </xmi:handler>

      <!-- Style for solid, Red, transparent blocks -->
      <xmi:style name="red-style">
          <param name="foreground" value="red"/>
        <param name="transparency" value="0.2"/>
        <param name="fill" value="true"/>
      </xmi:style>

      <!-- Rectangular red blocks -->
      <xmi:object name="red-block" class="iiuf.xmillum.displayable.Block">
        <param name="style" value="red-style"/>
        <param name="press1" value="merge"/>
        <param name="click3" value="split" opt="menu"/>
        <param name="over" value="split" opt="show"/>
      </xmi:object>
     .....
      <!-- Images -->
      <xmi:object name="image" class="iiuf.xmillum.displayable.Image"/>

      <!-- First layer: the background image -->
      <xmi:layer name="Background_Image">
        <image src="{@image}"/>
      </xmi:layer>

      <!-- Second layer: the different blocks -->
      <xmi:layer name="Blocks">
        <xsl:for-each select="Blocks/Block">
          <red-block x="{@x}" y="{@y}" w="{@w}" h="{@h}" />
        </xsl:for-each>
      </xmi:layer>
     .....
     </xmi:document>
  </xsl:template>
</xsl:stylesheet>
```

Listage 3.4 : Exemplaire de la feuille de style XSLT permettant la correction des résultats de reconnaissance en XML

Le listage 3.4 montre les références aux deux classes java "`BlockFusion`" et "`BlockSplit`" et les actions de découpage "`merge`" et fusion "`merge`" associés au clic droit et gauche de la souris définis dans l'objet xmillum "`red-block`"

Le résultat des corrections est représenté en XML sous un format décrit par la DTD du listage 3.5.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Blocks (Merged-blocks, Split-blocks)>
<!ELEMENT Merged-blocks (Merge+)>
<!ELEMENT Merge (Block+, Res)>
<!ATTLIST Merge
  user-action CDATA #REQUIRED
>
<!ELEMENT Res (Block)>
<!ELEMENT Block EMPTY>
<!ATTLIST Block
  x  CDATA #REQUIRED
  y  CDATA #REQUIRED
  w  CDATA #REQUIRED
  h  CDATA #REQUIRED
>
<!ELEMENT Split-blocks (Split+)>
<!ELEMENT Split (Block+)>
<!ATTLIST Split
  user-action CDATA #REQUIRED
>
```

Listage 3.5 : DTD validant les corrections de segmentation : les entités blocks.

Un exemple des résultats de corrections effectuées par l'utilisateur chercheur sur les résultats de reconnaissance pour les entités de type blocs de texte est présenté dans le listage 3.6.

```
<?xml version="1.0" encoding="UTF-8"?>
<Blocks>
  <Merged-blocks>
     <Merge user-action="1">
             <Block h="163" w="571" x="3201" y="2225"/>
             <Block h="202" w="534" x="3800" y="2192"/>
       <Res>
             <Block h="202" w="1133" x="3201" y="2192"/>
       </Res>
  </Merge>
     <Merge user-action="2">
             <Block h="213" w="2136" x="227" y="1322"/>
             <Block h="135" w="385" x="2391" y="1384"/>
      <Res>
             <Block h="213" w="2549" x="227" y="1322"/>
      </Res>
  </Merge>…
   </Merged-blocks>
   <Split-blocks>
     <Split user-action="3">
             <Block h="202" w="1133" x="3201" y="2192"/>
             <Block h="259" w="1421" x="2913" y="2395"/>
     </Split>
     <Split user-action="4">
             <Block h="64" w="926" x="3156" y="2030"/>
             <Block h="202" w="1133" x="3201" y="2192"/>
     </Split>
   </Split-blocks> ……
 </Blocks>
```

Listage 3.6 : Échantillon du résultat de corrections effectuées par l'utilisateur chercheur.

Le listage 3.6 montre que les entités de reconnaissance sont traitées par paires. S'il s'agit d'une fusion, alors nous aurons dans l'élément XML `<Merge>` les deux blocs à fusionner, et le bloc résultat de cette fusion figure dans l'élément fils `<Res>`. En revanche s'il s'agit d'un découpage alors nous aurons dans l'élément XML `<Split>` les deux blocs découpés. Il est à noter que nous avons inclut les entités correctes dans l'élément XML `<Split>` vu qu'il ne s'agit pas d'une fusion. Nous gardons trace de l'ordre des opérations de l'utilisateur chercheur par le biais de l'attribut XML "`user-action`".

3.4.1.3 Format des caractéristiques des entités

Pour permettre au réseau de neurones artificiels d'apprendre les corrections effectuées par l'utilisateur chercheur, nous devons lui fournir les caractéristiques des entités corrigées. Cette étape, qui intervient juste après l'étape de correction des résultats de reconnaissance, permet de préparer soit le fichier d'entraînement, soit le fichier de test des réseaux de neurones artificiels.

Au fichier des résultats de reconnaissance, nous ajoutons, les caractéristiques relatives aux entités blocs de texte. Ces caractéristiques sont calculées et normalisées par un programme écrit en Java. Le choix des caractéristiques est décrit en détail dans le chapitre 5.
Le format des caractéristiques des entités blocs de texte est représenté en XML sous un format décrit pas la DTD du listage 3.5.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Segmentation (Blocks)>
<!ATTLIST Segmentation
  image CDATA #REQUIRED
>
<!ELEMENT Blocks (User-Action+)>
<!ELEMENT User-Action (Block, Block)>
<!ATTLIST User-Action
  num CDATA #REQUIRED
  merge (0 | 1) #REQUIRED
>
<!ELEMENT Block (Density, CC)>
<!ATTLIST Block
  x CDATA #REQUIRED
  y CDATA #REQUIRED
  w CDATA #REQUIRED
  h CDATA #REQUIRED
>
<!ELEMENT Density EMPTY>
<!ATTLIST Density
  white CDATA #REQUIRED
  black CDATA #REQUIRED
>
<!ELEMENT CC EMPTY>
<!ATTLIST CC
  number CDATA #REQUIRED
>
```

Listage 3.7 : DTD validant les caractéristiques des entités blocs de texte en XML

Les éléments XML Density et CC, figurant dans le listage 3.7, représentent respectivement la densité de pixels noirs et blancs, et la densité des composantes connexes pour chaque entité bloc de texte. Un exemple du fichier des caractéristiques pour les entités blocs de texte, est présenté dans le listage 3.8.

```
<?xml version="1.0" encoding="UTF-8"?>
<Segmentation image="Annahar_04_11_2003.tif">
  <Blocks>
     <User-Action num="1" merge="1">
        <Block x="537" y="2100" w="76" h="41">
           <Density white="0.3103337612323492" black="0.6896662387676509"/>
           <CC number="0.00125"/>
        </Block>
        <Block x="204" y="1664" w="741" h="431">
           <Density white="0.1822645136847115" black="0.8177354863152885"/>
           <CC number="0.0795"/>
        </Block>
     </User-Action>
     <User-Action num="2" merge="1">
        <Block x="204" y="1664" w="741" h="477">
           <Density white="0.16742347725465898" black="0.832576522745341"/>
           <CC number="0.08075"/>
        </Block>
        <Block x="204" y="2143" w="740" h="95">
           <Density white="0.18432432432432433" black="0.8156756756756757"/>
           <CC number="0.01925"/>
        </Block>
     </User-Action>
     <User-Action num="3" merge="0">
        <Block x="1271" y="4804" w="469" h="389">
           <Density white="0.1661852324861188" black="0.8338147675138812"/>
           <CC number="0.0395"/>
        </Block>
        <Block x="737" y="3458" w="472" h="1717">
           <Density white="0.13740955351766482" black="0.8625904464823352"/>
           <CC number="0.176"/>
        </Block>
     </User-Action>
  </Blocks>
</Segmentation>
```

Listage 3.8 : Exemple du fichier des caractéristiques des entités blocs de texte.

Après avoir présenté les fichiers intermédiaires de *PLANET* au format XML, nous allons décrire les fichiers de *LUNET*. Ensuite, dans la section 3.5 nous présentons les formats de représentation de données du réseau de neurones artificiels des deux systèmes.

3.4.2 LUNET

Dans cette section nous décrivons les fichiers intermédiaires de *LUNET*. Nous avons adopté XML comme format de représentation de l'étiquetage, et format des caractéristiques des étiquettes.

3.4.2.1 *Format de représentation de l'étiquetage*

L'étiquetage effectué par l'utilisateur chercheur d'une manière interactive génère un fichier XML. L'ensemble d'étiquettes utilisées dans *LUNET* est défini dans une feuille de style XSLT rattachée à *xmillum*. Dans notre cas nous avons choisi les étiquettes suivantes (Titre, Auteur, Texte de base, Ancre vers et Légende), Le listage 3.9 présente la DTD que nous avons définie.

```
<!ELEMENT Labeling (Titres, TextesBase, AncreVers, Legendes, Auteurs)>
<!ATTLIST Labeling
CDATA #REQUIRED
>
<!ELEMENT Titres (Titre+)>
<!ELEMENT Titre EMPTY>
<!ATTLIST Titre
  x CDATA #REQUIRED
  y CDATA #REQUIRED
  w CDATA #REQUIRED
  h CDATA #REQUIRED
>
<!ELEMENT TextesBase (TxtBase+)>
<!ELEMENT TxtBase EMPTY>
<!ATTLIST TxtBase
  x CDATA #REQUIRED
  y CDATA #REQUIRED
  w CDATA #REQUIRED
  h CDATA #REQUIRED
>
<!ELEMENT AncreVers (AncV+)>
<!ELEMENT AncV EMPTY>
<!ATTLIST AncV
  x CDATA #REQUIRED
  y CDATA #REQUIRED
  w CDATA #REQUIRED
  h CDATA #REQUIRED
>
<!ELEMENT Legendes (Legende+)>
<!ELEMENT Legende EMPTY>
<!ATTLIST Legende
  x CDATA #REQUIRED
  y CDATA #REQUIRED
  w CDATA #REQUIRED
  h CDATA #REQUIRED
>
<!ELEMENT Auteurs (Auteur+)>
<!ELEMENT Auteur EMPTY>
<!ATTLIST Auteur
  x CDATA #REQUIRED
  y CDATA #REQUIRED
  w CDATA #REQUIRED
  h CDATA #REQUIRED
>
```

Listage 3.9 : DTD validant l'étiquetage en XML.

Le listage 3.10 montre le résultat d'étiquetage en format XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<Labeling image="AlHayat_01_12_2003.tif">
  <Titres>
```

```
      <Titre h="364" w="4160" x="220" y="1188"/>
      <Titre h="624" w="1421" x="2913" y="2030"/>
      <Titre h="551" w="452" x="2339" y="2038"/>
   </Titres>
   <TextesBase>
      <TxtBase h="357" w="673" x="3707" y="1628"/>
      <TxtBase h="432" w="708" x="2998" y="1553"/>
      <TxtBase h="432" w="698" x="2299" y="1553"/>
      <TxtBase h="432" w="703" x="1595" y="1553"/>
      <TxtBase h="432" w="703" x="891" y="1553"/>
      <TxtBase h="322" w="670" x="220" y="1553"/>
   </TextesBase>
   <AncreVers>
      <AncV h="109" w="670" x="220" y="1876"/>
      <AncV h="54" w="740" x="2852" y="3359"/>
      <AncV h="49" w="473" x="2329" y="3364"/>
   </AncreVers>
   <Legendes>
      <Legende h="41" w="1300" x="870" y="3558"/>
      <Legende h="42" w="737" x="2854" y="6141"/>
   </Legendes>
   <Auteurs>
      <Auteur h="74" w="673" x="3707" y="1553"/>
      <Auteur h="86" w="367" x="3966" y="2714"/>
      <Auteur h="60" w="415" x="2339" y="2620"/>
   </Auteurs>
</Labeling>
```

Listage 3.10 : Résultat d'étiquetage en XML.

Le listage 3.11 montre la feuille de style XSLT, rattachée à xmillum, dont laquelle nous définissons les différentes étiquettes permettant l'étiquetage des entités blocs de texte. L'ensemble des étiquettes est définit dans **<xmi:flag>**. Chaque étiquette est définie dans un style bien définit dont la définition est similaire à celle défini dans le listage 3.3.

```
<?xml version="1.0"?>
 <xsl:stylesheet version="1.0"  xmlns:tmp="tmp"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="segmentation">
    <xmi:document>
        <xmi:handler name="label" class="iiuf.xmillum.handlers.Labeling">
           <param name="flag" value="type"/>
        </xmi:handler>
        <xmi:flag name="type">
           <value name="Titre" style="titre-style"/>
           <value name="Auteur" style="auteur-style"/>
           <value name="TxtBase" style="txtbase-style"/>
           <value name="AncV" style="ancv-style"/>
           <value name="Legende" style="legende-style"/>
        </xmi:flag>
        <xmi:style name="titre-style">
           <param name="foreground" value="yellow"/>
           <param name="transparency" value="0.4"/>
           <param name="fill" value="true"/>
        </xmi:style>
        <xmi:style name="auteur-style">
           <param name="foreground" value="orange"/>
           <param name="transparency" value="0.2"/>
           <param name="fill" value="true"/>
        </xmi:style>
```

```
        <!-- Style for solid, Red, transparent blocks -->
        <xmi:style name="red-style">
           <param name="foreground" value="red"/>
           <param name="transparency" value="0.2"/>
           <param name="fill" value="true"/>
        </xmi:style>
        …
    </xmi:document>
  </xsl:template>
</xsl:stylesheet>
```

Listage 3.11 : Exemplaire d'une feuille de style XSLT permettant l'étiquetage.

La figure 3.6 illustre l'étiquetage des entités blocs de texte au moyen de *xmillum*. Dans cette figure l'utilisateur chercheur attribue une étiquette au bloc sélectionné par le biais d'un menu contextuel.

Figure 3.6 : étiquetage des entités blocs de texte avec *xmillum*

3.4.2.2 Format des caractéristiques des étiquettes

Pour permettre au réseau de neurones artificiels d'apprendre l'étiquetage logique des blocs effectués par l'utilisateur chercheur, nous devons extraire les caractéristiques à partir des blocs étiquetés. Cette étape permet de préparer soit le fichier d'entraînement, soit le fichier de test des réseaux de neurones artificiels.

Au fichier d'étiquetage logique, nous ajoutons, les caractéristiques relatives aux étiquettes. Ces caractéristiques sont calculées et normalisées par un programme écrit en Java. Le choix des caractéristiques est décrit en détail dans le chapitre 6.
Le format des caractéristiques des étiquettes que nous avons utilisés, est représenté en XML sous un format décrit pas la DTD du listage 3.11.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Labeling (Titre+, AncV+, TxtBase+, Legende+, Auteur+)>
<!ATTLIST Labeling
  image CDATA #REQUIRED
>
<!ELEMENT Titre (Rapport, Density, CC)>
<!ATTLIST Titre
  x CDATA #REQUIRED
  y CDATA #REQUIRED
  w CDATA #REQUIRED
  h CDATA #REQUIRED
>
<!ELEMENT AncV (Rapport, Density, CC)>
<!ATTLIST AncV
  x CDATA #REQUIRED
  y CDATA #REQUIRED
  w CDATA #REQUIRED
  h CDATA #REQUIRED
>
<!ELEMENT TxtBase (Rapport, Density, CC)>
<!ATTLIST TxtBase
  x CDATA #REQUIRED
  y CDATA #REQUIRED
  w CDATA #REQUIRED
  h CDATA #REQUIRED
>
<!ELEMENT Auteur (Rapport, Density, CC)>
<!ATTLIST Auteur
  x CDATA #REQUIRED
  y CDATA #REQUIRED
  w CDATA #REQUIRED
  h CDATA #REQUIRED
>
<!ELEMENT Legende (Rapport, Density, CC)>
<!ATTLIST Legende
  x CDATA #REQUIRED
  y CDATA #REQUIRED
  w CDATA #REQUIRED
  h CDATA #REQUIRED
>
<!ELEMENT Rapport EMPTY>
<!ATTLIST Rapport
  hauteur_largeur CDATA #REQUIRED
>
<!ELEMENT Density EMPTY>
<!ATTLIST Density
  white CDATA #REQUIRED
  black CDATA #REQUIRED
>
<!ELEMENT CC EMPTY>
<!ATTLIST CC
  number CDATA #REQUIRED
>
```

Listage 3.11 : DTD validant les caractéristiques des étiquettes.

Un exemple du fichier des caractéristiques des étiquettes, est présenté dans le listage 3.12.

```
<?xml version="1.0" encoding="UTF-8"?>
<Labeling image=" files\AlHayat_01_12_2003.tif">
  <Titre x="220" y="1188" w="4160" h="364">
     <Rapport hauteur_largeur="0.0875" />
     <Density white="0.152983" black="0.847016" />
     <CC number="0.03975" />
  </Titre>
  <Titre x="2913" y="2030" w="1421" h="624">
     <Rapport hauteur_largeur="0.439127" />
     <Density white="0.172882" black="0.827117" />
     <CC number="0.0155" />
  </Titre>
  …
  <AncV x="220" y="1876" w="670" h="109">
     <Rapport hauteur_largeur="0.162686" />
     <Density white="0.051937" black="0.948062" />
     <CC number="0.00475" />
  </AncV>
  …
  <TxtBase x="3707" y="1628" w="673" h="357">
     <Rapport hauteur_largeur="0.530460" />
     <Density white="0.094580" black="0.905419" />
     <CC number="0.04" />
  </TxtBase>
  <TxtBase x="2998" y="1553" w="708" h="432">
     <Rapport hauteur_largeur="0.610169" />
     <Density white="0.104421" black="0.895578" />
     <CC number="0.0615" />
  </TxtBase>
  …
  <Legende x="870" y="3558" w="1300" h="41">
     <Rapport hauteur_largeur="0.031538" />
     <Density white="0.131651" black="0.868348" />
     <CC number="0.01975" />
  </Legende>
  …
  <Auteur x="3707" y="1553" w="673" h="74">
     <Rapport hauteur_largeur="0.109955" />
     <Density white="0.069876" black="0.930123" />
     <CC number="0.00775" />
  </Auteur>
  …
</Labeling>
```

Listage 3.12 : Exemple du fichier des caractéristiques des étiquettes.

3.5 Formats de représentation des fichiers du réseau de neurones artificiels

Les fichiers de la construction des caractéristiques de *PLANET* et de *LUNET* ne peuvent pas être utilisés directement par le simulateur du réseau de neurones. Ce dernier possède un format propriétaire et il n'accepte pas les fichiers au format XML. De ce fait, une

conversion de ces fichiers des caractéristiques est requise dans le format supporté par le simulateur du réseau de neurones.

Nous détaillerons dans les sous-sections suivantes les différents formats de fichiers requis par le simulateur du réseau de neurones :

- le format du fichier d'entraînement et de test,
- le format du fichier résultat du réseau de neurones artificiels.

3.5.1 Format du fichier de données

Les fichiers de données utilisés pour l'entraînement et le test du réseau de neurones artificiels sont en format texte. La principale différence entre les deux fichiers réside dans le fait que le fichier de test ne comprend pas les sorties désirées de la couche de sortie du réseau.

La syntaxe du fichier d'entraînement, décrite dans le listage 3.13, comprend trois sections. La première section, décrit l'entête du fichier, elle comprend le numéro de version du simulateur utilisé et la date de création du fichier. La deuxième section comprend le nombre de patterns utilisés pour l'entraînement, le nombre de caractéristiques dans la couche d'entrée du réseau et le nombre de neurones dans la couche de sortie. La troisième section comprend les valeurs des caractéristiques pour chaque entrée et la valeur de la sortie désirée.

```
SNNS pattern definition file V <numéro de version>
generated at <date de génération>

No. of patterns : <nombre de données d'entraînement>
No. of input units : <nombre de caractéristiques dans la couche d'entrée>
No. of output units : <nombre de neurones dans la couche de sortie>

# Input pattern <numéro séquentiel indiquant le pattern> :
<val1, val2, val3, …..., val n>
# Ouput pattern  <numéro séquentiel indiquant le pattern> :
<val désirée 1, val désirée 2, val désirée 3, …..., val n>
….
```

Listage 3.13 : Syntaxe du fichier entraînement du réseau de neurones artificiels.

Un exemple du fichier d'entraînement du réseau de neurones artificiels de *PLANET* est présenté dans le listage 3.14.

```
SNNS pattern definition file V3.2
generated at Dec 30, 2003 3:24:41 PM

No. of patterns : 4
No. of input units : 11
No. of output units : 1

# Input pattern 1:
0.0164 0.025333334 0.689 0.310 0.00125 0.817 0.182 0.0795 0.1724 0.247 0.408
# Ouput pattern 1:
```

```
1
# Input pattern 2:
0.1908 0.247 0.832 0.167 0.08075 0.815 0.184 0.01925 0.038 0.24666 0.74
# Ouput pattern 2:
1
# Input pattern 3:
0.2296 0.247 0.830 0.169 0.1 0.796 0.203 0.00425 0.0172 0.112 0.539
# Ouput pattern 3:
1
# Input pattern 4:
0.2272 0.24733 0.824 0.175 0.101 0.805 0.194 0.00425 0.0172 0.11733 0.548
# Ouput pattern 4:
0
```

Listage 3.14 : Exemple du fichier d'entraînement du RNA de *PLANET*.

D'après le listage 3.14, les valeurs des neurones de la couche d'entrée du réseau (inputs patterns), représentent les valeurs de caractéristiques des entités extraites du fichier XML. Chaque valeur est attribuée à un neurone artificiel de la couche d'entrée. La valeur du neurone de la couche de sortie du réseau (output pattern), représente la sortie désirée avec comme valeurs possibles 1 ou 0. Pour *LUNET*, chaque valeur de sorties représente une étiquette parmi cet ensemble (Titre, Auteur, Texte de base, Ancre vers et Légende) que nous avons utilisé. Un exemple du fichier d'entraînement du réseau de neurones artificiels de *PLANET* est illustré dans le listage 3.15.

```
SNNS pattern definition file V3.2
generated at 7 févr. 2005 16:34:51

No. of patterns : 5
No. of input units : 6
No. of output units : 5

# Input pattern 1:
0.1924 1.0136666 0.158 0.769 0.230 0.02175
# Ouput pattern 1:
1    0    0    0    0
# Input pattern 2:
0.024 0.14733334 0.135 0.870 0.129 0.035
# Ouput pattern 2:
0    1    0    0    0
# Input pattern 3:
0.1424 0.162 0.732 0.815 0.184 0.04675
# Ouput pattern 3:
0    0    1    0    0
# Input pattern 4:
0.0188 0.16266666 0.096 0.949 0.050 0.00325
# Ouput pattern 4:
0    0    0    1    0
# Input pattern 4:
0.0148 0.44133332 0.027 0.850 0.149 0.02125
# Ouput pattern 4:
0    0    0    0    1
```

Listage 3.15 : Exemple du fichier d'entraînement du RNA de *LUNET*.

La syntaxe du fichier de test est similaire à celle du fichier d'entraînement ; sauf que les sorties désirées n'y figurent pas.

3.5.2 Format du fichier résultat du réseau de neurones artificiels

Le résultat de la reconnaissance par le réseau de neurones artificiels est un fichier qui comprend les valeurs de sorties du RNA. L'exemple de fichier résultat du RNA de *PLANET* présenté dans le listage 3.16, comprend la valeur calculée du neurone de la couche de sortie pour chaque entrée au RNA. Pour distinguer entre le fichier d'entraînement et de test, le simulateur ajoute les informations suivantes entité de début (`startpattern`), entité de fin (`endpattern`) et une numérotation particulière des entités comportant un numéro séquentiel représentant l'ordre d'apparition de l'entité dans le fichier. Par exemple (`#2.1`) désigne la deuxième entité.

```
SNNS result file V1.4-3D
generated at Tue Dec 30 15:32:16 2003

No. of patterns     : 4
No. of input units  : 11
No. of output units : 1
startpattern        : 1
endpattern          : 4
#1.1
0.42032
#2.1
0.9814
#3.1
0.99984
#4.1
0.99994
```

Listage 3.16 : Exemple du fichier résultat du réseau de neurones artificiels de *PLANET*.

Le fichier résultat du RNA de *LUNET* comporte plusieurs valeurs de sorties tel que présenté dans le listage 3.17.

```
SNNS result file V1.4-3D
generated at Wed Mar 09 11:36:46 2005

No. of patterns     : 4
No. of input units  : 6
No. of output units : 5
startpattern        : 1
endpattern          : 4
#1.1
1          0     0          0.00005    0
#2.1
1          0     0.00003    0.00446    0
#3.1
0.03678    0     0.94239    0          0
#4.1
0.00487    0     0.00108    0          0.96329
```

Listage 3.17 : Exemple du fichier résultat du réseau de neurones artificiels de *LUNET*.

Il est à noter que le fichier résultat de *PLANET* est converti en XML dans le but d'enrichir le fichier résultat de la reconnaissance de structures physiques. Cette conversion nécessite le fichier des caractéristiques respectif pour effectuer la mise en correspondance.

Après cette conversion, nous mettons à jour le fichier résultat de la reconnaissance de structures physiques avec le fichier résultat du réseau de neurones artificiels (RNA) de *PLANET*, converti en XML.

En revanche pour *LUNET*, le fichier résultat est converti en XML, sans qu'il y ait une mise à jour du fichier résultat de la reconnaissance de structures logiques avec le fichier résultat du réseau de neurones artificiels (RNA). Vu que nous n'avons pas développé un système de reconnaissance de structures logiques comme celui pour les structures physiques.

3.6 Conclusion

Dans ce chapitre nous avons tout d'abord montré l'architecture globale, le flux d'information véhiculé par le système de reconnaissance de structures physiques et de structures logiques, de la correction interactive des résultats de reconnaissance et d'apprentissage. Puis, nous avons montré les avantages procurés par l'utilisation de XML en tant que format de représentation des résultats de reconnaissance, et format de construction des caractéristiques pour le RNA. Enfin, nous avons décrit les formats de représentation des fichiers du réseau de neurones artificiels.

Devant la multitude de méthodes de reconnaissance de structures physiques et logiques, les chercheurs se trouvent confrontés aux problèmes d'échange de données et de format de représentation des résultats de reconnaissance. Une éventuelle solution à ces problèmes serait de généraliser l'utilisation de XML par la communauté scientifique traitant les documents.

Chapitre 4

Reconnaissance de documents complexes : cas de l'arabe.

De nombreux travaux de recherche se sont focalisés sur la reconnaissance de structures physiques de documents à structures simples en langue latine. Nous nous intéressons dans ce travail à la reconnaissance de structures physiques de documents complexes écrits en langue arabe pour développer un système de reconnaissance correspondant aux spécificités de ce genre de documents [35].

L'approche adoptée pour la reconnaissance de la structure d'un document se veut généraliste en vue de couvrir un large champ de classes de documents. Il a donc été jugé judicieux, dans le cadre de ce travail de recherche, de mettre au point un système de reconnaissance en deux phases. La première phase s'appuie sur les méthodes simples de reconnaissance de structures physiques utilisées pour les documents écrits en langue latine qui sont adaptées et étendues pour les documents complexes arabes. Nous obtenons ainsi, une plateforme standard qui permet la production de données utiles pour *PLANET*[5] qui constitue la deuxième phase étape de la thèse qui utilise des modèles à base de réseaux de neurones artificiels et qui sera décrit dans le chapitre suivant.

Les journaux, les brochures, les magazines,… sont autant de documents à structures complexes, parmi lesquels le journal a été choisi comme sujet d'études tout au long de ce travail de recherche. L'utilité d'un système de reconnaissance d'images de journaux est examinée dans ce chapitre avant de passer en revue les classes de documents qui ont été utilisées. Ensuite, l'adaptation des méthodes valables pour les documents à alphabets latins est détaillée avant de présenter les résultats obtenus avec cette approche.

4.1 L'utilité de la reconnaissance d'images de journaux

L'avènement d'Internet représente aux yeux des éditeurs de journaux une alternative intéressante au support papier classique. De ce fait, les éditeurs de journaux ont deux alternatives quand à l'utilisation d'Internet comme support de publication : soit une édition Internet du journal, soit vers une édition électronique du journal.
L'édition Internet du journal consiste en une refonte du contenu éditorial du journal avec, généralement, un enrichissement moyennant l'adjonction de liens, d'images, de

[5] Pour Physical Layout Analysis of classes of documents using artificial neural NETs

graphiques et de contenu multimédia avec du son et de la vidéo. En effet le rédacteur en chef du journal en ligne n'est plus contraint par la taille des articles, des images, des graphiques… et plus généralement n'a plus les contraintes imposées par l'édition papier. De ce fait, les articles dans la version Internet sont plus développés. Parallèlement, la recherche d'information dans Internet a progressé à grands pas et s'avère simple à implémenter grâce au format HTML structuré. Tous les grands moteurs de recherches sont capables d'une indexation de qualité d'un grand volume de document HTML

L'édition électronique du journal consiste à mettre à disposition du lecteur soit tout le contenu de la version papier soit des extraits de celle-ci. Dans ce cas, c'est le format PDF qui est généralement choisit. PDF est largement utilisé pour l'impression et le transfert de documents électroniques. De multiples améliorations ont été apportées au format PDF. Celles-ci permettent, grâce à des balises, de représenter aussi bien les structures physiques que logiques des documents mais ces balises restent inexploitées par les générateurs de ce format. Il en découle que la recherche à partir de ce format, sans connaissance des structures, ne donne pas de résultats probants surtout en présence de documents à structures complexes.
Il a été montré dans un récent travail de recherche [41] que l'extraction des données, toute seule, à partir d'un reverse engineering d'un document au format PDF ne permet pas d'obtenir les structures physiques du document, car ce document a juste été généré dans un but d'impression et de transfert. En vue d'extraire les structures un outil nommé Xed[6] [41] effectue une analyse conjointe de l'image du document PDF et des données extraites du document PDF pour en reconnaître les structures. Cette méthode d'analyse conjointe permet de construire des applications de recherche, d'archivage, d'extraction de structures logiques et autorise une recherche multicritère performante. Nous constatons dès lors l'utilité de la reconnaissance de structures physiques d'images de journaux pour les journaux en lignes au format PDF.
Pour les vieux journaux ayant été édité avant l'avènement des techniques informatiques et n'existant que sur support papier une étape préliminaire de numérisation est nécessaire, En effet, en dépit des efforts importants accomplis en matière de numérisation, essentiellement pour les bibliothèques (comme la bibliothèque nationale de France), il existe encore un nombre assez important de documents sur support papier uniquement, dont les vieux journaux. Une fois numérisés ces derniers passent par un prétraitement pour éliminer les artéfacts, avant de passer par l'étape de reconnaissance de l'image, en procédant à l'extraction de structures physiques. Les journaux récents existent sous forme numérique, ils ne nécessitent aucun prétraitement et ils passent directement à l'étape de reconnaissance de l'image.

Bien que les écrans actuels permettent d'afficher des résolutions assez élevées et permettent ainsi de faciliter la lecture d'un journal électronique en ligne, l'être humain préfère encore aujourd'hui la lecture du journal au format papier. Cependant, l'amélioration de la technologie actuelle des écrans et des performances de la reconnaissance de structures physiques et logiques permettra à terme de concevoir de nouveaux types d'applications qui pousseront de plus en plus l'être humain à préférer la version électronique à la version sur papier.

[6] Pour eXtracting electronic documents

4.2 Classes de documents utilisés

Les classes de documents utilisées tout au long de ce travail de recherche, sont des journaux en langue arabe qui possèdent une grande variabilité dans leur structure, aussi bien intra- qu'inter-classes. Ces journaux ont été utilisés aussi bien pour le système de la reconnaissance de structures physiques que pour *PLANET*. Ces journaux sont au nombre de trois : ANNAHAR, AL HAYAT et AL QUDS. Tous ces journaux existent sous forme électronique en format PDF et peuvent être téléchargés depuis Internet. Le premier, ANNAHAR, est un journal Libanais alors que le deuxième et le troisième sont des journaux indépendants édités à Londres. La figure 4.1 illustre un exemplaire de chacun des trois journaux.

Une fois téléchargés en format PDF, ils sont convertis dans une forme image. Il est à noter que cette forme peut être obtenue de n'importe quel document et est considérée comme étant le point commun entre tous les documents dans les différents formats. La forme image obtenue est sauvegardée en format TIFF [3]. Celle-ci est en niveaux de gris et elle est une image synthétisée ou idéale. En effet, nous appelons "image idéale" les images qui ne renferment ni du bruit, ni les autres artéfacts issus d'une numérisation. Elles sont utilisables directement, et aucun prétraitement n'est nécessaire.

En revanche les images numérisées par un scanner introduisent des déformations et doivent subir un prétraitement qui consiste en : filtrage, redressement, lissage, squelettisation, binarisation.

Figure 4.1 : Exemple de page des journaux ANNAHAR, AL HAYAT et AL QUDS

Avant de passer à l'étape de la reconnaissance de structures physiques les images en niveaux de gris obtenues sont converties en des images binaires par seuillage global. À partir des images binaires obtenues, nous procédons à l'extraction de structures physiques. Le système de reconnaissance de structures physiques développé au cours du présent travail de recherche sera détaillé dans les sections suivantes.

Dans la figure 4.1 on voit une nette différence entre les structures physiques des trois images de journaux : ceci représente la variabilité inter-classes. Cette différence est moins importante entre deux numéros d'un même journal. Chaque journal possède donc une structure physique bien définie qui varie en fonction du contenu éditorial et de diverses contraintes, entre autres typographiques : ceci représente la variabilité intra-classes telle que celle illustrée dans la figure 4.2.

Figure 4.2 : Variabilité intra-classes

Globalement, tous les journaux sont bâtis autour des mêmes entités physiques à savoir : les filets, les cadres, les images, les textes et blocs ; et nous nous intéressons à leurs reconnaissances.

Néanmoins, l'utilisation des entités pour bâtir la structure physique du journal diffère d'un éditeur à l'autre. Par exemple, les filets utilisés pour séparer les articles du journal, peuvent être représentés par un segment continu chez l'un et par des segments discontinus chez un autre éditeur.

Le système de reconnaissance de structures physiques que nous avons développé a été conçu de manière à être simple et flexible, tout en permettant l'ajout de nouvelles classes de documents sans aucune modification mais permettant aussi d'y greffer un module d'apprentissage. Le module d'apprentissage peut être supervisé moyennant une interaction homme machine, comme c'est le cas pour *PLANET*.

4.2.1 Quelques spécificités des journaux ANNAHAR, AL HAYAT et AL QUDS

Il est difficile d'énumérer toutes les spécificités des journaux ANNAHAR, AL HAYAT et AL QUDS en raison de la variabilité intra et inter-classes de ces derniers. Cependant, il existe certaines spécificités récurrentes aux trois journaux arabes. En effet, les entités suivantes sont communes aux trois journaux : les filets, les cadres, les images, les lignes et les blocs de texte. La disposition de ces entités est variable et les articles composés à

partir de ces entités ne possèdent pas un style unique. En réalité plusieurs combinaisons possibles sont plausibles.
Les trois journaux possèdent des spécificités communes :

- une disposition variable des blocs ; ces derniers sont représentés sous forme rectangulaire, et leurs tailles sont variables,
- une taille et une disposition variable des images,
- une présence de filets et de cadres séparant certains articles.

Essayons de voir de plus près les spécificités intrinsèques aux journaux ANNAHAR, AL HAYAT et AL QUDS. En effet, nous avons voulu voir comment les éditeurs utilisent différemment les entités.

a) ANNAHAR

La grande majorité des articles sont séparés des espaces blancs. Le reste des articles est séparé par des filets ou par des cadres.

b) AL HAYAT

Pour AL HAYAT, certaines images sont incluses dans des cadres. Ces derniers peuvent être de deux types : complètement fermés ou non. Ils peuvent s'étaler sur la largeur du journal voir figure 4.3 (a) et dans certaines éditions ces cadres représentent des éditoriaux. On peut constater pour ce journal que certaines régions des unes sont vides, probablement en raison de la présence de publicités destinées uniquement à la version papier et que certains articles possèdent comme fond une image, tel qu'illustré dans la figure 4.3 (b).

(a) **(b)**

Figure 4.3 : Spécificités du journal AL HAYAT : (a) cadre sur toute la largeur du journal, (b) article possédant comme fond une image : en bas à gauche.

c) AL QUDS

Pour AL QUDS certains cadres sont disposés sur la hauteur du journal à l'instar du journal AL HAYAT. Il est à noter que certains cadres représentent des éditoriaux voir figure 4.4.


القدس العربي
AL-QUDS AL-ARABI
مظاهرات حاشدة تؤيد طلب السيستاني للانتخابات
العراق على طريق «الخلخلة»

Figure 4.4 : Spécificités du journal AL QUDS : cadre sur toute la hauteur du journal.

4.3 Reconnaissance de documents complexes en langue arabe

Dans le chapitre 2 nous avons passé en revue les différents travaux de recherche pour la reconnaissance de structures physiques et nous avons distingué : les méthodes descendantes, les méthodes ascendantes et les méthodes mixtes. L'adaptation des méthodes simples pour la reconnaissance de structures physiques de documents complexes pour traiter la langue arabe que nous avons développée repose sur l'utilisation d'une méthode ascendante et d'une méthode mixte. La méthode ascendante que nous avons développée permet : la séparation texte image, l'extraction des filets et l'extraction des cadres. Elle repose sur l'utilisation des composantes connexes. En revanche, la méthode mixte que nous avons développée permet l'extraction des lignes de texte. Elle consiste en l'utilisation conjointe de l'algorithme de lissage RLSA et d'une méthode ascendante à base de la méthode des composantes connexes. Quant à la fusion des lignes de texte en blocs, elle est à base de règles qui prennent en considération les caractéristiques de la langue arabe.

Notre approche ressemble à celle proposée par Esposito [30], et cela, au niveau de la combinaison du RLSA et de la méthode des composantes connexes. Néanmoins, le choix

des documents est différent, l'évaluation effectuée par Esposito n'a pas été effectuée sur des documents à structures complexes. L'évaluation que nous avons effectuée repose sur des documents à structures complexes en langue arabe. La méthode que nous avons conçue est à base de règles simples pour permettre l'utilisation d'un grand nombre de classes de documents et pour permettre l'adjonction de l'apprentissage.

Afin de faciliter l'utilisation de notre système de reconnaissance, nous avons conçu ce dernier comme un ensemble d'outils qui peuvent être utilisés séparément. Par exemple si l'utilisateur a besoin de l'outil d'extraction de filets alors il invoquera l'outil approprié.

Dans les sous-sections suivantes nous allons détailler les méthodes pour réaliser les tâches suivantes :

- l'extraction des filets,
- l'extraction des cadres,
- la séparation texte image,
- l'extraction des lignes de texte,
- la fusion des lignes de texte en blocs.

4.3.1 Extraction des filets

Les filets sont essentiellement utilisés dans le but de délimiter les articles de journaux. Ils peuvent suivre soit une direction verticale, soit une direction horizontale et ils sont de deux types ; les filets internes et les filets externes. Les premiers sont utilisés pour séparer au sein d'un article entre l'auteur et les paragraphes de texte composant l'article. Les seconds servent à séparer des articles ou un ensemble d'articles.

La figure 4.5 illustre les différents filets à extraire.

Figure 4.5 : Filets internes (a) et filets externes horizontaux et verticaux (b)

D'après la figure 4.5, nous constatons que les filets sont composés de segments horizontaux ou verticaux dont l'épaisseur peut varier selon les éditeurs.

Notre méthode d'extraction des filets opère de la manière suivante : nous appliquons la méthode d'extraction des composantes connexes sur toute l'image binaire du document. Nous obtenons un ensemble de composantes connexes, chacune étant décrite par son rectangle englobant. Pour chaque composante connexe de cet ensemble nous calculons les rapports largeur/hauteur et hauteur/largeur pour les filets horizontaux et verticaux.

Ces rapports sont comparés avec un seuil qui est défini de la manière suivante, nous classons les largeurs des lignes de texte par ordre croissant et nous choisissons comme seuil la plus petite largeur de ligne. Tous les rapports calculés supérieurs à ce seuil sont potentiellement des filets.

De cette façon, nous évitons de prendre en compte comme filets tous les segments que nous trouvons à l'intérieur des mots ou des pseudo-mots et qui sont dus à la justification de texte en langue arabe.

4.3.2 Extraction des cadres

Un cadre est un ensemble de quatre filets horizontaux et verticaux qui se touchent par leurs extrémités. Nous distinguons le cadre complet et incomplet. Ce dernier est un cas particulier, puisqu'il comporte cinq filets dont les deux filets horizontaux du dessus ne se touchent pas par leurs extrémités. Généralement, un cadre englobe un article et peut représenter soit :

- un article en entier (voir figure 4.6 (a)),
- un article dépendant d'un autre (voir figure 4.6 (b)),
- l'éditorial (voir figure 4.6 (c)).

Les cadres sont utilisés pour mettre en valeur un article.

الاوساط المالية في عمان تنتظر عودة «شومان الصغير» وترتيبات «أردنية» لصالح الحريري أحبطت مخططا لسيطرة اجنبية على البنك العربي

عمان ـ «القدس العربي»

من يسمع يدارين:

ينتظر الجميع في الاردن بفارغ الصبر عودة النجل الوحيد للمصرفي النابلزي الراحل معز خالد شومان الذي تسبب بحالة من الإرتباك والقرف في «الاسبوع الماضي» خلال مفاوضات مكثفة في مقر البنك العربي الكبير المؤسسات المالية الاردنية والعربية لحضور «سموي ترجيه» التي يمثلها رئيس الوزراء اللبناني رفيق الحريري.

وشومان الصغير يتواجد خارج الاردن هذه الايام لكنه يفترض ان يعود قريبا لإنجاز اضخم صفقة لبيع الاسهم يشهدها القطاع المالي الاردني منذ تأسيسه وهي صفقة اثارت الكثير من الجدل لانها سمحت للحريري حصريا بالدخول لأكبر مالك للاسهم في البنك العربي «١٤٫٥٪ من الدي يسمح له بخلق ملف القديم للدخل بالسيطرة على البنك ونقل إدارته إلى بيروت وعمانه لشركة «سعودي اوجيه» تحديدا. وليس لاسم «المعجز عبد الحميد شومان مؤسس البنك العربي ورئيس مجلس إدارته حاليا» أو لشركاء أردنيين. لكن يبدو ان الدوائر اليهودية لإمتلاك حصص في البنك سارعت في ظهور ترتيبات لتطويق التي رفضت كلمة السطة على «عربية او اردنية» حصريا الحريري.

لكن مصطلحات متداولة تشير إلى العلاقات داخل عائلة شومان تفسها دفعت ورثة الراحل خالد شومان المؤسس الثاني للبنك وملكه الأبرز لاختيار الحريري دون غيره من الشركاء لبيع الحصة المورثة والقدرة ١١٫٥ من قيمة الاسهم التكلية في صفقة تناسب مع سعي عمان لبقاء البنك أردنيا. علما بان الحريري دخل خلال الاعوام الثلاثة الماضية بالكثير من معركة سورية وطنية للسيطرة على البنك ورفع حصته فيه وبالتالي يزعجه، شومان لشومان المعجز عن طريق توقيع إتفاق مع خالد شومان.

وهي هاجس الصفقة التي رفعت بحوب إتفاق مكتوب تكتوما لم تسجل بعد رسميا في وثائق سوق عمان المالي ووجود إتفاقيات متعددة للمكرمة الأردنية التي إتهمت ضمنيا بانها «سكتت» إزاء «سكولات الحريري السيطرة على البنك الأهم في الساحة المحلية وامتنعت عن إستغلال نفوذها وعلاقاتها مع الحريري لإقناعه بعدم الاقتراب من البنك العربي.

ووجهت هذه الإتهامات قبل بروز نظرية تقول بأن دخول الحريري على خط الصفقة تفسها كان محصلة لتوافق إضطراري بين المعجز عبد الحميد شومان وتحلة عبد الحميد، ودون مستويات القرار الأردنية.

ولا زالت الوساط السياسية تشعر بالقلق الشديد إن الحريري متمسك بحلمه القديم على السيطرة على البنك العربي وتعلن أن شومان عنه. وكان الحريري قد إقتنص وصول العلاقات داخل العرف، في عائلة شومان لطريق مسدود وسدد ضربة قوية لشومان المعجز عن طريق توقيع إتفاق مع خالد شومان.

ونجح الحريري بالحصول على الصفقة بمساعدة واضح ويريده في أرسلة الفال والأعمال بأنه أحدا أطراف سعودية التي إتهمت ضمنيا بانها «سكتت» إزاء «سكولات الحريري السيطرة على البنك الأهم في الساحة المحلية وامتنعت عن إستغلال نفوذها وعلاقاتها مع الحريري لإقناعه بعدم الاقتراب من البنك العربي.

وتأثير الصفقة شبهات كثيرة ومحاذير متعددة أبرزها إضعاف مكانة المعجز شومان في إدارة البنك وإحكام السيطرة عليه من قبل الحريري وجماعته، الأمر الذي سيتعكس سلبا على الإقتصاد الوطني الأردني كما يقول «القدس العربي» العلل الإقتصادي البارز خالد الريماوي الذي تحدث عن عدم وجود ضمانات تمنع الحريري إذا سيطر على الإدارة العليا من التأثير في السياسات الإئتمانية للبنك والتوجهات الإستثمارية بالإتجاه التي تناسب معه ولبس مع مصالح الأردن حكومة وشعبا.

والريماوي قاد عمليا في أيام حملة لإنقاذ ٨٫٥ مليون دينار من أصل الـ ٩٥ مليون دينار بين الحريري وورثة خالد شومان. وقال ان الأمر الإقتصادي الوطني يتطلب بقاء البنك العربي «أردنيا وعربيا» بجميع الأحوال إن هذا البنك يساهم بشكل يطل ما يقارب ٥٠٪ من النشاط الكلي في الأردن وأن وجوده في عمان يخدم في ٤٠٪ من وظيفة وإجتماعية وسياسية.

ويشكك خبراء إقتصاديون بان الحكومة الأردنية أم تتدخل في العملية بصفقة الحريري بسبب الإستثمارات الضخمة التي سيقوم بها الأخير في الأردن خلال السنين الخمس المقبلة.

لكن الحكومة وإزاء الضغوط التي أثارها الحريريون على «أردنية» البنك العربي تدخلت خلال اليومين الماضيين ومن خلال أعلى المستويات لإجراء «تعديلات» طفيفة على الصفقة فما ظهر شفويا حتى الآن بعد إضافات طارئ تدخلت عمان والرياض وبيروت والعشرات من البليزين في هذه العواصم أن يوافق الحريري على تخفيض الحصة التي سيشتريها من عمر شومان وان يكتفي بشراء ١٤٫٥٪ من أسهم البنك العربي و٨ بالمئة من أسهم البنك الإسلامي الإتراني القابض الذي قام به الحريري خلف الخطوط هو نتيجة الأفضل بأسوأ الخيارات، التي دعمها المعجز أن إفشل فيها في المحطات الأخيرة قبل دخول الحريري الذي يقال انه حلق بعيدا «الحضور» أردني كانت أطراف أخرى طبوخة، تحاول إقناعهم مساعدة في مكتبه البنك العربي عن طريق رجال أعمال يحظون الجنسية الأمريكية إقترب كثيرا من التحصول الفعلي على حصة ورثة الراحل خالد شومان قبل تدخل «الثلاثية الأردنية» بأسبوع إستغاث منها الحريري وإضطرها المعجز شومان.

وظي هاشل نص القصة ثمة فتاحة لدى الآن. أما ٥٫٨ بالمئة من أسهم البنك إستراتيجي هو مؤسسة فضمان الإجتماعي بنسبة ٥٫٨٪ على أن يتمكن المعجز شومان من شراء ٢٫٥ بالمئة من بنوك شقيقه عمر شومان.

وتحدثت هذه الترتيبات لقضاة الجميع ولضمان عدم إنقلاب الحريري أيضا إلا أن الصفقة لم تحسم بعد فالجميع ينتظر حضور شومان الصغير لضمان وينتظر القرب لمنوان للتسجيل في سوق عمان المالي لتوقيع الصفقة التي يمكن إعتبارها أفضل ما إستطاع شومان المعجز إنقاذه، من الحصة التي كانت ستذهب لشركاء آخرين حاولوا بقوة الدخول على خط البنك العربي.

ورغم ذلك لا توجد ضمانات بان يوافق شومان الصغير على «الترتيبات الأردنية» فالأدنا ذكر بان عدة أطراف تحاول تغيير اللاعبين الذين وعدتها بعض عبر الإيحاء بالموافقة على الترتيب الأردني ومبر إقناع شومان الصغير بالمضي قدما بالصفقة في

ISSN 1358-3379

(a)

Figure 4.6 : Cadres (a) article, (b) article dépendant d'un autre, (c) éditorial.

D'après la figure 4.6 nous constatons que les quatre filets composant le cadre sont dans la plupart des cas délimités par une suite de pixels blancs. En effet, si nous englobons chaque filet, composant le cadre, dans le centre d'un rectangle, alors la partie de dessus et de dessous du filet sera délimitée par une suite de pixels blancs.

La méthode d'extraction des cadres repose sur les étapes suivantes :

- nous commençons par reprendre l'ensemble des composantes connexes calculées précédemment et nous appliquons un filtre en ne prenant en considération que celles qui sont plus grandes à un seuil donné.
- afin de déterminer la présence des suites de pixels blancs en dessous de chaque filet, pour les filets horizontaux, et à coté, pour les filets verticaux composant le cadre, nous construisons pour chaque filet un rectangle ayant au minimum les dimensions du filet.
  Ces rectangles sont au nombre de quatre et sont nommés ; r_N, r_O, r_E et r_S pour rectangle nord, ouest, est et sud. Ils sont définis ainsi à partir du rectangle principal r [r.x, r.y, r.largeur, r.hauteur] :
  - r_N [r.x, r.y, r.largeur, hNord],
  - r_O [r.x, r.y, hOuest, r.hauteur],
  - r_E [r.x + r.largeur – hEst, r.y, hEst, r.hauteur],
  - r_S [r.x, r.y + r.hauteur – hSud, r.largeur, hSud]

  Les valeurs de hNord, hOuest, hEst et hSud, elles ont été choisies de telle manière à englober des pixels blancs, voir figure 4.7. Expérimentalement, nous

avons trouvé que pour les trois classes de documents la valeur de 5 pixels convenait le mieux pour hNord, hOuest, hEst et hSud.

Figure 4.7 : Le rectangle nord

- pour chaque rectangle rN, rO, rE et rS nous calculons la densité de pixels blancs. Si la densité de pixels bancs pour chaque rectangle (nord, sud, ouest et est) est supérieure à un seuil alors la composante connexe est un cadre potentiel.

La figure 4.8 illustre les rectangles Nord (r_N), Sud (r_S), Ouest (r_O) et Est (r_E) utilisés pour l'extraction des cadres.

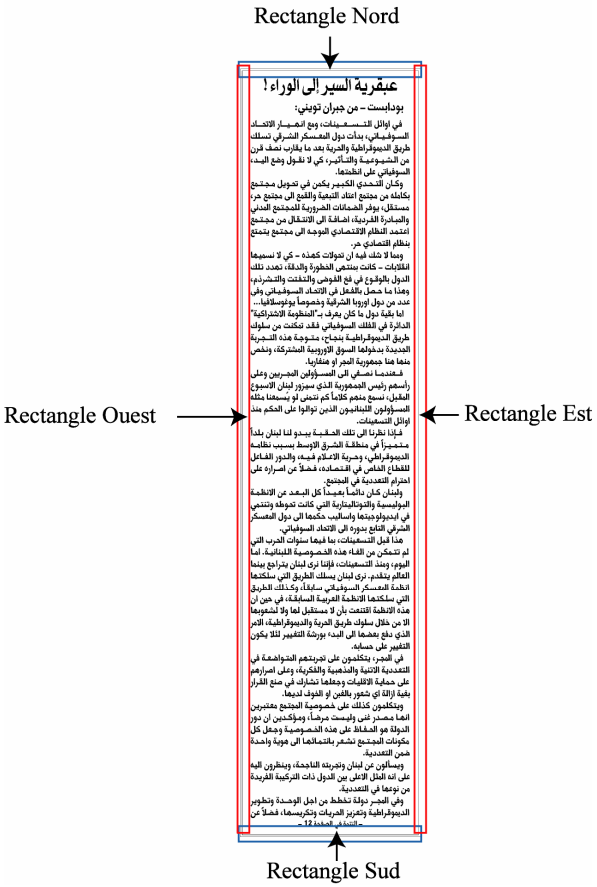


Figure 4.8 : Rectangles nord, sud, ouest et est

4.3.3 Séparation texte / image

Les images sont très utilisées dans les journaux, généralement en complément à un article, pour illustrer un événement. Il est nécessaire dans le cas de la reconnaissance de structures physiques de documents des les séparer du texte.

La méthode adoptée pour la séparation texte/image a été appliquée sur des images de documents binaires. Ce choix est justifié par le fait que nous avons voulu uniformiser toutes nos méthodes d'extraction de filets, de cadres,… en les appliquant sur des images binaires de documents, et aussi par le fait que avons voulu que notre système de reconnaissance de structures physiques soit conçu de manière a être simple et flexible.

Les étapes de notre méthode pour la séparation texte/image sont les suivantes :

1. Nous reprenons l'ensemble des composantes connexes obtenues lors des étapes de traitements précédents et nous appliquons un filtre pour ne garder que les plus grandes car il a été constaté que les images sont représentées par des composantes de grande taille,

2. Le filtre des cadres est appliqué de nouveau pour n'obtenir que les composantes connexes candidates à la classe image,
3. Ayant constaté, que les images binaires possèdent soit une densité de pixels noirs élevée, soit une densité de pixels blancs élevée ; pour chaque composante connexe obtenue par ce dernier filtre, les densités de pixels noirs et blancs sont calculés. Si l'une de ces densités est supérieure à un seuil donné alors la composante connexe est une image potentielle,
4. Nous avons constaté que parmi les images potentielles, certaines composantes connexes se superposent. Pour résoudre ce problème nous appliquons la fusion des composantes connexes superposées.

4.3.4 Extraction des lignes de texte

Une fois la séparation texte/image effectuée nous procédons à l'extraction des lignes de texte. Dans la littérature plusieurs méthodes ont été utilisées pour l'extraction des lignes de texte. La méthode d'extraction des lignes de texte adoptée est une méthode mixte combinant l'algorithme de lissage RLSA et l'extraction des composantes connexes.
L'algorithme RLSA est appliqué en premier ; suivi par la méthode de composantes connexes. L'utilisation de l'algorithme RLSA implique l'utilisation d'un seuil, qui a été estimé par le biais d'un histogramme de la hauteur des composantes connexes pour un ensemble de documents d'une même classe. Le seuil adopté est obtenu en calculant la moyenne des hauteurs les plus fréquentes sur l'ensemble de documents. Ce processus est repris pour toutes les classes de documents.

Expérimentalement il a été vérifié que la méthode d'extraction des lignes de texte adoptée pour la segmentation en texte donne effectivement des lignes de texte correctement segmentées à 93% pour les trois classes de documents comme l'illustre la figure 4.9.


واجتماعياً. فزيارته لمنطقة البترون
امس وملاقاته البطريرك صفير في
هذه المناسبة، جاءتا عقب اجتماعيه مع
"لقاء قرنة شهوان" و"اللقاء
التشاوري"، وهما اللقاءان اللذان كان
فيه دور البطريرك الماروني ماثلا بقوة
في المداولات التي جرت والتي عكست
اقتناعا شاملا بان مواقف البطريرك لا
بد من ان تكون محور حوار جدي يراد
له النجاح، وكذلك محور كل الاتصالات
المرتبطة بالمناخ الحواري. ويشار في
– التتمة في الصفحة ٩ –

Figure 4.9 : Segmentation en lignes de texte

En fait, lors de l'étape de détermination du seuil nous avons privilégié la segmentation correcte des lignes de textes composant le corps de l'article au détriment d'une bonne segmentation des lignes de textes composant les titres de l'article. En effet, les titres d'articles sont souvent des lignes de texte avec des caractères de plus grande taille que le corps de l'article.
La segmentation des titres avec cette méthode donne souvent soit des points diacritiques (placé au dessus ou au dessus de la lettre) non rattachés aux lettres, soit des lettres non rattachées aux lignes trouvées, soit enfin le dédoublement de lettres (chadda) qui est traité comme un point diacritique.
Ces anomalies sont dues à la fixation du seuil dans l'algorithme RLSA ; si nous fixons le seuil pour segmenter les lignes de paragraphes, celle-ci sont correctement segmentées alors que les lignes de titre ne le sont pas et inversement si nous fixons le seuil pour les titres.

La figure 4.10 illustre les erreurs de sur-segmentation obtenues avec les titres.

Figure 4.10 : Titres sur-segmentés

Pour résoudre les problèmes de sur-segmentation des titres cités précédemment et illustrés dans la figure 4.10, il a été procédé à une fusion dans trois directions :

- de droite à gauche pour fusionner les mots ou les pseudos mots,
- vers le haut à partir du bord inférieur du bloc, pour la fusion les points diacritiques existant au dessus de la lettre ainsi que le dédoublement de lettres (la chadda)
- vers le bas à partir du bord inférieur du bloc pour la fusion les points diacritiques existant en dessous de la lettre.

La figure 4.11 illustre bien les résultats de la fusion appliquée aux titres d'articles.

(a) (b)

Figure 4.11 : Étapes et résultat de la fusion : (a) directions, (b) résultat

Le processus de fusion étendu qui a été développé pour corriger les erreurs de segmentation opère ainsi :

1. les rectangles sont triés par ordre décroissant avec comme critère la hauteur du rectangle pour n'avoir que les rectangles qui composent le titre,
2. pour chaque rectangle nous construisons quatre rectangles r_N, r_O, r_E et r_S (pour rectangle nord, ouest, est et sud) à partir du rectangle principal r [r.x, r.y, r.largeur, r.hauteur].
   Ces rectangles sont définis ainsi :
   - r_N [r.x, r.y – r.hauteur –s1, r.largeur, r.hauteur],
   - r_O [r.x – s2 – w1, r.y, w1, r.hauteur],
   - r_E [r.x + r.largeur + s2, r.y, w1, r.hauteur],
   - r_S [r.x, r.y + r.hauteur + s1, r.largeur, r.hauteur]
3. pour chacun de ces rectangles r_N, r_O, r_E et r_S, nous recherchons l'ensemble des rectangles qui les intersectent.
4. nous fusionnons ces rectangles pour obtenir les lignes titres de texte.

La figure 4.12 illustre les différents rectangles ainsi que les seuils utilisés.

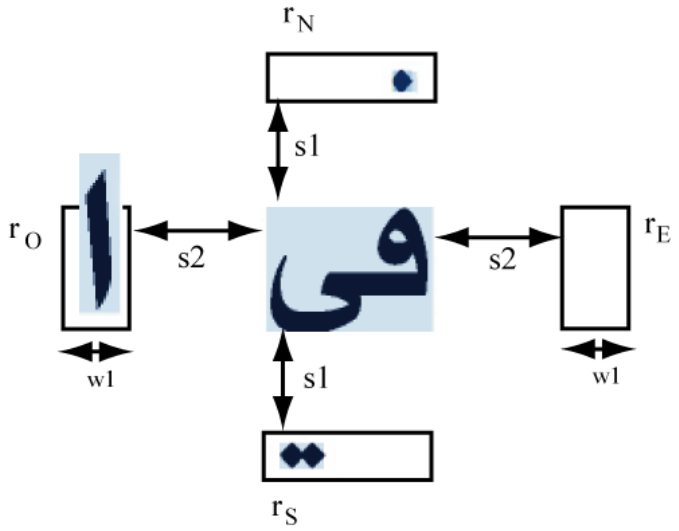


Figure 4.12 : Les rectangles nord, ouest, est et sud et les seuils

4.3.5 Fusion des lignes de texte en blocs

La dernière étape consiste en la fusion des lignes de texte, que nous avons obtenu précédemment, en blocs. Afin de réaliser cette fusion, nous avons utilisé des règles en prenant en considération les caractéristiques de la langue arabe. Parmi ces règles nous trouvons celle relative au sens de l'écriture de la langue arabe.

Les étapes de l'algorithme sont les suivantes :

1- Pour chaque ligne de texte représentée par un rectangle nommé rectangle principal (r_P) et issue de l'ensemble des lignes de texte obtenu : nous construisons

deux rectangles : le rectangle nord (r_N) et le rectangle sud (r_S) qui sont distants au maximum de χ pixels du rectangle principal :

r_N [r.x, r.y – r.hauteur – χ, r.largeur, r.hauteur],
r_S [r.x, r.y + r.hauteur + χ, r.largeur, r.hauteur]

La valeur de χ est déterminée en calculant la distance moyenne entre les lignes dans un paragraphe. Cette distance moyenne est calculée de la manière suivante : nous classons les lignes de texte par ordre croissant avec comme critère la largeur du rectangle, nous groupons les lignes de texte par leur largeur, le groupe qui possède le nombre de lignes de texte le plus élevé est sélectionné, pour chaque paire de lignes de texte de ce groupe nous calculons la distance les séparant (rectangles). Ensuite nous calculons la moyenne de valeurs des paires de lignes de texte.

2- Nous recherchons les rectangles qui intersectent le rectangle nord (r_N) et le rectangle sud (r_S) à l'intérieur de l'ensemble des lignes de texte.

3- Pour la fusion, nous distinguons 2 cas ; un cas général (a, b, c) et un cas particulier (d). Nous fusionnons donc les lignes de texte obtenues si et seulement si les conditions suivantes a et b et (c ou d) sont vérifiées :

   a. Si les rectangles intersectés possèdent la même hauteur que le rectangle principal (pour l'ensemble du paragraphe) avec un décalage par rapport à x :
      i. si $[r_N.hauteur] = [r_P.hauteur]$ et $[r_N.x + r_N.largeur] – [r_P.x + r_P.largeur] < seuil (\varsigma)$ ou si $[r_S.hauteur] = [r_P.hauteur]$ et $[r_S.x + r_S.largeur] – [r_P.x + r_P.largeur] < seuil (\varsigma)$

   b. Si la largeur du rectangle nord intersecté auquel nous ajoutons un ratio de droite est égale à la largeur du rectangle principal avec un décalage par rapport à x (pour la première ligne du paragraphe) :
      i. si $[r_N.x + r_N.largeur + ratio\_d] – [r_P.x + r_P.largeur] < seuil (\varsigma)$.

   c. Si la largeur du rectangle sud intersecté auquel nous ajoutons un ratio de gauche est égale à la largeur du rectangle principal avec un décalage par rapport à x (pour la dernière ligne du paragraphe) :
      i. si $[ratio\_g + r_S.x + r_S.largeur] – [r_P.x + r_P.largeur] < seuil (\varsigma)$;

   d. Pour la dernière ligne du paragraphe que nous trouvons centrée tel qu'illustré dans la figure 4.9. Ce cas est traité de la manière suivante :
      i. si les rectangles r_N et r_S possèdent la même hauteur et si $[ratio\_g + r_S.x + r_S.largeur + ratio\_d] – [r_P.x + r_P.largeur] < seuil (\varsigma)$.

4- Reprendre l'étape 1 pour la ligne suivante.

Nous avons constaté que la justification des lignes de texte n'entraîne pas un positionnement exact de ces lignes au pixel prés, en effet, il existe un petit décalage de quelques pixels. Le seuil ς permet d'approximer ce décalage et il est déterminé de la manière suivante : nous regroupons les lignes de texte par leur largeur, pour chaque groupe nous calculons la différence entre les positions x des rectangles englobants, ensuite nous calculons la moyenne des ces décalages pour les différents groupes.

Les ratios `ratio_d` et `ratio_g` sont déterminés à partir de la différence des largeurs entre le rectangle intersecté et les rectangles nord ou sud.

Les étapes de notre algorithme de fusion sont illustrées dans la figure 4.13.

Figure 4.13 : Étapes de notre algorithme de fusion

4.4 Résultats obtenus

Le système de reconnaissance que nous avons développé a été évalué sur un ensemble de journaux issus des trois classes. Nous rappelons que l'objectif est d'adapter les méthodes simples de reconnaissance de structures physiques de la langue latine pour la langue arabe sans ajouter une optimisation particulière sauf pour les titres. En effet, d'une part une telle optimisation rendrait le système plus dépendant des classes auxquels il a été confronté et d'autre part, il est prévu de traiter le problème au travers la reconnaissance assistée et l'apprentissage évolutif, tel que décrits dans le chapitre 5.

La procédure d'évaluation consiste à compter le nombre de similarités entre les entités (filets, cadres, images, lignes de texte et blocs de texte) extraites à partir de l'image de document segmenté et le fond de vérité correspondant. Ensuite un taux de reconnaissance est calculé pour chaque entité. Cette méthode de comparaison est similaire à celle proposée par Gatos [32] mais au lieu de comparer les ensembles de pixels intersectés entre le fonds de vérité et ceux obtenus par les méthodes d'extraction, c'est les rectangles englobant les entités qui sont comparés puisque ces dernières sont représentées par des rectangles.
Sur 99 images de documents issues de journaux ANNAHAR, AL HAYAT et AL QUDS nous avons obtenus les résultats présentés dans la table 4.1.

%	Filets	Cadres	Images	Lignes de texte	Blocs de texte
ANNAHAR	50.748	99.623	97.391	96.616	95.319
AL HAYAT	99.452	90.987	91.178	92.368	91.437
AL QUDS	94.666	96.667	94.333	92.594	91.033

Table 4.1 : Résultats de segmentation pour les trois journaux.

La table 4.1 montre que le taux de reconnaissance pour les différentes entités issues des trois classes de documents est relativement bon. Cependant, nous avons voulu savoir les raisons pour lesquelles certains taux sont moins élevés que d'autres.
Nous avons constaté que l'algorithme d'extraction des cadres ne fonctionne pas en présence des rectangles non fermés tel que ceux utilisés dans la classe de documents AL HAYAT. Et que l'algorithme d'extraction de filets ne fonctionne pas en présence des filets particuliers avec des textures telles que ceux utilisés dans la classe de documents ANNAHAR.
En ce qui concerne la segmentation en lignes de texte, elle est relativement bonne à part quelques erreurs avec les points diacritiques. Ces erreurs consistent en la fusion des points diacritiques inférieurs de la première ligne et supérieur de la deuxième ligne quand ils sont proches l'un de l'autre. Malheureusement, ces erreurs se propagent vers la phase suivante à savoir la fusion des lignes de texte en blocs qui repose précisément sur les résultats corrects de la segmentation en lignes de texte.

4.5 Conclusion

Dans ce chapitre nous avons présenté l'utilité de la reconnaissance de structures physiques pour les documents complexes et plus exactement pour les journaux au format PDF.
Certains travaux ont étudié la reconnaissance de structures physiques pour les documents complexes mais aucun pour le même type de documents en langue Arabe. Les documents à structures complexes en langue Arabe ont servi de base pour l'élaboration d'un système de reconnaissance de structures physiques en adaptant les méthodes simples de la langue latine existantes.
L'évaluation de ce nouveau système de reconnaissance de structures physiques a donné de bons résultats. Cependant, la reconnaissance purement automatique de structures physiques n'est pas appropriée pour les documents complexes vu sa grande variabilité. Nous pensons qu'il n'est pas possible d'améliorer la performance de cette approche entièrement automatique tout en maintenant la généralité par rapport aux diverses classes de documents.
La bonne démarche pour l'amélioration de ces résultats consiste à faire intervenir l'utilisateur par une reconnaissance assistée et à transférer l'expertise de l'utilisateur vers le système de reconnaissance en convertissant ses corrections en connaissances. La reconnaissance assistée et l'apprentissage sont les objectifs fixés pour *PLANET*, qui sera présenté en détail dans le chapitre suivant.

Chapitre 5

PLANET : système de reconnaissance de structures physiques doté d'apprentissage évolutif basé sur les réseaux de neurones artificiels.

Après avoir décrit dans le chapitre précédent le système de reconnaissance de structures physiques de documents arabes à structures complexes sans apprentissage, nous nous intéressons dans ce chapitre à greffer l'apprentissage au système préalablement décrit. Le nouveau système de reconnaissance de structures physiques doté d'apprentissage évolutif basé sur les réseaux de neurones artificiels s'intitule *PLANET* (Physical Layout Analysis of classes of documents using neural NETs). Ce système, décrit dans [37], permet d'améliorer le taux de reconnaissance obtenu par rapport à notre système de reconnaissance de structures physiques présenté dans le chapitre précédent, en lui ajoutant l'apprentissage des caractéristiques de chaque classe de documents.

Ce chapitre traite de la reconnaissance de structures physiques. Nous y présentons tout d'abord les principes de construction de *PLANET* ainsi que les modèles qui le composent. Ensuite, nous décrivons le principe de fonctionnement des réseaux de neurones artificiels. Puis nous décrivons les caractéristiques utilisées et la topologie des RNAs de *PLANET*. Finalement, nous présentons la démarche pour l'évaluation de *PLANET* et les résultats obtenus.

5.1 Reconnaissance assistée dotée d'apprentissage évolutif

Il est très difficile de construire un système automatique de reconnaissance qui donne de bons résultats sur des documents quelconques. C'est ce que nous avons observé avec notre système d'analyse de pages de journaux, décrit dans le chapitre 4. Par contre, il existe des systèmes de reconnaissance qui, appliqués à un ensemble restreint de documents semblables, donnent de relativement bons résultats.

Pour qu'un système de reconnaissance donne de bons résultats sur un ensemble varié de documents, il faut qu'il puisse s'adapter à l'ensemble des documents qu'il doit traiter.
A cet effet, nous introduisons la définition suivante :

Une *classe de documents* représente un ensemble de documents de même type, possédant des structures de pages similaires. Par exemple, nous pouvons regrouper tous les articles publiés dans la revue scientifique IJDAR sous forme de classe.

La notion de classe de document peut être appliquée à des niveaux de granularité différents. En d'autres termes, les classes de documents peuvent être organisées hiérarchiquement. Par exemple, les revues IJDAR peuvent être englobées dans une classe plus générale constituée de toutes les revues scientifiques. L'ensemble de tous les documents qui présentent un intérêt pour le type d'analyse qui nous intéresse sera appelé la *classe universelle*.

Afin d'améliorer les résultats de reconnaissance du système général de reconnaissance de structures physiques, trois directions s'offrent au développeur, à savoir :

- coder les connaissances extraites des documents sous forme d'algorithme : cette approche est difficilement adaptable à d'autres classes de documents ; en conséquence cette approche n'est praticable que pour des gros volumes de documents.
- utiliser des règles : cette façon de faire nécessite l'intervention d'un expert pour la définition des règles, que ce soit lors de la mise en place du système ou lors de la prise en compte d'une nouvelle classe de documents. Ces règles sont externes et sont exprimées dans un langage type IA ou grammaires attribuées.
- effectuer un apprentissage par la machine : cette approche nécessite des données d'apprentissage extraites des fonds de vérité. Le problème majeur des fonds de vérité, est que souvent, ces derniers n'existent pas ou sont coûteux à construire. Pour cette raison, il peut être utile de construire les données d'apprentissage à partir de l'intervention de l'utilisateur qui fournit ainsi des exemples qu'il a validés dans une boucle interactive.

Après avoir énuméré les trois points, nous avons opté pour le développement d'un système de reconnaissance assistée par l'utilisateur. Le système s'appelle *PLANET* et vise la reconnaissance des structures physiques de classes de documents variés, en utilisant un modèle évolutif qui s'adapte à toute nouvelle classe de documents traitée.

5.2 Les modèles

Dans cette section, nous traitons les modèles de reconnaissance. Ces derniers correspondent à des réseaux de neurones artificiels (RNAs). Un modèle représente les connaissances spécialisées d'une classe de documents nécessaires à la reconnaissance. A cet effet, nous introduisons la notion de superclasse de documents en plus des notions de classe universelle et de classes de documents que nous avons définies précédemment.

Dans la suite, nous considérons deux niveaux de classes que nous appelons respectivement :

- les classes de documents représentant chacune le journal d'un éditeur donné,
- la superclasse de documents représentant l'ensemble de tous les journaux arabes.

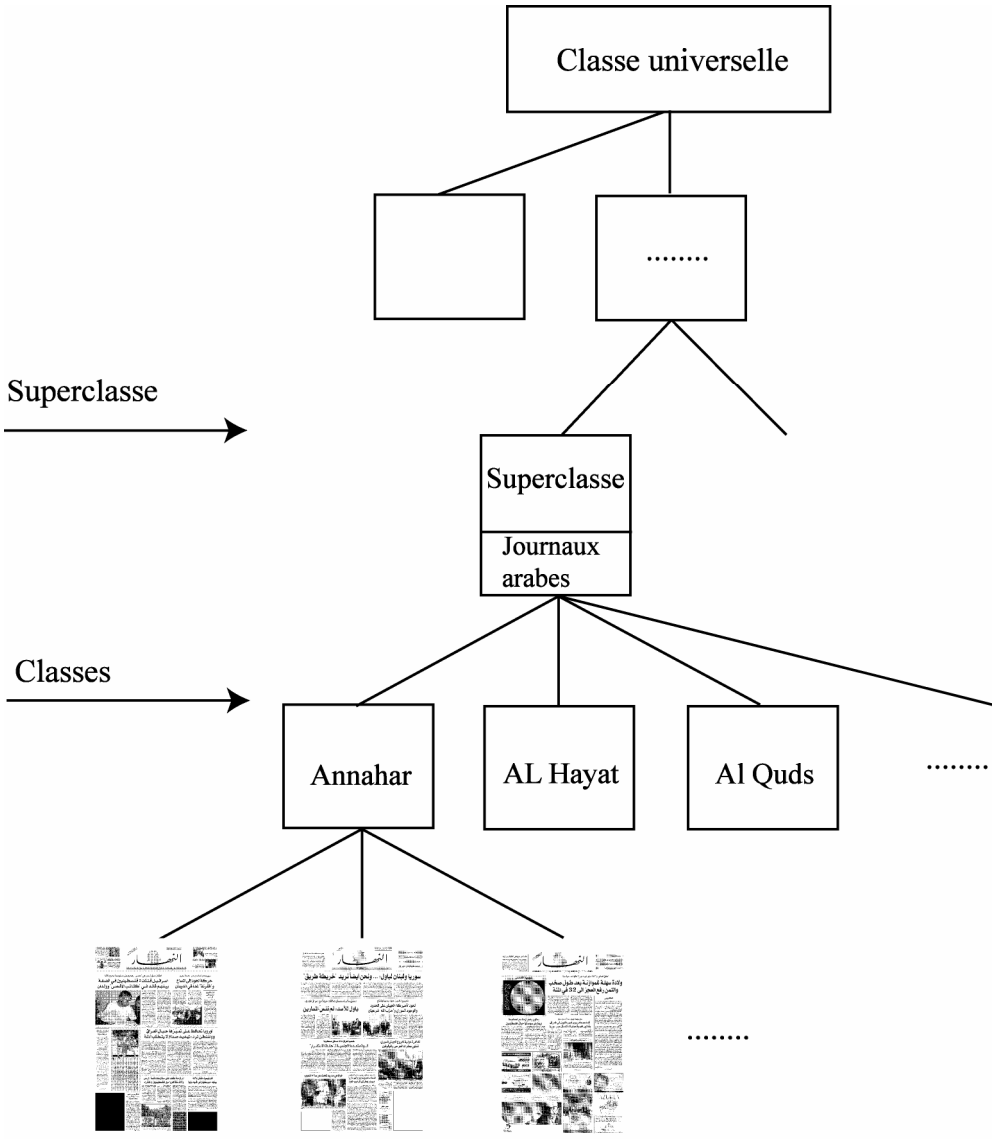


Figure 5.1 : Représentation hiérarchique des documents.

La figure 5.1 illustre la représentation hiérarchique des classes. C'est un arbre dont la racine contient la classe universelle. Au niveau n-1 de cet arbre, nous trouvons le nœud superclasse de journaux arabes et au niveau n, les feuilles représentant les différentes classes de documents (n représente le niveau de granularité).

La figure 5.2 illustre l'architecture de *PLANET* qui utilise plusieurs modèles. Dans *PLANET*, nous distinguons deux niveaux de modèles : *le modèle général* et *les modèles dédiés*. Le modèle général permet d'avoir une connaissance générale de la superclasse. A chaque classe de documents est associée un modèle, dit *dédié*, dont le but est de représenter des caractéristiques de cette classe.

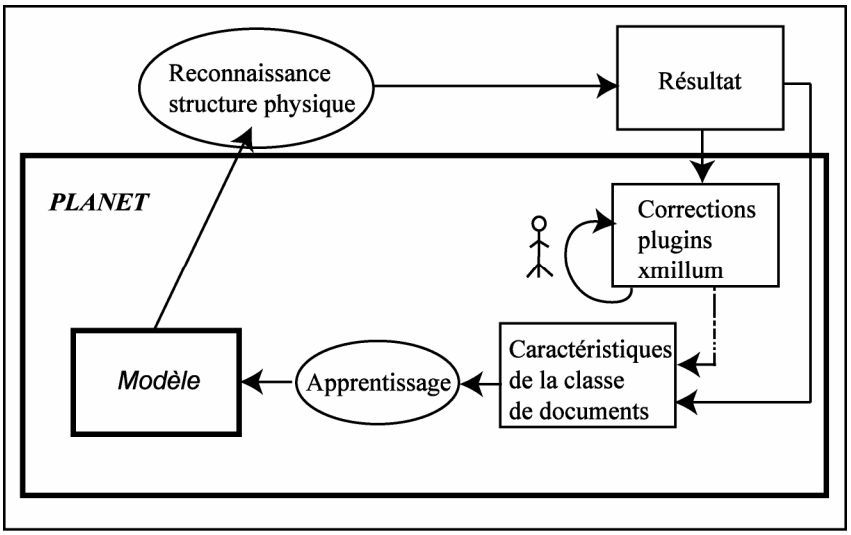


Figure 5.2 : Architecture de *PLANET*.

Dans les sous-sections suivantes, nous présentons le modèle général et les modèles dédiés.

5.2.1 Le modèle général

Le modèle général permet d'avoir une connaissance générale de la superclasse. Il est créé à partir des connaissances extraites d'un ensemble de classes de documents appartenant à cette superclasse. Le modèle général est créé, c'est-à-dire entraîné avec des échantillons obtenus à partir d'un ensemble de données segmentées par un algorithme, validés ou corrigés par un utilisateur.

Une fois l'apprentissage du modèle général effectué, la connaissance générale de chaque modèle général est copiée dans les autres modèles inclus dans la même superclasse pour leur permettre d'avoir une mémoire initiale.

La figure 5.3 illustre la construction du modèle général.

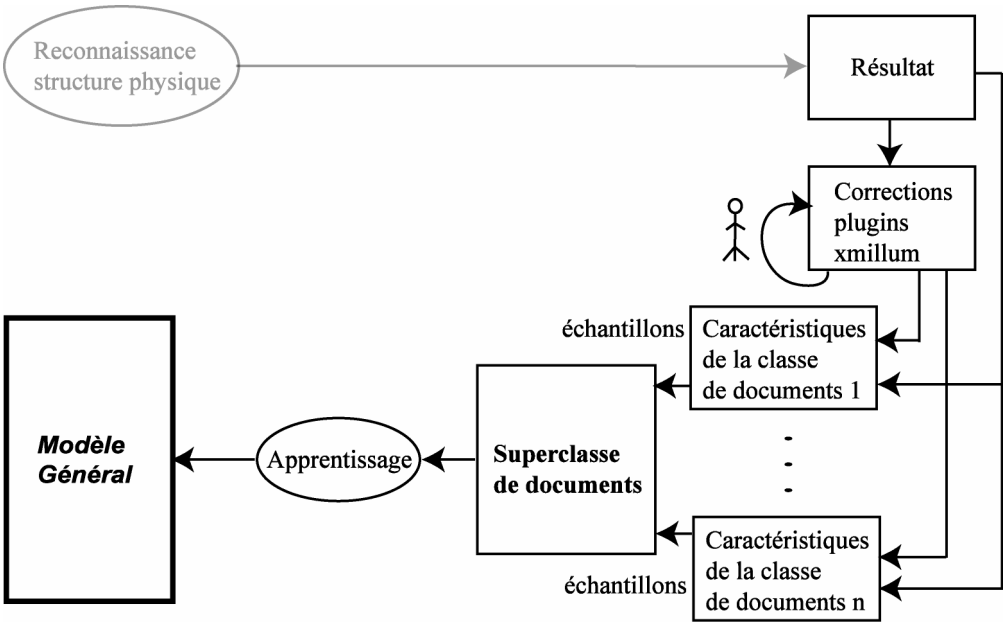


Figure 5.3 : La construction du modèle général.

L'architecture de *PLANET* est adaptable ; nous pouvons greffer plusieurs modèles généraux traitant chacun une superclasse de documents appropriés.

5.2.2 Les modèles dédiés

Les modèles dédiés représentent la connaissance propre de chaque classe de documents. Dans *PLANET* nous avons autant de modèles dédiés que de classes. Nous voulons que chaque modèle dédié se spécialise dans la reconnaissance d'une classe de documents.

On initialise un modèle dédié avec la meilleure connaissance possible, en l'occurrence avec les connaissances de la superclasse, c'est-à-dire le modèle général.

Le modèle dédié est créé à partir des connaissances extraites d'une classe de documents. Il est entraîné à partir des échantillons de la classe, plus exactement à partir d'un ensemble de données segmentées par un algorithme, validés ou corrigés par un utilisateur.

Une fois l'apprentissage effectué, le modèle dédié se spécialise pour la classe de documents à laquelle il appartient. L'apprentissage permet donc de constituer le modèle de cette classe de documents.

La figure 5.4 illustre les modèles dédiés.

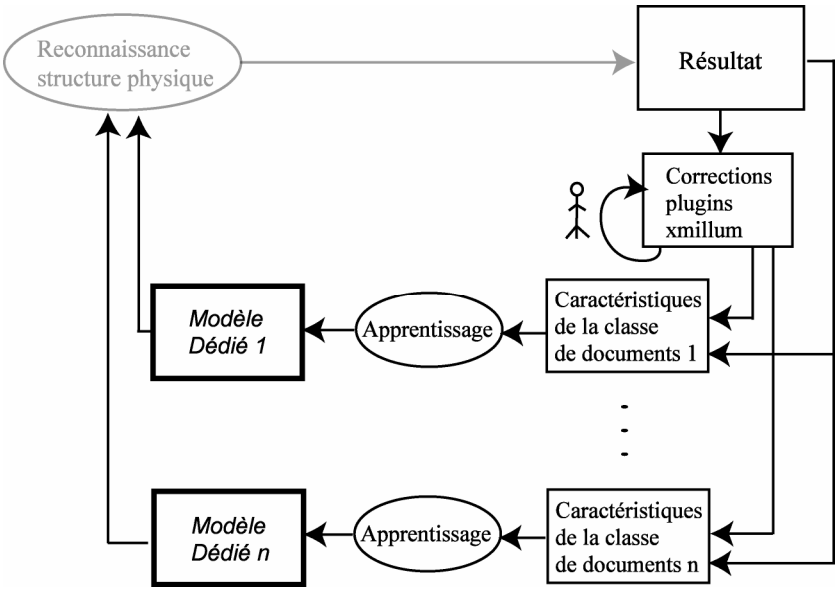


Figure 5.4 : Les modèles dédiés.

Nous pourrions particulariser encore davantage un modèle dédié en considérant un sous-ensemble de la classe de documents auquel il est rattaché. Par exemple, nous pouvons utiliser un sous-ensemble du journal ANNAHAR, contenant que les journaux ANNAHAR de l'année 2005, et lui associer un modèle dédié plus précis.

Bien que cette approche soit intéressante pour traiter certaines applications, nous ne l'avons pas utilisée dans notre évaluation.

Les modèles dédiés évoluent au cours du temps. Au début, les modèles dédiés possèdent une mémoire initiale issue du modèle général. Ensuite, pour chaque page de document traitée, des connaissances sont extraites et introduites dans le modèle dédié. Celles-ci viennent s'ajouter aux connaissances existantes du modèle. De ce fait, plus nous traitons des pages de documents, plus les modèles dédiés s'enrichissent, ce qui permet d'assurer une évolutivité. Il est à noter qu'à la longue les connaissances représentant la mémoire initiale du modèle général deviennent moins pertinentes que les connaissances ajoutées ultérieurement.

5.3 Principe de fonctionnement des RNAs

Dans cette section, nous décrivons le principe de fonctionnement des RNAs. Au début, nous définissons le neurone formel. Ensuite, nous présentons les différentes topologies des RNAs et les différents types d'apprentissage. Enfin nous décrivons les phases de reconnaissance et d'apprentissage des RNAs.

5.3.1 Le neurone formel

L'unité de base du RNA est le neurone formel tel qu'illustré dans la figure 5.5.

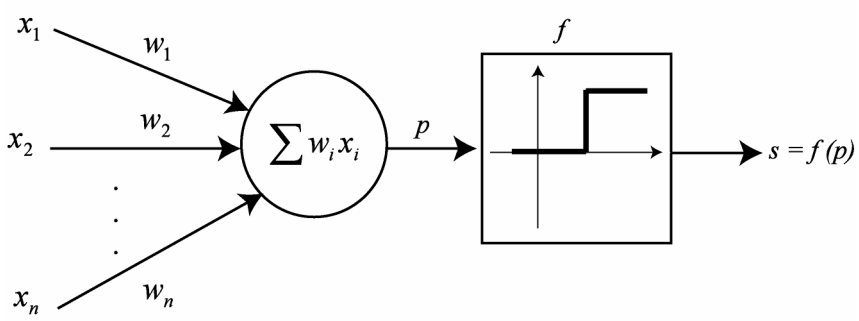


Figure 5.5 : Neurone formel.

Le neurone formel reçoit un ensemble de signaux d'entrées x_i et fournit un signal de sortie s. Ces signaux sont pondérés par des poids w_i. Il est à noter que x_i et w_i sont représentés sous forme de vecteur. Chaque neurone formel calcule son potentiel $p = \sum w_i x_i$, ensuite le signal de sortie du neurone (ou activation) $s = f(p)$ est calculée en invoquant une fonction de transfert (une fonction seuil) f qui est appliquée sur le potentiel calculé.

Dans le cas du neurone de McCulloch et Pitts [100], la fonction de transfert est une fonction binaire dont le fonctionnement est le suivant : si le potentiel est supérieur à un seuil θ alors la sortie du neurone est égale à 1, sinon elle égale à -1.

si $P > \theta$ alors $S = 1$
sinon $S = -1$

D'autres fonctions de transfert ont été introduites par la suite à savoir les fonctions linéaires, les linéaires par morceaux, les sigmoïdes, et les gaussiennes. Ces fonctions possèdent les propriétés mathématiques de dérivabilité et de continuité. La fonction de transfert sigmoïde est largement utilisée comme fonction seuil et est définie ainsi :

$$s = \frac{1}{1 + e^{-p}}$$

5.3.2 Topologies des RNAs

Un RNA est un maillage de plusieurs neurones formels. Les connexions entre les neurones qui composent un RNA décrivent la topologie. Il existe deux types de RNA : les RNA bouclés et les RNA non bouclés. Les différentes topologies des RNAs bouclés sont les suivantes [100] :

- Les réseaux à connexions récurrentes : les connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouches.
- Les réseaux à connexion complète : ces RNAs possèdent la structure d'interconnexion la plus générale. Chaque neurone est connecté à tous les neurones du réseau y inclu lui-même.

Les RNAs non bouclés sont des RNAs à couches, ils sont constitués d'un ensemble de neurones formels regroupés dans une couche. Chaque couche est reliée à une autre couche par les biais des connexions. En effet, les couches du RNA effectuent leurs traitements et transmettent le résultat de leurs analyses à la couche suivante. L'information donnée au réseau va donc se propager couche par couche, de la couche d'entrée à la couche de sortie, en passant soit par aucune, une ou plusieurs couches intermédiaires (dites couches cachées) tel qu'illustré par la figure 5.6.

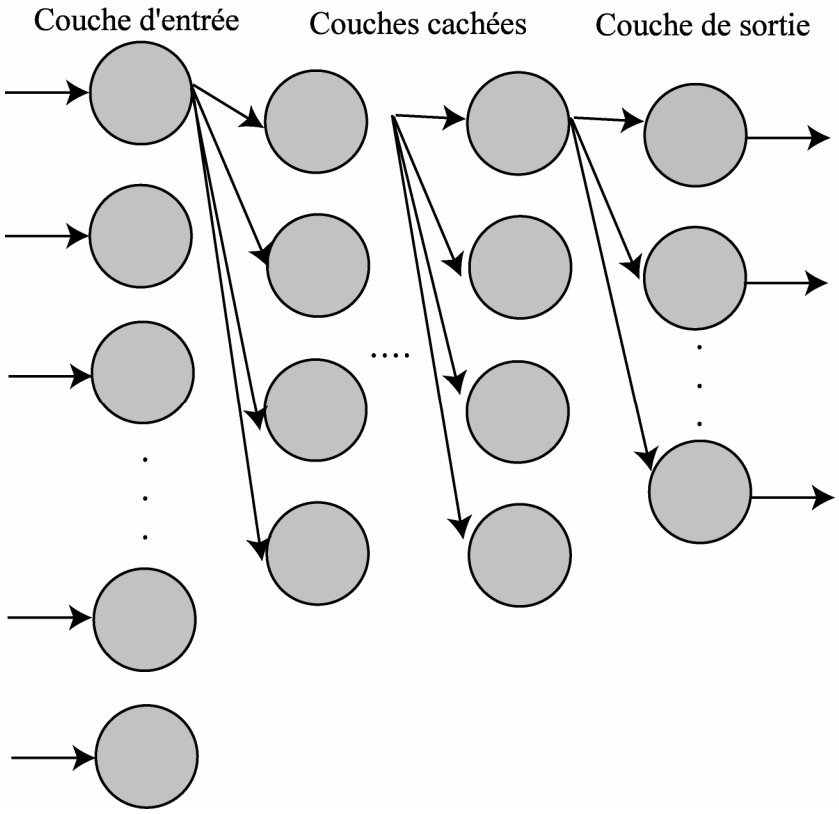


Figure 5.6 : Couches d'un RNA.

La couche d'entrée reçoit un vecteur d'entrée représentant l'entité à reconnaître, la couche cachée apprend à recoder les entrées et la couche de sortie fournit le résultat de reconnaissance. Les neurones de la couche de sortie représentent les classes de reconnaissance.

Les différentes topologies des RNAs à couches sont les suivantes [100] :

- Les réseaux multicouches ou Perceptron multicouches : dans ces RNAs, les neurones sont arrangés par couches. Il n'y a pas de connexion entre neurones d'une même couche et les connexions ne se font qu'avec les neurones des couches suivantes. Souvent, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et seulement ceux-ci.
- Les réseaux à connexions locales : ces RNAs possèdent aussi une structure multicouche. Chaque neurone est connecté avec un nombre réduit et localisé de neurones de la couche suivante. Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouches classique.

Dans cette thèse, nous utilisons les réseaux multicouches.

5.3.3 Types d'apprentissage

En plus de la description de la topologie du RNA, nous choisissions le paradigme d'apprentissage. Il existe plusieurs paradigmes qui accomplissent une phase d'apprentissage consistant en la modification des poids des connexions. Ces paradigmes sont classés en deux catégories : "apprentissage supervisé" ou "apprentissage non supervisé". Dans l'apprentissage supervisé, le RNA apprend par l'exemple, en comparant les résultats obtenus avec les sorties désirées, ce qui permet de calculer l'erreur et ensuite de réajuster les poids des connexions. C'est un apprentissage guidé, vu que le RNA est forcé à converger vers un état final précis en même temps qu'on lui présente un échantillon. Par contre, dans l'apprentissage non supervisé, le RNA est laissé libre de converger vers n'importe quel état final lorsqu'on lui présente un échantillon.

5.3.4 Phase de reconnaissance des RNAs

Supposons que les poids sont fixés, à ce moment le RNA entre en phase de reconnaissance ou d'évaluation. Cette phase permet d'une part de tester les facultés d'apprentissage du RNA sur un ensemble de données différent de celui défini lors de la phase d'apprentissage et d'autre part d'exploiter le réseau.

La figure 5.7 illustre un RNA multicouches que nous utilisons pour l'explication des phases de reconnaissance et d'apprentissage.

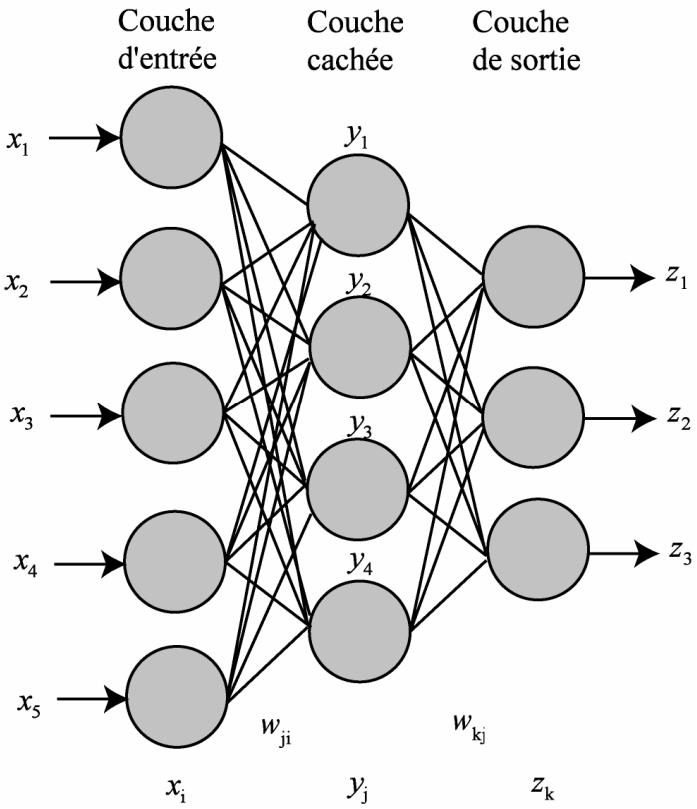


Figure 5.7 : RNA multicouches.

En phase de reconnaissance, chaque vecteur d'entrée x_i issue de l'ensemble de test est introduit au RNA. Avec les poids des signaux w_i fixés, chaque neurone d'une couche donnée, calcule son activation et la transmet comme signal d'entrée aux neurones de la couche suivante auquel il est rattaché. Nous obtenons un vecteur de sortie z_k dont ses valeurs sont comparées au vecteur de référence pour voir si l'entrée x_i est reconnue ou pas.
L'activation de chaque cellule de sortie z_k est calculée ainsi :

$$z_k = f\left(\sum_j w_{kj} y_j\right) = f\left(\sum_j w_{kj} f\left(\sum_i w_{ji} x_i\right)\right)$$

où :

- z représente la sortie calculée,
- w_{ji} représente les poids entre les neurones de la couche d'entrée x_i et les neurones de la couche cachée y_j
- w_{kj} représente les poids entre les neurones de la couche cachée y_j et les neurones de la couche de sortie z_k.
- f représente la fonction de transfert.

5.3.5 Phase d'apprentissage des RNAs

Dans cette section, nous décrivons la phase d'apprentissage des RNAs multicouches par le paradigme d'apprentissage classique de la "rétropropagation du gradient". Ce paradigme d'apprentissage supervisé consiste en une technique d'optimisation : "la descente du gradient" qui cherche à minimiser une fonction d'énergie d'erreur [100].

L'apprentissage supervisé consiste en une correction d'erreurs. Cet apprentissage se termine avec succès lorsque le RNA, pour tous les échantillons de l'ensemble d'apprentissage, donne des sorties proches des sorties désirées. Pour atteindre cet objectif on effectue une minimisation d'erreurs par itérations successives (cycles) du RNA jusqu'à atteindre une stabilité.

L'erreur d'apprentissage pour un vecteur x de l'ensemble d'entraînement est définie comme

$$E(x, \mathbf{w}) = \tfrac{1}{2} \sum_k (t_k - z_k)^2$$

Avec :
- t représente la sortie désirée,
- z représente la sortie calculée,

L'ajustement des poids du RNA s'effectue ainsi :
- pour les connexions existantes entre la couche cachée et la couche de sortie :
  $$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} = \eta \delta_k^z y_j \quad \text{avec} \quad \delta_k^z = (t_k - z_k) f'(P_k^z)$$
- pour les connexions existantes entre la couche d'entrée et la couche cachée :
  $$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} = \eta \delta_j^z x_i \quad \text{avec} \quad \delta_j^z = (t_j - z_j) f'(P_j^z)$$

Avec :
- η représente le pas de l'algorithme de la descente du gradient.
- f' représente la dérivée de la fonction de transfert.
- P représente le potentiel du neurone

Les étapes de l'apprentissage d'un RNA multicouches avec le paradigme de la rétropropagation du gradient sont les suivantes [100] :

1- Initialiser les poids aléatoirement.
2- Choisir une donnée, sous forme de vecteur, de l'ensemble d'entraînement et l'introduire au RNA dans la couche d'entrée.
3- Calculer la sortie du RNA.
4- Calculer l'erreur entre la sortie du RNA et la sortie désirée.
5- Ajuster les poids du RNA.
6- Reprendre l'étape 2 pour chaque vecteur de l'ensemble d'entraînement jusqu'à ce que l'erreur totale soit inférieure à un seuil.

5.4 Le choix des caractéristiques

Dans l'application de la fusion des lignes de texte en bloc du système de reconnaissance de la structure physique (voir chap. 4), nous avons constaté que le taux de reconnaissance pour la fusion des lignes de texte en blocs est relativement bas en raison des erreurs de sur-segmentation. De ce fait, nous avons conçu un classifieur (RNA) qui apprend à corriger ces erreurs. Il s'agit d'un petit réseau qui a l'avantage de converger plus rapidement et d'avoir un temps de réponse pour la reconnaissance le plus bas possible.

Dans cette section, nous décrivons la détermination du voisinage des blocs, ensuite, nous détaillons les caractéristiques extraites et leurs normalisations. Enfin, nous présentons la manière dont ces caractéristiques sont utilisées dans le RNA.

5.4.1 Détermination du voisinage des blocs

Pour que le système apprenne à corriger les erreurs de sur-segmentation, il doit être d'abord capable de déterminer le voisinage des blocs. Pour un bloc b_i, nous avons : les voisins supérieurs, les voisins inférieurs, les voisins de gauche et les voisins de droite. Les voisins supérieurs sont tous les voisins situés au-dessus du bloc. Le même principe est appliqué pour les autres voisins. La figure 5.8 illustre les blocs voisins du bloc b_i.

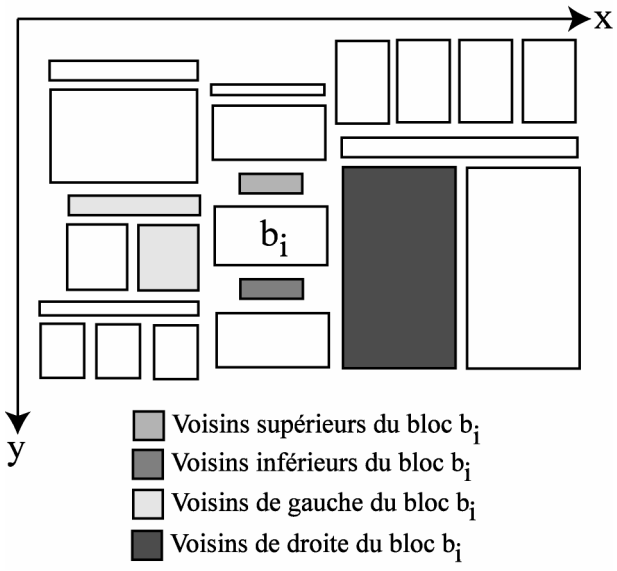


Figure 5.8 : Voisins du bloc b_i.

La détermination du voisinage repose sur le principe que nous avons utilisé pour la correction des erreurs de sur-segmentation des titres, à savoir nous construisons quatre rectangles rN, rO, rE et rS (pour rectangle nord, ouest, est et sud) à partir du rectangle principal du bloc b_i représenté par un rectangle (voir section 4.3.4). Pour chacun de ces

rectangles rN, rO, rE et rS, nous recherchons l'ensemble des rectangles qui les intersectent.

5.4.2 L'extraction des caractéristiques

Soit b_i et b_j deux blocs représentés par des rectangles. Les caractéristiques que nous avons choisies sont les suivantes :

- $l(b_i)$ la largeur du bloc b_i
- $h(b_i)$ la hauteur du bloc b_i
- bp(b_i) la densité de pixels noirs du bloc b_i
- cc(b_i) le nombre de composantes connexes incluses dans le bloc b_i ;
- d(b_i ,b_j) la distance séparant les deux blocs b_i et b_j.

La figure 5.9 illustre deux exemples de disposition de deux blocs b_i et b_j.

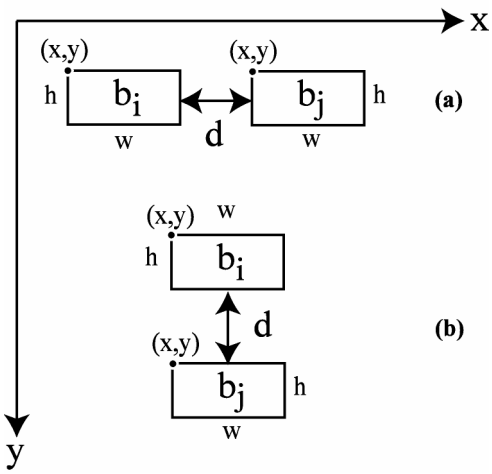


Figure 5.9 : Exemple de disposition de blocs.

La distance pour les deux exemples (a) et (b) de la figure 5.9, est calculée ainsi :
(a) $d(b_i,b_j) = b_j.x - (b_i.x + b_i.w)$
(b) $d(b_i,b_j) = b_j.y - (b_i.y + b_i.h)$

Pour fusionner deux blocs b_i et b_j, nous avons besoin de neuf caractéristiques ; quatre pour le premier bloc b_i, quatre pour le deuxième bloc b_j et une distance qui est commune aux deux blocs. Ces caractéristiques sont données en entrée au RNA.

5.4.3 La normalisation des caractéristiques

Les caractéristiques ne sont pas exploitables directement par un RNA, une étape de normalisation est requise. En effet, la normalisation est essentielle pour les RNAs multicouches car elle permet de maintenir les poids du réseau dans des intervalles relativement restreints, d'optimiser les conditions d'apprentissage et par la même occasion d'améliorer la convergence. Étant donné que le calcul des caractéristiques "densité de pixels noirs" donne des valeurs normalisées, nous avons normalisé, entre 0 et 1, les caractéristiques "hauteur", "largeur du bloc", "distance entre les deux blocs" et "nombre de composantes connexes" de la manière suivante :

- Pour les caractéristiques "hauteur" et "largeur", nous avons déterminé sur tous les documents de toutes les classes, une hauteur et une largeur maximales des blocs. Ensuite, nous divisons chaque caractéristique "hauteur" et "largeur" d'un bloc par les valeurs maximales respectives pour obtenir des mesures comprises entre 0 et 1.
- Pour la caractéristique "distance", nous déterminons sur tous les documents de toutes les classes la valeur maximale de la distance séparant deux blocs voisins dans les quatre directions (nord, sud, est et ouest). Par la suite, chaque caractéristique distance inter-bloc est divisée par la valeur maximale trouvée.
- Pour la caractéristique "nombre de composantes connexes", nous avons déterminé sur tous les documents de toutes les classes le nombre de composantes connexes d'un document. Ensuite, la caractéristique "nombre de composantes" connexes d'un bloc de texte est divisée par cette valeur maximale.

5.4.4 Utilisation des caractéristiques dans les RNAs

Pour l'apprentissage des RNAs, nous procédons comme suit :

- L'ensemble d'apprentissage est composé des données d'entrée et des valeurs de sortie correspondantes désirées. Les données d'entrée sont composées des caractéristiques d'une paire de blocs. Soit VD (valeurs désirées), l'ensemble des valeurs de sortie désirées constitué des valeurs 0 et 1. La valeur 1 désigne les blocs à fusionner et 0 indique les blocs découpés et les blocs corrects.

  $VD = \{ 0, 1 \}$

  Soit B, l'ensemble des blocs segmentés.

  $B = \{ b_1, b_2, b_3, \ldots, b_n \}$

  Il est à noter qu'un bloc b_i peut être correctement segmenté ou non correctement segmenté. Soit Γ l'ensemble des blocs correctement segmentés, et soit Ψ l'ensemble des blocs sur-segmentés. Tout bloc $b_i \in \Gamma \cup \psi$;

- Nous constituons les couples d'apprentissage composés :
  - des blocs correctement segmentés et découpés avec la valeur désirée 0, d'où : $((b_i, b_j), 0)$.
  - des blocs sur-segmentés avec la valeur désirée 1, d'où : $((b_k, b_l), 1)$

  avec b_i et $b_j \in B$.

  Les paires sont choisies par l'utilisateur lors de la correction (voir section 3.4.1.2). Il est à noter que les blocs correctement segmentés, sont construits, une fois les corrections de l'utilisateur sont achevées. De l'ensemble B, nous déduisons l'ensemble des blocs corrigés par l'utilisateur et pour chaque bloc correctement segmentés nous déterminons ses voisins tels que défini dans la section 5.4.1.

- Ces couples d'apprentissage sont passés au RNA : (b_i, b_j) et (b_k, b_l). En effet, ce sont les caractéristiques extraites de ces couples qui sont introduites dans le RNA.
- Donc à un couple de blocs donné (b_k, b_l), correspond une entrée dans le RNA

En phase de reconnaissance, l'ensemble de test est composé uniquement des caractéristiques des données d'entrée. Les paires de blocs sont choisies de la manière suivante :

- Pour chaque bloc b_i, nous déterminons ses blocs voisins.
- De chaque voisin du bloc b_i, nous constituons les paires de blocs et nous introduisons les caractéristiques de ces paires au RNA.

Dans la section suivante, nous présentons les topologies des RNAs.

5.5 Topologies des RNAs de *PLANET*

La topologie des RNAs utilisés dans *PLANET* est la suivante :

- la couche d'entrée possède neuf neurones qui correspondent aux caractéristiques extraites.
- la couche cachée. Il n'y a pas de règle absolue qui permet de déterminer avec exactitude le nombre de neurones à utiliser dans cette couche. De ce fait, nous avons effectué plusieurs tests (7, 8, 9, 10, 11 neurones) et nous avons trouvé expérimentalement que dix neurones dans la couche cachée étaient suffisants.
- la couche de sortie est composée d'un seul neurone de sortie statuant si le bloc doit être fusionné ou conservé tel quel.

Etant donné que les RNAs sont entièrement connectés, le nombre total de connexions pour chaque RNA est de 100 (9 x 10 + 10 x 1).

La topologie du RNA pour *PLANET* est représentée dans la figure 5.10.

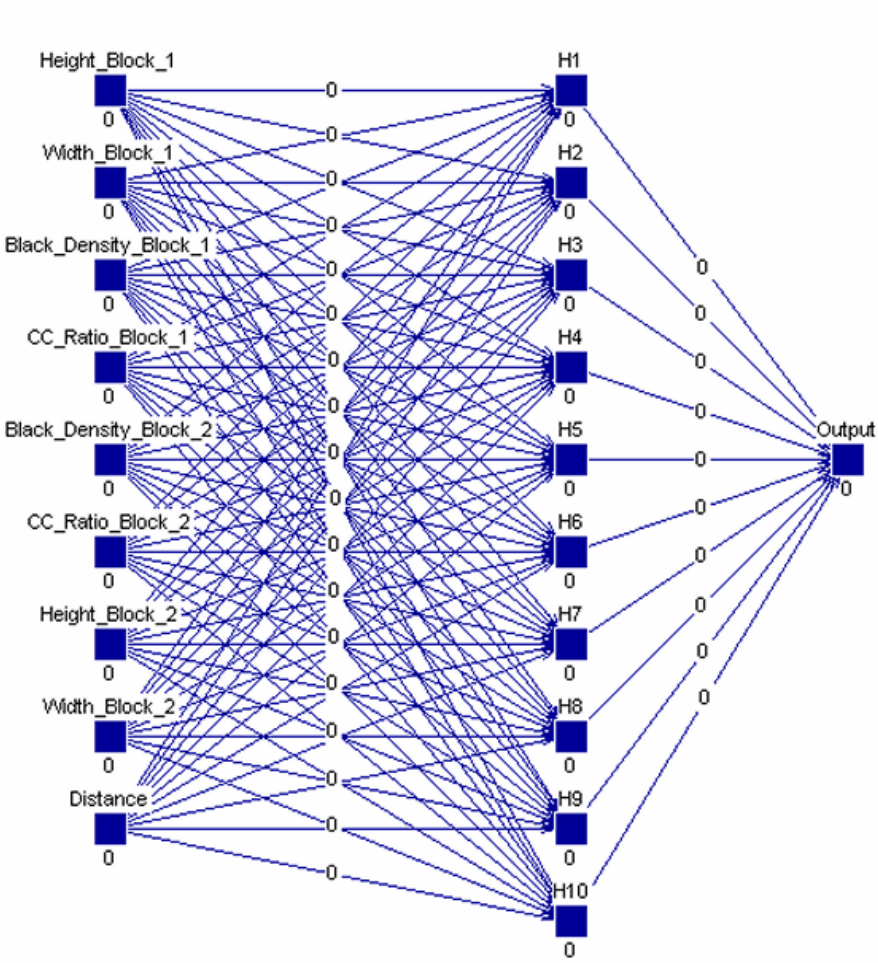


Figure 5.10 : Topologie du RNA

De nos jours, il existe plusieurs simulateurs de réseaux de neurones artificiels sous différentes formes :

- Des outils. Ils nécessitent une application mère. Par exemple, la boite à outils de réseaux de neurones pour les logiciels de calcul mathématique nécessite les logiciels Matlab ou Mathematica.
- Des librairies dans un langage de programmation donné. Par exemple, la librairie SNNS pour Stuttgart Neural Network Simulator est écrite en langage C.
- Des programmes exécutables, comme par exemple JavaNNS, développé par l'université de Tübingen [54], et doté d'une interface graphique dont le noyau est basé sur la librairie SNNS.

Notre choix s'est porté sur JavaNNS pour sa gratuité, le nombre important de paradigmes d'apprentissage supporté, la richesse de ses outils et sa convivialité.

5.5.1 Paramètres d'apprentissage pour les RNAs

Les paramètres d'apprentissage pour les RNAs des différents modèles sont les suivants :

- La valeur du pas η de l'algorithme de la descente du gradient définie dans les deux dernières équations du paragraphe 5.4.5, l a été fixé à 0.2. Il est à noter que le choix d'une valeur trop grande du pas risque de faire diverger le processus de détection du minimum optimal, et inversement, une valeur du pas trop petite augmente les temps de calcul ;
- Le nombre d'itérations a été fixé à 5000.

L'initialisation du RNA se fait avec des poids aléatoires ayant des valeurs comprises entre -1.0 et +1.0. La fonction de transfert que nous avons utilisée est la fonction logistique standard (la sigmoïde). Afin que le RNA ne soit pas dépendant d'un ordre d'apprentissage arbitraire, nous avons mélangé l'ordre des patterns de l'ensemble d'entraînement.

Avec une configuration actuelle (Pentium 4 2GHz), les temps d'apprentissage sont de l'ordre de la minute.

5.6 Évaluation de *PLANET*

Dans cette section, nous passons en revue la procédure d'évaluation de *PLANET* et les résultats obtenus avec les classes de documents que nous avons choisies.

Le nombre de classes de documents utilisés dans l'évaluation s'élève à trois. Ces classes correspondent aux unes des journaux ANNAHAR, AL HAYAT et AL QUDS, la superclasse étant celle des journaux arabes. L'ordre d'évaluation est le suivant, nous considérons d'abord le modèle général, puis chaque modèle dédié est évalué séparément. Dans les sous-sections suivantes, nous détaillons la phase d'évaluation pour le modèle général et les modèles dédiés. Ensuite, nous présentons l'évaluation croisée des modèles.

5.6.1 Le modèle général

L'ensemble d'apprentissage du RNA du modèle général est construit à partir de cinq documents issus de chacune des classes dont la segmentation a été validée ou corrigée par l'utilisateur.

Nous avons évalué les performances du modèle général. L'ensemble de test est composé de 30 documents répartis comme suit : 10 images pour le journal ANNAHAR, 10 images pour le journal AL HAYAT et 10 images pour le journal AL QUDS.

Afin de pouvoir évaluer le résultat de reconnaissance, nous devons utiliser un fonds de vérité de l'image du document. Ce fonds de vérité nous permet de le comparer au résultat de reconnaissance obtenu par le RNA du modèle général. La construction du fonds de vérité est effectuée d'une manière interactive et comprend les blocs validés ou corrigés par l'utilisateur. Nous avons calculé le taux de reconnaissance moyen pour les trois classes de documents, ensuite nous avons comparé ce taux avec celui obtenu pour le

système de reconnaissance de la structure physique décrit dans le chapitre 4. La table 5.1 illustre cette comparaison.

%	ANNAHAR	AL HAYAT	AL QUDS
Taux de reconnaissance moyen du système de reconnaissance de la structure physique	95.319	91.437	91.033
Taux de reconnaissance moyen du modèle général	**96.424**	**96.931**	**96.832**

Table 5.1 : Comparaison des résultats obtenus par le modèle général avec ceux obtenus par le système de reconnaissance.

En comparant ces taux avec ceux obtenus par notre système de reconnaissance de la structure physique pour la reconnaissance d'images de journaux en arabe, nous constatons une amélioration.

5.6.2 Les modèles dédiés

Nous avons effectué une expérience similaire en utilisant les modèles dédiés. Pour ce faire, nous particularisons les RNAs des modèles dédiés à partir des connaissances du modèle général, par une spécialisation de ce dernier. Cette démarche consiste à démarrer l'étape d'apprentissage du RNA de chaque classe de documents avec les connaissances du modèle général.

L'ensemble d'entraînement de chaque modèle dédié est construit à partir des données issues des images de documents segmentées par le système de reconnaissance et validées ou corrigées par l'utilisateur. L'ensemble de test est composé de 69 documents répartis comme suit : 35 images pour le journal ANNAHAR, 10 images pour le journal AL HAYAT et 24 images pour le journal AL QUDS.

Le processus d'entraînement de chaque modèle dédié se déroule comme dans l'expérience précédente :

1. Construction d'une suite aléatoire de pages de documents de l'ensemble d'entraînement. La succession aléatoire de pages de documents permet de garantir l'indépendance du RNA de l'ordre chronologique des « unes des journaux ».
2. Entraînement du RNA sur les documents de l'ensemble d'entraînement selon la chronologie de la suite établie lors de l'étape 1.

Les pages utilisées lors de l'entraînement pour les trois classes de documents possèdent une grande variabilité.

Pour le RNA du modèle dédié ANNAHAR, nous avons construit deux suites aléatoires auxquelles nous avons appliqué la procédure d'entraînement décrite ci-dessus. La figure 5.11 illustre l'évolution de la performance de reconnaissance du modèle dédié ANNAHAR avec les deux ensembles de pages aléatoires.

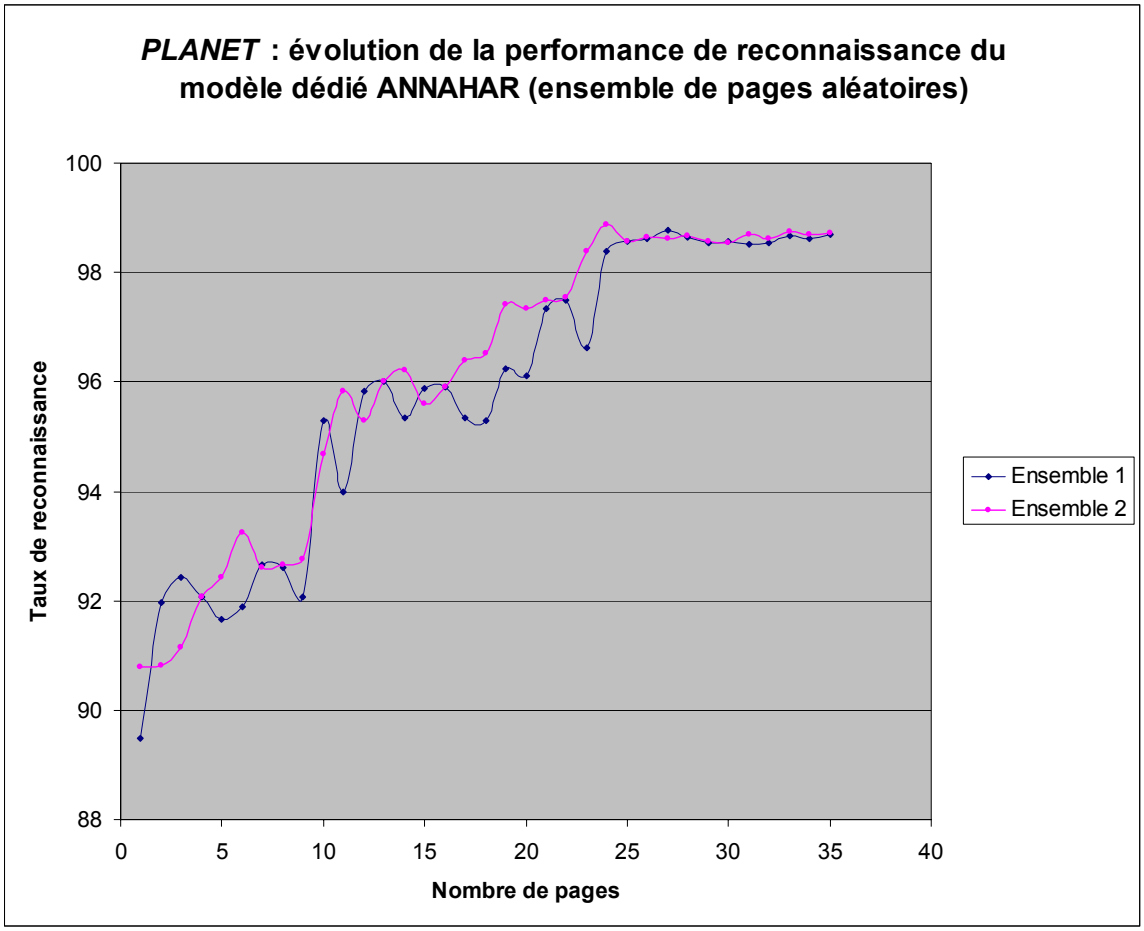


Figure 5.11 : Évolution de la performance de reconnaissance du modèle dédié ANNAHAR.

Dans la figure 5.11, nous constatons une convergence du RNA du modèle dédié ANNAHAR à partir de la 25ème page pour les deux ensembles. Les deux apprentissages ont un comportement similaire.

Pour le RNA du modèle dédié AL QUDS, nous avons construit une seule suite aléatoire de pages du journal AL QUDS et nous avons appliqué la procédure d'entraînement décrite plus haut. La figure 5.12 illustre l'évolution de la performance de reconnaissance du modèle dédié ALQUDS avec la suite de pages aléatoires.

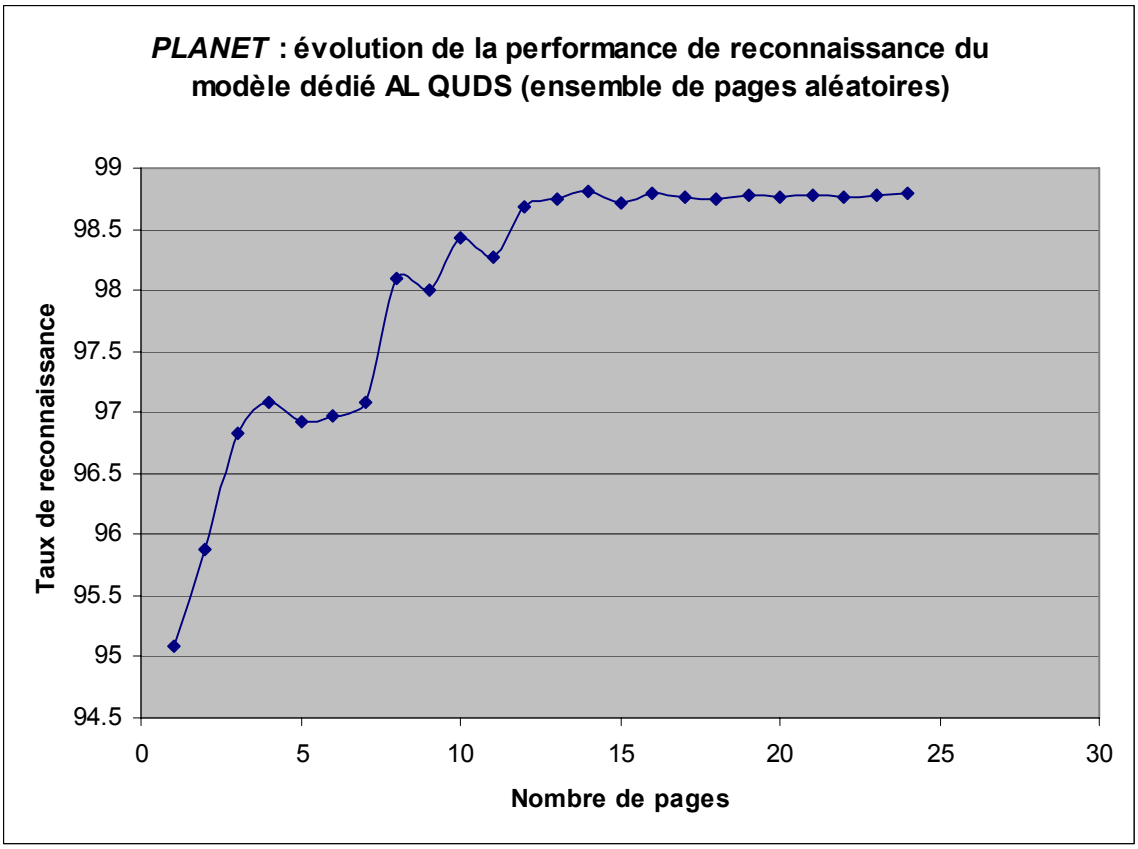


Figure 5.12 : Évolution de la performance de reconnaissance du modèle dédié AL QUDS.

Dans la figure 5.12, nous constatons une convergence du RNA du modèle dédié AL QUDS à partir de la 14ème page.

Nous avons voulu connaître les raisons pour lesquelles le RNA du modèle dédié AL QUDS converge plus rapidement que celui du journal ANNAHAR. AL QUDS contrairement à ANNAHAR, contient peu de publicité ; de ce fait chaque page contient davantage de blocs. La figure 5.13 illustre un exemple de page du journal ANNAHAR comportant la publicité.

Figure 5.13 : Exemple de page du journal ANNAHAR comportant la publicité.

Une fois les RNAs des trois modèles entraînés, nous procédons à leurs évaluations et ce en testant les performances avec un ensemble de test. Ce dernier est le même que nous avons utilisé pour l'évaluation du modèle général (voir section 5.6.1).

Pour l'évaluation des trois modèles dédiés, nous adoptons la même démarche que celle que nous avons décrite dans la section précédente pour l'évaluation du modèle général.

Nous avons aussi voulu comparer le taux de reconnaissance moyen obtenu par le modèle général avec celui obtenu par les trois modèles dédiés. La table 5.2 illustre cette comparaison ainsi que le facteur de réduction de l'erreur.

%	ANNAHAR	AL HAYAT	AL QUDS
Taux de reconnaissance moyen du modèle général	96.424	96.931	96.832
Taux de reconnaissance moyen des modèles dédiés	**98.789**	**98.343**	**98.931**
Facteur de réduction de l'erreur	66.135	46.008	66.256

Table 5.2 : Comparaison des résultats obtenus par le modèle général avec ceux obtenus par les modèles dédiés.

Le facteur de réduction de l'erreur (FRE) est calculé ainsi :

$$FRE = \frac{NombreErreursModGénéral - NombreErreursModDédié}{NombreErreursModGénéral} * 100$$

En comparant le taux obtenu par le modèle général avec celui obtenu par les modèles dédiés, nous constatons une amélioration due à l'introduction des données d'apprentissage propres à chaque modèle dédié.

5.6.3 L'évaluation croisée

Dans *PLANET*, chaque modèle dédié est entraîné avec les données d'apprentissage issues de la classe de documents auquel il est rattaché. Nous avons voulu étudier le comportement de ces modèles dédiés en les croisant.

Ce croisement nous permet d'évaluer les modèles dédiés avec des classes de documents autres que celles avec lesquelles ils ont été entraînés. Il s'agit par exemple, d'évaluer l'ensemble test du modèle dédié AL HAYAT avec le RNA du modèle dédié ANNAHAR et inversement. Nous effectuons les mêmes tests pour les autres modèles dédiés AL HAYAT et AL QUDS.

La table 5.3 illustre le taux de reconnaissance croisé pour les trois modèles dédiés et comparaison par rapport au modèle général.

Modèle dédié évalué / Ensemble test du modèle dédié	ANNAHAR	AL HAYAT	AL QUDS
ANNAHAR	**98.789**	98.163	97.846
AL HAYAT	92.157	**98.343**	96.886
AL QUDS	96.371	98.081	**98.931**
Modèle général	*96.424*	*96.931*	*96.832*

Table 5.3 : *PLANET* : Résultats de la reconnaissance croisée pour les trois modèles dédiés.

En analysant les résultats obtenus dans la table 5.3, nous constatons que les RNAs des modèles dédiés ANNAHAR, AL HAYAT et AL QUDS sont devenus spécialisés et que le taux de reconnaissance décroît quand le modèle dédié est évalué sur un ensemble de test auquel il n'a pas été confronté dans le protocole d'apprentissage. Nous remarquons aussi que chaque modèle dédié connaît ses caractéristiques propres. La comparaison de ces résultats avec ceux obtenus avec le modèle général, nous a permis de noter que l'évaluation croisée est meilleure que celle du modèle général. Ceci peut s'expliquer par le nombre restreint d'échantillons utilisés dans le modèle général par rapport aux modèles dédiés.

5.7 Conclusion

Dans ce chapitre, nous avons présenté un système de reconnaissance évolutif pour la reconnaissance des structures physiques. Ce système appelé *PLANET*, est initialisé avec un modèle général, supposé contenir les meilleures connaissances possibles pour traiter le problème quelle que soit la classe de documents considérée. Lorsque le système est utilisé pour traiter un ensemble de documents appartenant à une classe restreinte, le modèle s'adapte automatiquement à partir des corrections effectuées par l'utilisateur.

Nous avons décrit l'application du modèle général et des modèles dédiés de *PLANET* à la reconnaissance d'images de documents ("les journaux arabes"), la démarche d'évaluation et les résultats obtenus.

L'analyse des résultats obtenus par *PLANET* montre une amélioration du taux de reconnaissance par rapport à l'algorithme de fusion développé au sein de notre système général de reconnaissance d'images de documents.

PLANET est une méthode généralisable à d'autres applications de la reconnaissance, comme par exemple la reconnaissance de la structure logique qui ne nécessite aucun changement au niveau de l'architecture. A cet effet, nous avons conçu *LUNET*, un système de reconnaissance de structures logiques doté d'apprentissage évolutif basé sur les réseaux de neurones artificiels qui est présenté dans le chapitre suivant.

Chapitre 6

LUNET : système de reconnaissance de structures logiques doté d'apprentissage évolutif basé sur les réseaux de neurones artificiels.

Ce chapitre traite de la reconnaissance de structures logiques. La construction d'un système automatique de reconnaissance de structures logiques, qui donne de bons résultats sur des documents quelconques, est très difficile en raison de la variabilité de la complexité des documents et de l'application à traiter. Par contre, il existe des systèmes de reconnaissance de structures logiques qui appliqués à un ensemble restreint de documents semblables donnent relativement de bons résultats.

Pour qu'un système de reconnaissance donne de bons résultats sur un ensemble varié de documents, il faut qu'il puisse s'adapter à l'ensemble des documents qu'il doit traiter. Comme nous l'avons déjà mentionné dans le chapitre précédent, l'amélioration des résultats de reconnaissance demeure possible en ayant recours à une reconnaissance assistée dotée d'un apprentissage évolutif. Nous avons adopté de nouveau la même démarche pour la reconnaissance de la structure logique. Le système de reconnaissance de structures logiques de classes de documents, doté d'apprentissage évolutif basé sur les réseaux de neurones artificiels, s'intitule *LUNET* (Logical labeling of classes of documents Using artificial neural NETs) [36].

Dans ce chapitre nous présentons tout d'abord les principes de construction de *LUNET* ainsi que les modèles qui le composent. Ensuite, nous décrivons les caractéristiques utilisées et la topologie des RNAs de *LUNET*. Enfin, nous présentons la démarche pour l'évaluation de *LUNET* et les résultats obtenus.

6.1 Les modèles de *LUNET*

Dans cette section, nous traitons les modèles de reconnaissance.

La figure 6.1 illustre l'architecture de *LUNET* qui utilise plusieurs modèles. Il est à noter que nous utilisons dans *LUNET* les mêmes concepts que ceux définis dans *PLANET* à savoir le modèle général et les modèles dédiés (voir section 5.2.1 et 5.2.2). Le modèle général permet d'avoir une connaissance générale d'une superclasse. A chaque classe de

documents est associée un modèle, dit *dédié*, dont le but est de représenter des caractéristiques de cette classe.

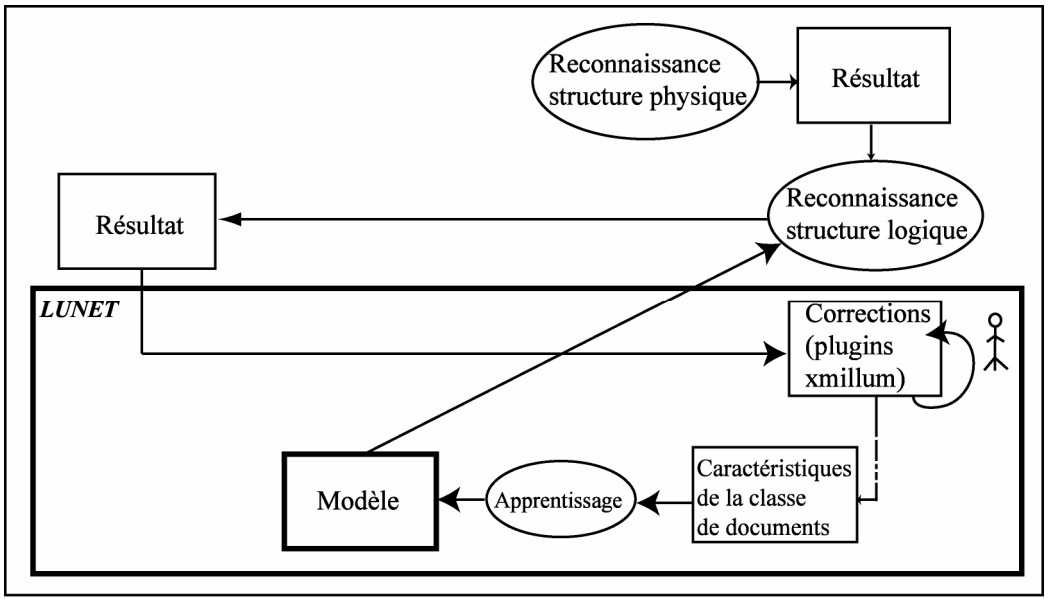


Figure 6.1 : Architecture de *LUNET*.

L'architecture de *LUNET* permet la reconnaissance de la structure logique de classes de documents.

6.2 Les classes logiques

Après avoir étudié les différentes classes de documents ANNAHAR, AL HAYAT et AL QUDS, nous nous sommes rendu compte que l'étiquetage logique peut être effectué à travers les classes logiques suivantes :

- "Titre" désigne le titre d'un article.
- "Auteur" désigne l'auteur de l'article.
- "Texte de base" désigne un bloc de texte.
- "Ancre Vers" désigne une référence à la suite de l'article.
- "Légende" désigne l'illustration d'une image.

Ces classes logiques permettent d'avoir une structure logique de ce type de documents.

La figure 6.2 illustre les différentes classes logiques.

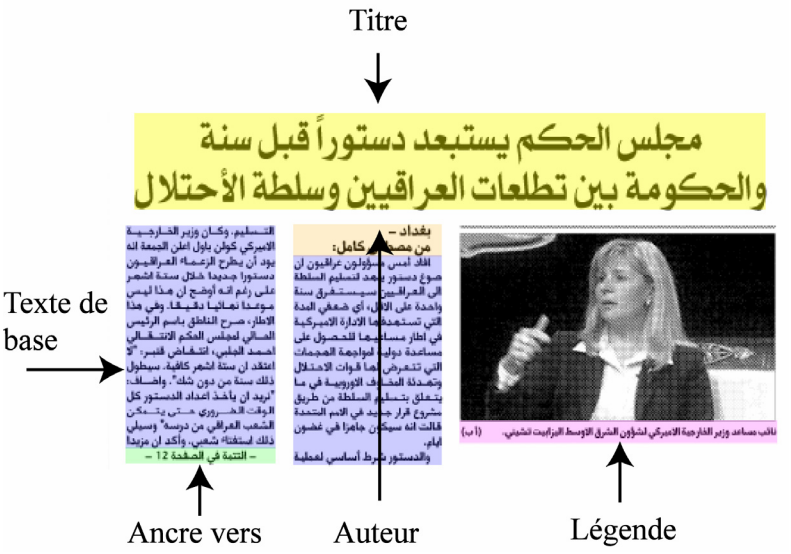


Figure 6.2 : Les différentes classes logiques.

Vu que cet étiquetage logique s'effectue sur les blocs de texte correctement segmentés issus de *PLANET*, certains blocs de texte doivent être découpés en sous blocs pour permettre à l'utilisateur de les étiquetés. Pour effectuer cette tâche, l'utilisateur a recours au module de correction pour découper le bloc (voir section 3.4.1.2). Par exemple, un bloc de texte peut être découpé en deux blocs, le premier étiqueté avec l'étiquette "texte de base" et le deuxième étiqueté avec l'étiquette "ancre vers" tel qu'illustré dans la figure 6.3.

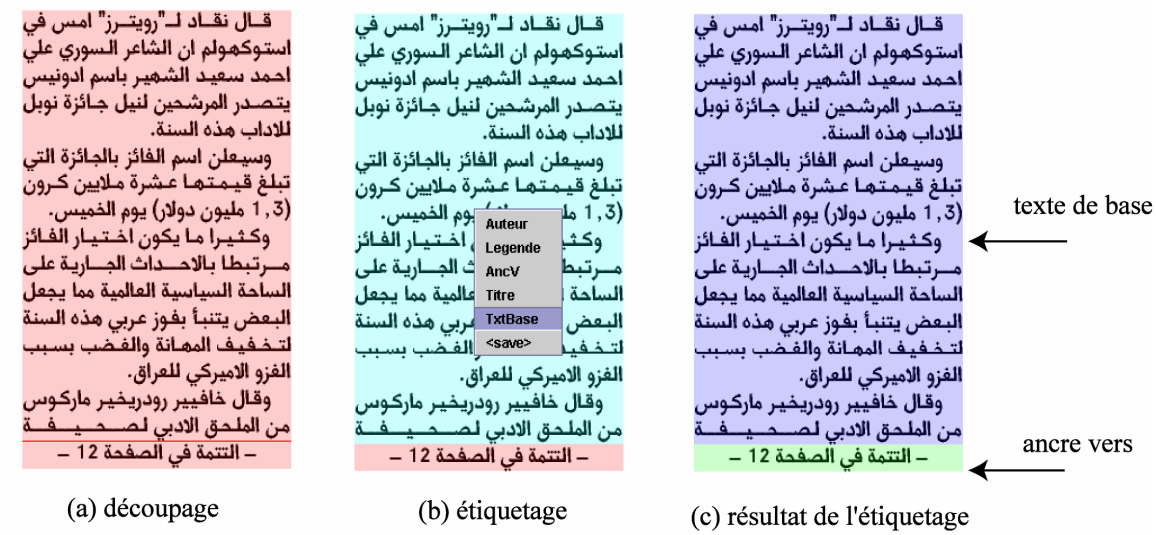


Figure 6.3 : Découpage et étiquetage des blocs de texte.

6.3 Les caractéristiques

Pour permettre aux modèles de *LUNET* d'effectuer l'étiquetage logique, nous avons conçu un classifieur (RNA) qui apprend à étiqueter les blocs de texte issus de *PLANET*. Il s'agit d'un petit réseau qui à l'avantage de converger rapidement et d'avoir un temps de réponse pour la reconnaissance le plus bas possible.

Dans cette section, nous détaillons les caractéristiques extraites, leurs normalisations et la manière dont ces caractéristiques sont utilisées dans le RNA.

6.3.1 L'extraction des caractéristiques

Soit b_i un bloc représenté par un rectangle. Les caractéristiques que nous avons choisies sont les suivantes :

- $l(b_i)$ la largeur du bloc b_i ;
- $h(b_i)$ la hauteur du bloc b_i ;
- bp(b_i) la densité de pixels noirs du bloc b_i ;
- cc(b_i) le nombre de composantes connexes incluses dans le bloc b_i ;
- $l(b_i)/h(b_i)$ le rapport largeur sur la hauteur du bloc b_i.

Ces caractéristiques sont données en entrée au RNA.

6.3.2 La normalisation des caractéristiques

Cependant, avant leurs introductions dans le RNA, une normalisation de ces caractéristiques est requise. Étant donné que les caractéristiques "hauteur", "largeur du bloc", et "nombre de composantes connexes" et "densité de pixels noirs" sont similaires à celles utilisés dans *PLANET*, nous appliquons la même démarche de normalisation (voir section 5.4.3). La normalisation de la caractéristique "hauteur/largeur" s'effectue en divisant la hauteur normalisée par la largeur normalisée.

6.3.3 Utilisation des caractéristiques dans les RNAs

Pour l'apprentissage des RNAs, nous procédons comme suit :

- L'ensemble d'apprentissage est composé des données d'entrées et des valeurs de sorties correspondantes désirées. Les données d'entrées sont composées des caractéristiques d'un bloc. Soit ^{VD} (valeurs désirées), l'ensemble des valeurs de sorties désirées constitué des classes logiques :

VD = {Titre, Auteur, Texte de base, Ancre Vers, Légende}
Donc à un bloc donné b_i , correspond une entrée dans le RNA.

En phase de reconnaissance, l'ensemble de test est composé uniquement des caractéristiques du bloc sans les valeurs désirées.

Dans la section suivante, nous présentons les topologies des RNAs.

6.4 Topologie des RNAs de *LUNET*

La topologie des RNAs utilisés dans *LUNET* est la suivante :

- la couche d'entrée possède cinq neurones qui correspondent aux caractéristiques extraites ;
- la couche cachée. Il n'y a pas de règle absolue pour déterminer le nombre de neurones dans cette couche cachée. De ce fait, nous avons effectué plusieurs tests avec 6, 7, 8, 9 neurones et nous avons trouvé expérimentalement que huit neurones dans la couche cachée étaient suffisants ;
- la couche de sortie est composée de cinq neurones qui correspondent aux étiquettes "Titre", "Auteur", "Texte de base", "Ancre Vers", "Légende".

Etant donné que les RNAs sont entièrement connectés, le nombre total de connexions pour chacun de ces RNA est de 80 (5 x 8 + 8 x 5).

L'architecture du RNA de *LUNET* est représentée dans la figure 6.4

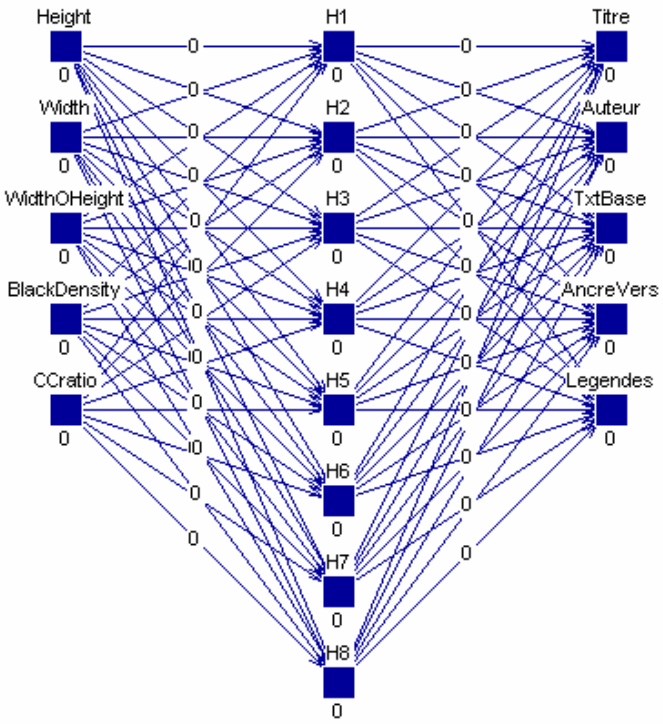


Figure 6.4 : Architecture du RNA

Les différents paramètres pour l'apprentissage du RNA sont les mêmes que ceux définis dans *PLANET* (voir section 5.5.1).

6.5 Évaluation de *LUNET*

Dans cette section nous décrivons la procédure d'évaluation de *LUNET* et les résultats obtenus.
Les classes de documents et l'ordre d'évaluation sont les mêmes que ceux défini dans *PLANET*.
Dans les sous-sections suivantes, nous détaillons la phase d'apprentissage et d'évaluation pour le modèle général et les modèles dédiés. Ensuite, nous présentons l'évaluation croisée des modèles.

6.5.1 Le modèle général

L'ensemble d'apprentissage du RNA du modèle général comprend des échantillons de blocs étiquetés par l'utilisateur provenant de cinq images de documents issues de chaque classe de documents.

Nous avons évalué les performances du modèle général. L'ensemble de test est composé de 18 images de documents repartis comme suit : 6 pour ANNAHAR, 5 pour AL HAYAT et 7 pour AL QUDS.

Afin de pouvoir évaluer le résultat de reconnaissance, nous devons utiliser un fonds de vérité de l'image de document. La construction de ce fonds de vérité est effectuée de manière interactive par l'utilisateur et elle comprend l'étiquetage des blocs issus de *PLANET*. Nous avons calculé le taux de reconnaissance moyen pour les trois classes de documents.
Dans la table 6.1 nous dressons le taux de reconnaissance moyen obtenu par les trois classes de documents pour le modèle général pour les étiquettes : "Titre", "Auteur", "Texte de base", "Ancre vers" et "Légende".

%	ANNAHAR	AL HAYAT	AL QUDS	*Moyenne*
Titre	82.143	98.182	97.114	92.479
Auteur	69.444	55.833	72.823	66.033
Texte de base	93.869	96.687	95.985	95.513
Ancre vers	67.063	58.615	75.000	66.892
Légende	61.111	80.000	69.048	70.053
Moyenne	80.662	82.978	90.779	84.806

Table 6.1 : Taux de reconnaissance moyen obtenu par les trois classes de documents pour le modèle général.

Le faible taux de reconnaissance pour l'étiquette "Auteur" pour les trois classes de documents s'explique par le fait du manque d'échantillons parmi les images de documents sélectionnés.

6.5.2 Les modèles dédiés

La particularisation des modèles dédiés de *LUNET* à partir du modèle général se fait d'une manière analogue à celle de *PLANET*. Cette particularisation consiste à démarrer l'étape d'apprentissage du RNA de chaque classe de documents avec les connaissances du modèle général.

L'ensemble d'entraînement de chaque modèle dédié est construit à partir des caractéristiques extraites des blocs de texte étiquetés. L'ensemble de test est composé de 18 documents répartis comme suit : 6 images pour le journal ANNAHAR, 5 images pour le journal AL HAYAT et 7 images pour le journal AL QUDS.

Le processus d'entraînement de chaque modèle dédié se déroule d'une manière similaire que celle décrite pour le modèle dédié de *PLANET*. Les dates des pages utilisées lors de l'entraînement pour les trois classes de documents possèdent une grande variabilité. Pour le RNA du modèle dédié ANNAHAR, nous avons construit une suite aléatoire à laquelle nous avons appliqué la procédure d'entraînement.
La figure 6.5 illustre, pour chaque étiquette, l'évolution de la performance de reconnaissance du modèle dédié ANNAHAR avec l'ensemble des pages aléatoires.

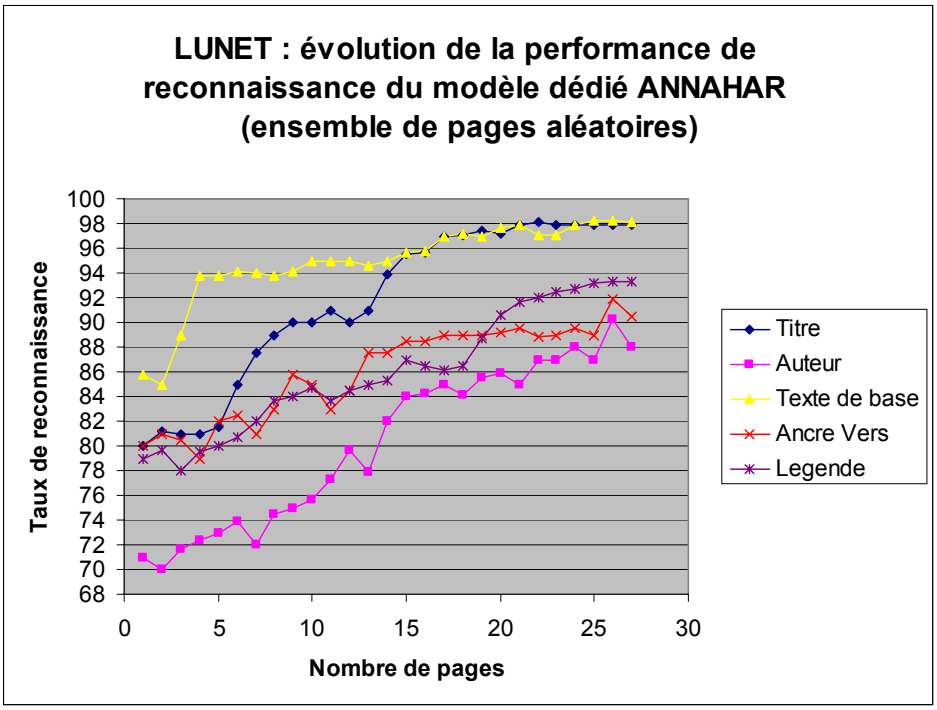


Figure 6.5 : évolution pour chaque étiquette de la performance de reconnaissance du modèle dédié ANNAHAR.

D'après la figure 6.5, nous constatons une convergence pour les étiquettes "Titre" et "Texte de base". En revanche, pour les autres étiquettes, la convergence n'est pas encore atteinte.

Nous avons voulu savoir l'évolution de la performance de reconnaissance pour l'ensemble des étiquettes. De ce fait, nous avons calculé la moyenne du taux de reconnaissance pour l'ensemble des étiquettes par rapport au nombre de pages. La figure 6.6 illustre l'évolution de la performance du modèle dédié ANNAHAR pour l'ensemble des étiquettes avec un ensemble aléatoire.

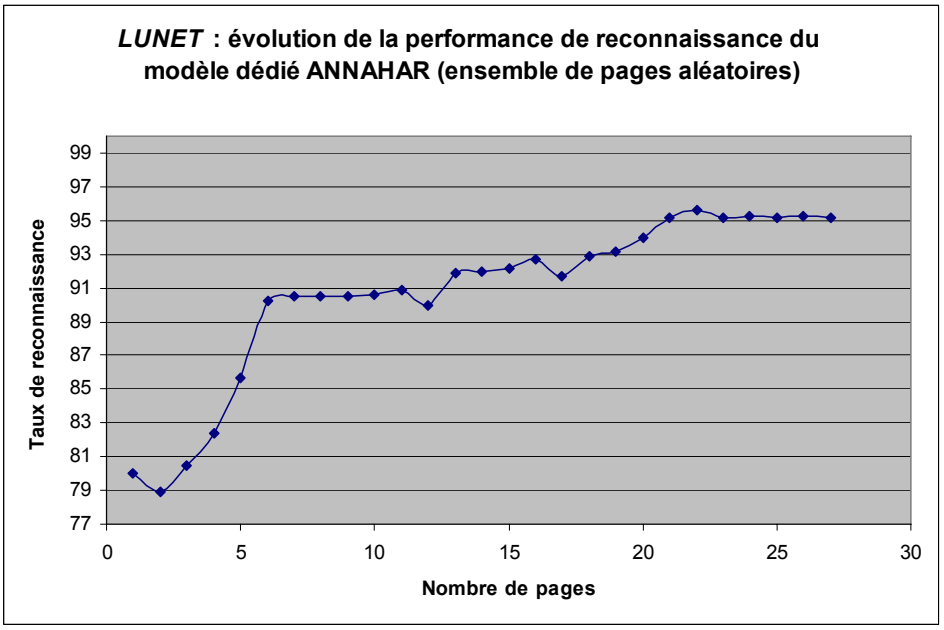


Figure 6.6 : évolution de la performance de reconnaissance du modèle dédié ANNAHAR

D'après la figure 6.6, nous constatons une convergence du RNA du modèle dédié ANNAHAR à partir de la 22ème page.

Pour le RNA du modèle dédié AL QUDS, nous avons construit une suite aléatoire de pages du journal AL QUDS et nous avons appliqué la procédure d'entraînement.

La figure 6.7 illustre, pour chaque étiquette, l'évolution de la performance de reconnaissance du modèle dédié ALQUDS avec la suite de pages aléatoires.

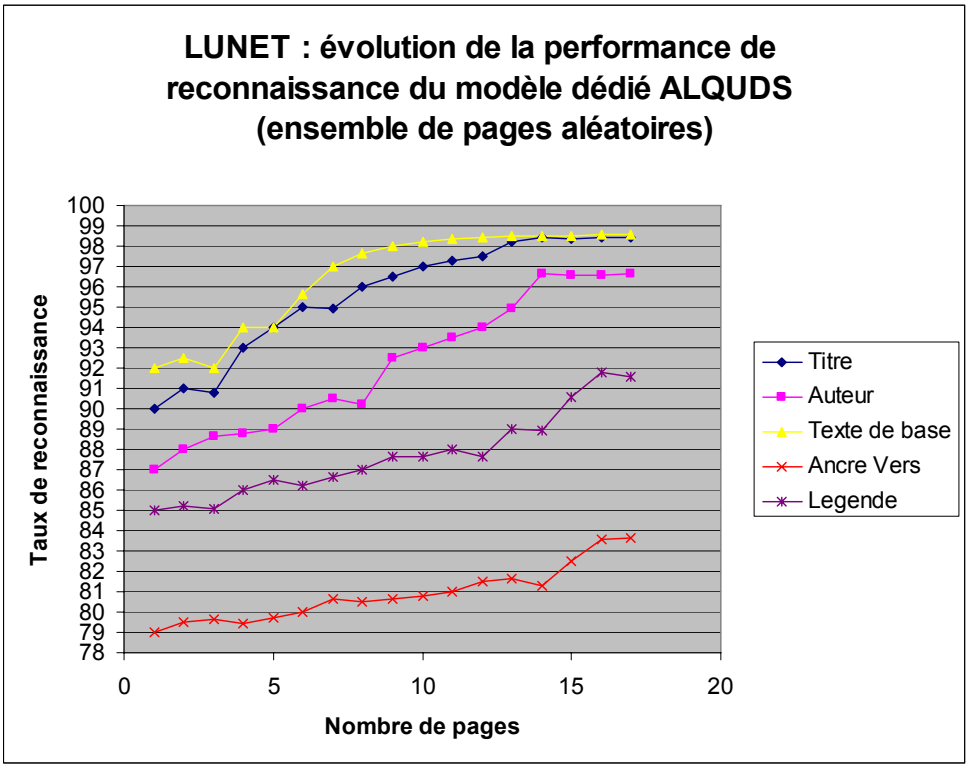


Figure 6.7 : évolution pour chaque étiquette de la performance de reconnaissance du modèle dédié AL QUDS

D'après la figure 6.7, nous constatons une convergence pour les étiquettes "Titre" "Auteur" et "Texte de base". En revanche, pour les autres étiquettes, la convergence n'est pas encore atteinte.

Nous avons voulu savoir l'évolution de la performance de reconnaissance pour l'ensemble des étiquettes. De ce fait, nous avons calculé la moyenne du taux de reconnaissance pour l'ensemble des étiquettes par rapport au nombre de pages.

La figure 6.8 illustre l'évolution de la performance du modèle dédié ALQUDS pour l'ensemble des étiquettes avec un ensemble aléatoire.

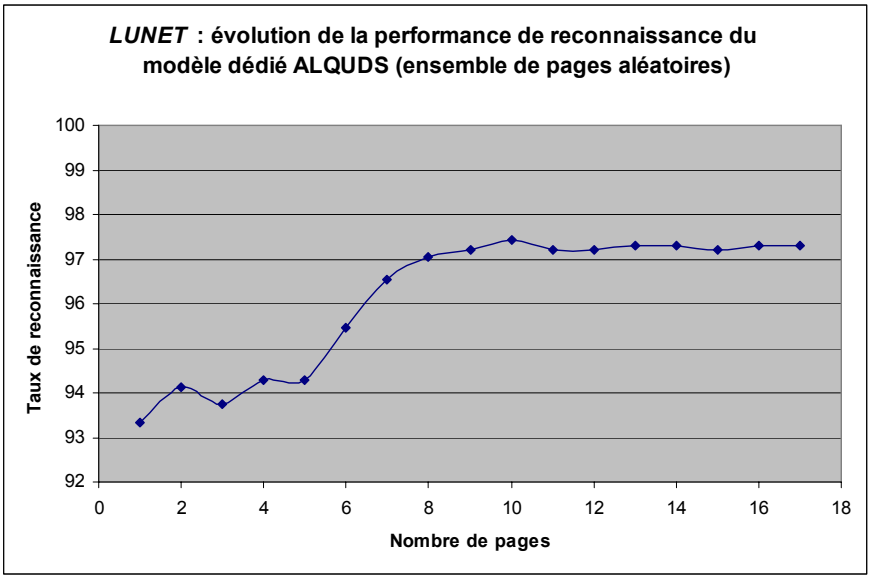


Figure 6.8 : évolution de la performance de reconnaissance du modèle dédié AL QUDS

D'après la figure 6.8, nous constatons une convergence du RNA du modèle dédié ALQUDS à partir de la 11ème page.

Nous avons voulu savoir les raisons pour lesquelles le RNA du modèle dédié AL QUDS converge plus rapidement que celui du journal ANNAHAR. AL QUDS contrairement à ANNAHAR, contient peu de publicité ; de ce fait chaque page contient davantage de blocs.

Pour l'évaluation des trois modèles dédiés, nous adoptons la même démarche que celle que nous avons décrite dans la section précédente pour l'évaluation du modèle général.
Dans la table 6.2 nous dressons le taux de reconnaissance moyen obtenu par les trois classes de documents pour chaque modèle dédié.

%	ANNAHAR	AL HAYAT	AL QUDS	*Moyenne*
Titre	97.917	100.000	98.485	98.800
Auteur	90.278	57.611	96.667	81.518
Texte de base	98.052	97.937	98.589	98.192
Ancre vers	91.905	93.102	83.333	89.446
Légende	93.333	90.000	91.667	91.666
Moyenne	95.408	90.087	97.445	94.313

Table 6.2 : Taux de reconnaissance moyen obtenue par les modèles dédiés pour les différentes étiquettes.

D'après la table 6.2, nous constatons que le taux de reconnaissance pour la classe "Auteur" du modèle dédié AL HAYAT est relativement bas, ceci est dû au manque de données d'apprentissage.

Nous avons constaté que le RNA du modèle dédié ANNAHAR confond souvent entre les deux classes "Auteur" et "Ancre vers", ceci est dû au fait que visuellement ces deux classes sont assez proches tel qu'illustré dans la figure 6.9.

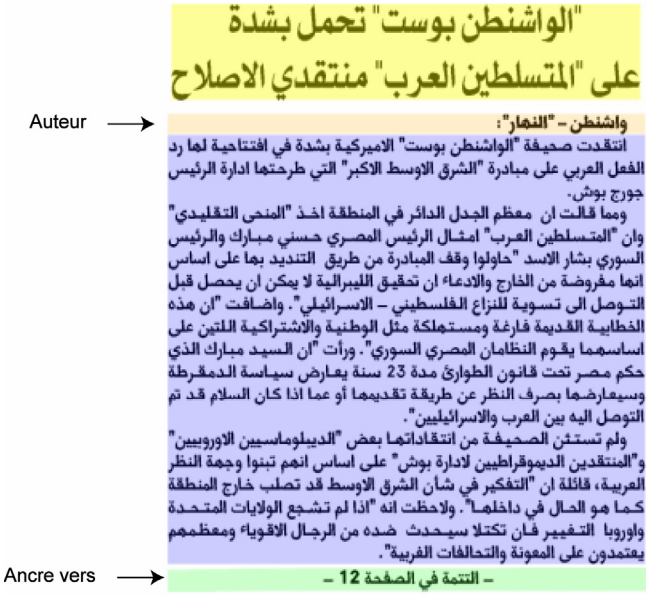


Figure 6.9 : Exemple de la classe "auteur" et de la classe "ancre vers" du journal ANNAHAR

La comparaison du taux de reconnaissance obtenu par le modèle général avec celui obtenu par les trois modèles dédiés est décrite dans la table 6.3.

%	ANNAHAR	AL HAYAT	AL QUDS
Taux de reconnaissance moyen du modèle général	80.662	82.978	90.779
Taux de reconnaissance moyen des modèles dédiés	**95.408**	**90.087**	**97.445**
Facteur de réduction de l'erreur	76.254	41.763	72.291

Table 6.3 : Comparaison des résultats de reconnaissance obtenus par le modèle général avec ceux obtenus par les modèles dédiés.

D'après la table 6.3, nous constatons une amélioration du taux de reconnaissance moyen des modèles dédiés par rapport à celui du modèle général.

6.5.3 L'évaluation croisée

Nous avons étudié aussi le comportement des modèles dédiés de *LUNET* en les croisant.

Les tables 6.4 et 6.5 illustrent le taux de reconnaissance croisé pour les trois modèles dédiés.

Modèle dédié évalué / Ensemble test du modèle dédié		**ANNAHAR**	**AL HAYAT**	**AL QUDS**
ANNAHAR	Titre	**97.917**	96.364	85.640
	Auteur	**90.278**	35.889	60.918
	Texte de base	**98.052**	95.873	93.333
	Ancre Vers	**91.905**	69.957	66.667
	Légende	**93.333**	80.000	61.905
AL HAYAT	Titre	89.484	**100.000**	92.947
	Auteur	48.611	**57.611**	41.020
	Texte de base	95.385	**97.937**	92.419
	Ancre Vers	60.635	**93.102**	58.333
	Légende	71.111	**90.000**	76.190
AL QUDS	Titre	85.913	87.823	**98.485**
	Auteur	59.722	61.661	**96.667**
	Texte de base	93.437	95.734	**98.589**
	Ancre Vers	67.381	72.958	**83.333**
	Légende	72.778	80.000	**91.667**
Modèle Général	Titre	82.143	98.182	97.114
	Auteur	69.444	55.833	72.823
	Texte de base	93.869	96.687	95.985
	Ancre Vers	67.063	58.615	75.000
	Légende	61.111	80.000	69.048

Table 6.4 : *LUNET* : Résultats de la reconnaissance croisée pour les trois modèles dédiés pour chaque étiquette.

Modèle dédié évalué / Ensemble test du modèle dédié	ANNAHAR	AL HAYAT	AL QUDS
ANNAHAR	**95.408**	80.772	84.617
AL HAYAT	82.247	**90.087**	83.459
AL QUDS	81.627	83.785	**97.445**
Modèle Général	*80.662*	*82.978*	*90.779*

Table 6.5 : *LUNET* : Moyenne de la reconnaissance croisée pour les trois modèles dédiés.

D'après les tables 6.5 et 6.6, nous remarquons que les modèles dédiés, appliqués aux autres classes sont moins performants et souvent même pire que le modèle général. Les RNAs des modèles dédiés ANNAHAR, AL HAYAT et AL QUDS de *LUNET* sont devenus spécialisés.

6.6 Conclusion

Dans ce chapitre, nous avons présenté un système de reconnaissance évolutif pour la reconnaissance de structures logiques. Ce système, intitulé *LUNET*, est initialisé, d'une manière analogue à *PLANET*, avec les meilleures connaissances possibles pour traiter le problème quelle que soit la classe de documents considérée. Lorsque le système est utilisé pour traiter un ensemble de documents appartenant à une classe restreinte, le modèle s'adapte automatiquement à partir de l'étiquetage effectué par l'utilisateur sur les *fonds de vérité* obtenus par *PLANET*.

Nous avons décrit l'application du modèle général et des modèles dédiés de *LUNET* à la reconnaissance de la structure logique d'images de documents, la démarche d'évaluation et les résultats obtenus.

L'analyse des résultats obtenus par *LUNET* montre qu'ils sont bons et ils permettent entre autres de valider l'architecture proposée par *PLANET*. *LUNET* a montré la faisabilité de l'extensibilité de *PLANET* dans le sens nous pouvons l'appliquer à d'autres applications de la reconnaissance et ceci sans rien changer au niveau de l'architecture. Une extension possible de *LUNET* serait la reconstitution de l'ordre de lecture.

Chapitre 7

Conclusion

Dans ce chapitre nous allons dresser le bilan du travail effectué. La première section récapitule les objectifs que nous nous sommes fixés, les étapes de notre approche méthodologique ainsi que le bilan de notre travail dans le domaine de la reconnaissance d'images de documents. La section suivante inventorie les extensions en vue d'un possible prolongement.

7.1 Résumé des contributions

L'objectif de cette thèse est d'étudier l'évolutivité des modèles dans un contexte interactif pour la reconnaissance de structures physiques et logiques de documents riches en structures et en variabilité.

La littérature scientifique dans le domaine de la reconnaissance d'images de documents regorge de travaux portant sur la reconnaissance de structures physiques et logiques. Néanmoins, l'attention des chercheurs de la communauté du document est quasi-inexistante pour la reconnaissance de documents arabes. Ceci s'explique par la difficulté de la langue mais aussi du manque d'engouement de la part des chercheurs.

Notre approche a été la suivante : dans un premier temps, nous avons adapté les algorithmes de reconnaissance de structures physiques d'images de documents de la langue latine à la langue arabe pour les documents riches en structures et en variabilité. Cette adaptation a été effectuée en tenant compte des spécificités de la langue arabe : écriture de droite à gauche, points diacritiques et doublement de lettres. Nous avons travaillé de telle manière que cette adaptation ne soit pas dépendante d'une classe de documents particulière. Le système de reconnaissance de structures physiques développé a été évalué sur les cinq tâches décrites dans le chapitre 4. Ces cinq tâches ont été évaluées sur des exemplaires de trois classes de documents de type journal.

Ensuite nous avons enrichi le système de reconnaissance par un apprentissage évolutif en traduisant les actions de l'utilisateur, par le biais de l'interactivité, en connaissances. Le système développé est ouvert, il permet de traiter les classes de documents. L'interactivité et l'apprentissage sont deux caractéristiques intégrantes des deux systèmes à apprentissage évolutif que nous avons développé :

- *PLANET* pour la reconnaissance des structures physiques,
- *LUNET* pour la reconnaissance des structures logiques.

Les deux systèmes *PLANET* et *LUNET* sont composés de modèles évolutifs qui sont utilisés dans un contexte interactif. Chaque modèle : général et dédié est spécialisé pour traiter respectivement une superclasse et une classe de documents. *PLANET* a été évalué sur la fusion des lignes de texte en blocs en considérant le modèle général, puis les trois modèles dédiés. *LUNET* a été appliqué à une tâche : l'étiquetage logique.

Les modèles des deux systèmes *PLANET* et *LUNET* ont permis de réduire le taux d'erreurs, en moyenne de 53% par rapport au modèle général. Avec *PLANET* nous avons atteint 98% de taux de reconnaissance moyen pour les trois classes de documents. Pour *LUNET* ce taux est de 94%.

Afin de faciliter l'échange des résultats de la reconnaissance des structures physiques et logiques et de favoriser la création des fonds de vérité, les résultats sont représentés en XML. Pour la visualisation de ces résultats, nous avons utilisé *xmillum* en développant des plugins et en définissant des feuilles de style XSLT. Celles-ci définissent comment les résultats de reconnaissance sont affichés et comment l'utilisateur peut interagir pour corriger ou valider les erreurs.

7.2 Extensions envisagées

Diverses extensions pourraient être envisagées pour améliorer *PLANET* et *LUNET*. Dans les sous-sections suivantes nous passons en revue ces différentes propositions.

7.2.1 Choix des caractéristiques

Dans les chapitres 5 et 6 nous avons décrit les caractéristiques extraites de l'entité bloc de texte nécessaires pour la reconnaissance des structures physiques et logiques. Le choix actuel des caractéristiques a abouti à de bons résultats, cependant il est clair que l'ajout d'autres caractéristiques permettrait d'améliorer le taux de reconnaissance. Il est à noter qu'avec un nombre assez grand de caractéristiques, le temps de réponse du système devient grand. On a choisit le compromis suivant : faire de telle sorte que le système donne un bon taux de reconnaissance avec un temps de réponse adéquat.

7.2.2 Mise à jour des connaissances des modèles dédiés

Dans *PLANET* et *LUNET* les modèles dédiés évoluent au cours du temps. Des connaissances sont ajoutées aux modèles dédiés pour les enrichir. Une extension possible serait de développer un outil de mise à jour des connaissances en supprimant des exemples des modèles dédiés. Cet outil permettrait de supprimer des connaissances des modèles dédiés. Une des manières à procéder serait de supprimer des connaissances aléatoirement, ceci afin de ne pas dégrader les performances du système, en éliminant par exemple les anciennes connaissances qui peuvent être pertinentes pour le modèle.

7.2.3 Test avec d'autres applications

Notre système de reconnaissance a été évalué sur cinq tâches ; séparation texte image, extraction des filets, extraction des cadres, extraction des lignes de texte et fusion des lignes de texte en blocs. *PLANET* a été évalué avec la fusion des lignes de texte en blocs. Dans le cadre d'une éventuelle extension de *PLANET*, les autres tâches : séparation texte image, extraction des filets, extraction des cadres et extraction des lignes de texte pourraient être évaluées.

Par ailleurs, une extension importante de *LUNET* serait la reconstitution de l'ordre de lecture.

7.2.4 Autres documents et autres langues

Nos systèmes de reconnaissance de structures physiques et logiques, *PLANET* et *LUNET* ont été implémentés et évalués sur une superclasse de documents : les documents riches en structures et en variabilité à savoir les journaux en langue arabe. Une extension possible serait d'évaluer nos systèmes de reconnaissance de structures physiques et logiques avec d'autres documents et avec d'autres langues telles que par exemple l'hébreu, les langues cyrilliques ou les langues asiatiques.

7.2.5 Réversibilité des opérations de l'utilisateur

Jusqu'à présent lors des opérations de corrections et d'étiquetage, l'utilisateur possède une certaine liberté d'action. En effet, si l'utilisateur effectue une erreur dans le choix de l'opération de correction ou dans le choix de l'entité à corriger, cette erreur est répercutée dans *PLANET / LUNET* sans possibilité ultérieure de la supprimer.

Afin de remédier à ce problème, le développement d'un module d'annulation des opérations de correction et d'étiquetage effectués par l'utilisateur empêcherait ces opérations erronées d'être enregistrées d'une manière définitive.

7.3 Conclusion finale

Dans cette thèse, nous avons étudié l'évolutivité des modèles dans un contexte interactif pour la reconnaissance de structures physiques et logiques de documents riches en structures et en variabilité.

La création de fonds de vérité est une opération fastidieuse, coûteuse et ne peut pas se faire d'une manière automatique. Grâce à *PLANET / LUNET* le processus de construction de fonds de vérité est simplifié.

La création de plusieurs fonds de vérité avec des outils interactifs dotés d'apprentissage évolutif tel que ceux que nous avons développé (*PLANET / LUNET*) mais aussi leurs enrichissements au fil du temps permettra de rendre un jour la reconnaissance de documents un problème déjà résolu. En effet, nous pensons que l'enrichissement des fonds de vérité par de la sémantique, par exemples des connaissances métiers, est nécessaire pour une meilleure catégorisation des classes de documents.

L'introduction de deux niveaux de modèles : le modèle général et les modèles dédiés dans les deux systèmes *PLANET* et *LUNET*, a permis une meilleure catégorisation des classes de documents et une meilleure spécialisation des modèles.

Les architectures ouvertes des deux systèmes à base de modèles, permettent une adaptation facile en vue de traiter d'autres thèmes de la reconnaissance de documents.

Les systèmes dotés d'apprentissage évolutif à l'instar de *PLANET* et de *LUNET*, permettent de traiter avec succès les applications de reconnaissance de taille moyenne. En effet, ces systèmes sont plus adéquats pour la reconnaissance de documents riches en structures et en variabilité. Cependant, *PLANET* et *LUNET* excelleraient pour les documents à structures simples.

Ainsi, nous estimons avoir démontré la pertinence des systèmes *PLANET* et *LUNET* pour la reconnaissance de structures physiques et logiques de documents riches en structures et en variabilité.

Nous espérons que l'étude que nous avons effectuée permettra une avancée aussi relative soit-elle dans le domaine de la reconnaissance de documents.

Bibliographie

[1] Adobe Systems Incorporated. “Adobe Portable Document Format Reference”, http://partners.adobe.com/asn/tech/pdf/specifications.jsp.

[2] Adobe Systems Incorporated. “PostScript Language”, Addison-Wesley, 1986.

[3] Adobe Systems Incorporated. “Tagged Image File Format Specification Revision 6.0”, http://partners.adobe.com/asn/tech/tiff/specification.jsp.

[4] O. T. Akindele and A. Belaid. “Page Segmentation by Segment Tracing”, Proceedings of the 2nd International Conference on Document Analysis and Recognition, Tsukuba, Japan, 1993, pp. 341-344.

[5] O. T. Akindele and A. Belaid. “Construction of Generic Models of Document Structures using Inference of Tree Grammars”, Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, Canada, 1995, pp. 206-209.

[6] A. Amin. “Off-Line Arabic Character Recognition: The State Of The Art”, Pattern Recognition, Vol. 31, No. 5, 1998, pp. 517-530

[7] T. Andersen and W. Zhang. “Features for Neural Net Based Region Identification of Newspaper Documents”. Proceedings of the 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, August 2003, pp. 403-407.

[8] A. Antonacopoulos. “Page Segmentation Using the Description of the Background”, Computer Vision and Image Understanding, volume 70, No 3, 1998, pp. 350-369.

[9] A. Antonacopoulos and F.P. Coenen. “Region Description and Comparative Analysis Using a Tesseral Representation”. Proceedings of the 5th International Conference on Document Analysis and Recognition, Bangalore, India, September 1999, pp. 193-196.

[10] A. Antonacopoulos and H. Meng. “A Ground-Truthing Tool for Layout Analysis Performance Evaluation”, Proceedings of the 5th International Workshop on Document Analysis Systems, DAS2002, Princeton, USA, August 2002, pp. 236-244.

[11] A. Antonacopoulos and R.T. Ritchings. “Flexible Page Segmentation Using the Background”, Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem, Israel, October 1994, pp. 339-344.

[12] A. Azokly. “Une approche générique pour la reconnaissance de la structure physique de documents composites”. PhD thesis, IIUF-University of Fribourg, 1995.

[13] H. S. Baird. “Background structure in document images”. In H. Bunke, P. S. P. Wang, & H. S. Baird (Eds.), Document Image Analysis, World Scientific, Singapore, 1994, pp. 17-34.

[14] H. S. Baird, S. E. Jones and S. J. Fortune. “Image Segmentation by Shape-Directed Covers”. Proceedings of the 10th International Conference on Pattern Recognition, Atlantic City, NJ, June 1990, pp. 820-825.

[15] F. Bapst. “Reconnaissance de documents assistée : architecture logicielle et intégration de savoir-faire”, PhD thesis, IIUF-University of Fribourg, Fribourg, Switzerland 1998.

[16] F. Bapst, R. Brugger. A. Zramdini and R. Ingold. “Integrated Multi-Agent Architecture for Assisted Document Recognition”, Proceedings of the 2th International Workshop on Document Analysis Systems, DAS1996, Pennsylvania, USA, October 1996, pp. 172-188.

[17] F. Bapspt, R. Brugger, A. Zramdini and R. Ingold. “L’intégration des données dans un système de reconnaissance de documents assistée”. CNED1996, Nantes, France.

[18] F. Bapspt and R. Ingold. “Using Typography in Document Image Analysis”. In R. D. Hersch, J. Andre and H. Brown, editors, Electronic Publishing Artistic Imaging and Digital Typography, number 1375 in Lecture notes in computer science, St-Malo, France, March 1998, pp. 240-251.

[19] F. Bapst, A. Zramdini and R. Ingold. “A Scenario Model Advocating User-driven Adaptive Recognition Systems”, Proceedings of the 4th International Conference on Document Analysis and Recognition, Ulm, Germany, August 1997, pp. 745-748.

[20] N. E. Ben Amara. “Sur la problématique et les orientations en reconnaissance de l’écriture arabe”, Proceedings of the Colloque International Francophone sur l’Ecrit et le Document, CIFED’2002, Hammamet, Tunisia, October 2002, pp. 2-10.

[21] T.M. Bruel. “Two Geometric Algorithms for Layout Analysis”, Proceedings of the 5th International Workshop on Document Analysis Systems, DAS2002, Princeton, USA, August 2002, pp. 188-199.

[22] R. Brugger, F. Bapst and R. Ingold. “A DTD Extension for Document Structure Recognition”, In R. D. Hersch, J. Andre and H. Brown, editors, Electronic Publishing Artistic Imaging and Digital Typography, number 1375 in Lecture notes in computer science, St-Malo, France, March 1998, pp. 343-354.

[23] R. Brugger, A. Zramdini and R. Ingold. “Modeling Documents for Structure Recognition Using Generalized n-grams”. Proceedings of the 4th International Conference on Document Analysis and Recognition, Ulm, Germany, August 1997, pp. 56-60.

[24] R. Cattoni, T. Coianiz, S. Messelodi and C. M. Modena. “Geometric Layout Analysis Techniques for Document Image Understanding: a Review”. Technical Report, IRST, Trento, Italy, 1998.

[25] F. Cesarini, M. Gori, S. Marinai and G. Soda. “Structured Document Segmentation and Representation by the Modified X-Y tree”. Proceedings of the 5th International Conference on Document Analysis and Recognition, Bangalore, India, September 1999, pp. 563-566.

[26] F. Cesarini, M. Lastri, S. Marinai and G. Soda. “Encoding of Modified X-Y trees for document classification”. Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, September 2001, pp. 1131-1136.

[27] F. Chang, K. Liang, T. Tan and W. Hwang. “Binarization of Document Images using Hadamard Multiresolution Analysis”. Proceedings of the 5th International Conference on Document Analysis and Recognition, Bangalore, India, September 1999, pp. 374-377.

[28] A. Dengel and B. Klein. “smartFIX: A Requirements-Driven System for Document Analysis and Understanding”, Proceedings of the 5th International Workshop on Document Analysis Systems, DAS2002, Princeton, USA, August 2002, pp. 433-444.

[29] D. Drivas and A. Amin. “Page Segmentation and Classification Utilizing Bottom-Up Approach”. Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, Canada, 1995, pp. 610-614.

[30] F. Esposito, D. Malerba and G. Semeraro. “A Knowledge-Based Approach to the Layout Analysis”. Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, Canada, 1995, pp. 466-471.

[31] J. Fisher, S. Hinds and K. D'Amato. “A Rule-Based System for Document Image Segmentation”. Proceedings of the 10th International Conference on Pattern Recognition, Atlantic City, USA, 1990, pp. 113-122.

[32] B. Gatos, S. L. Mantzaris and A. Antonacopoulos. “First International Newspaper Segmentation Contest”. Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, September 2001, pp. 1190-1194.

[33] B. Gatos, S.L. Mantzaris, K.V. Chandrinos, A. Tsigris and S.J. Perantonis. “Integrated Algorithms for Newspaper Page Segmentation and Article Tracking”. Proceedings of the 5th International Conference on Document Analysis and Recognition, Bangalore, India, September 1999, pp. 559-562.

[34] V. Govindaraju, S. W. Lam, D. Niyogi, D. B. Sher, R. Srihari, S. N. Srihari and D. Wang. “Newspaper Image Understanding”. In S. Ramani, R. Chandrasekar and K. S.

R. Anjaneyulu editors, Knowledge Based Computer Systems, Narosa Publishing House New Delhi India, 1990, pp. 375-384.

[35] K. Hadjar and R. Ingold. "Arabic Newspaper Page Segmentation". Proceedings of the 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, August 2003, pp. 895-899.

[36] K. Hadjar and R. Ingold. "Logical labeling of Arabic Newspapers using Artificial Neural Nets". Proceedings of the 8th International Conference on Document Analysis and Recognition, Seoul, Korea, August 2005, pp. 426-430.

[37] K. Hadjar and R. Ingold. "Physical Layout Analysis of Complex Structured Arabic Documents using Artificial Neural Nets", Proceedings of the 6th International Workshop on Document Analysis Systems, DAS2004, Florence, Italy, September 2004, pp. 170-178.

[38] K. Hadjar, O. Hitz and R. Ingold. "Newspaper Page Decomposition using a Split and Merge Approach". Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, September 2001, pp. 1186-1189.

[39] K. Hadjar, O. Hitz, L. Robadey and R. Ingold. "Configuration REcognition Model for Compley Reverse Engineering Methods: 2(CREM)", Proceedings of the 5th International Workshop on Document Analysis Systems, DAS2002, Princeton, USA, August 2002, pp. 469-479.

[40] K. Hadjar, O. Hitz, L. Robadey and R. Ingold. "2(CREM) Une méthode de reconnaissance interactive pour les documents à structure complexe", Proceedings of the Colloque International Francophone sur l'Ecrit et le Document, CIFED'2002, Hammamet, Tunisia, October 2002, pp. 235-244.

[41] K. Hadjar, M. Rigamonti, D. Lalanne and R. Ingold. "Xed: a new tool for eXtracting hidden structures from Electronic Documents", Proceedings of the 1st International Workshop on Document Image Analysis for Libraries, DIAL2004, Palo Alto, USA, January 2004, pp. 212-221.

[42] H. Haouala and M. C. Fehri. "Arabic document analysis and recognition a case study: official journal of the Tunisian republic ", Proceedings of the Computational Engineering in Systems Applications, CESA'98, Hammamet, Tunisia, April 1998, pp. 9-12.

[43] R. M. Haralick. "Document Image Understanding: Geometric and Logical Layout". Proceedings of the International Conference on Computer Vision and Pattern Recognition, 1994, pp. 385-390.

[44] E. Herwijnen. "Practical SGML". Kluwer Academic Publisher, 1990.

[45] O. Hitz and R. Ingold. “Visualisation of Document Recognition Results using XML Technology”. Proceedings of the Colloque International sur le Document Electronique, Lyon, France, July 2000.

[46] O. Hitz, L. Robadey and R. Ingold. “Using XML in Document Recognition”, Document Layout Interpretation and its Applications, Bangalore, India, September 1999.

[47] O. Hitz, L. Robadey and R. Ingold. “An architecture for Editing Document Recognition Results using XML Technology”, Proceedings of the 4th International Workshop on Document Analysis Systems, DAS2000, Rio de Janeiro, Brazil, December 2000, pp. 385-396.

[48] R. Ingold and D. Armangil. “A Top-Down Document Analysis Method for Logical Structure Recognition”. Proceedings of the 1st International Conference on Document Analysis and Recognition, Saint-Malo, France, 1991, pp. 41-49.

[49] T. Hu and R. Ingold. “A mixed approach toward an efficient logical structure recognition from document images”. Electronic Publishing, 6(4), 1993, pp. 457-468.

[50] A. K. Jain and K. Karu. “Learning Texture Discrimination Masks”. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no 2, 1996, pp. 195-205.

[51] A. K. Jain and B. Yu. “Document Representation and its Application to Page Decomposition”. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 3, March 1998, pp. 294-308.

[52] A. K. Jain and B. Yu. “Page Segmentation Using Document Model”. Proceedings of the 4th International Conference on Document Analysis and Recognition, Ulm, Germany, August 1997, pp. 34-37.

[53] A. K. Jain and Y Zhong. “Page Segmentation Using Texture Analysis”. Pattern Recognition, vol. 29, no 5, 1996, pp. 743-770.

[54] JavaNNS. “Java Neural Network Simulator”. http://www-ra.informatik.uni-tuebingen.de/software/JavaNNS/welcome_e.html

[55] J. Kanai, S.V. Rice, T.A. Nartker and G. Nagy. “Automated Evaluation of OCR Zoning”. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 17, no 1, January 1995, pp. 86-90.

[56] K. Kise, A. Sato and M. Iwata. “Segmentation of Page Images Using the Area Voronoi Diagram”. Computer Vision and Image Understanding, vol. 70, no. 3, 1998, pp. 370-382.

[57] M. Krishnamoorthy, G. Nagy, S. Seth and M. Viswanathan. "Syntactic Segmentation and Labeling of Digitized Pages from Technical Journals". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, No. 7, July 1993, pp. 737-747.

[58] F. Lebourgeois, Z. Bublinski and H. Emptoz. "A Fast and Efficient Method for Extracting Text Paragraphs and Graphics From Unconstrained Documents". Proceedings of the 11th International Conference on Pattern Recognition, The Hague, 1992, pp. 272-276.

[59] J. Liang and D. Doermann. "Logical Labeling of Document Images Using Layout Graph Matching with Adaptive Learning", Proceedings of the 5th International Workshop on Document Analysis Systems, DAS2002, Princeton, USA, August 2002, pp. 224-235.

[60] J. Liang, I.T. Philips and R.M. Haralick. "Performance Evaluation of Document Structure Extraction Algorithms", Computer Vision and Image Understanding, volume 84, Issue 1, October 2001, pp. 144-159.

[61] F. Liu, Y. Luo, M. Yoshikawa and D. Hu. "A New Component based Algorithm for Newspaper Layout Analysis". Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, September 2001, pp. 1176-1179.

[62] J. Liu, Y. Tang, Q. He and C. Suen. "Adaptive Document Segmentation and Geometric Relation Labeling: Algorithms and Experimental Results". Proceedings of the 13th International Conference on Pattern Recognition, Vienna, Austria, 1996, pp. 763-767.

[63] D. Malerba, F. Esposito and O. Altamura. "Correcting the Document Layout: A Machine Learning Approach". Proceedings of the 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, August 2003, pp. 97-102.

[64] S. Mao and T. Kanungo. "Automatic Training of Page Segmentation Algorithms: An Optimization Approach", Proceedings of International Conference on Pattern Recognition, Barcelona, Spain, September 2000, pp. 531-534.

[65] S. Mao, A. Rosenfeld and T. Kanungo. "Document Structure Analysis Algorithms: A Literature Survey". Proc. SPIE Electronic Imaging, Santa Clara, California, USA, January 2003, pp. 197-207.

[66] P. E. Mitchell and H. Yan. "Newspaper Document Analysis featuring Connected Line Segmentation". Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, September 2001, pp. 1181-1185.

[67] M. Nadler. "A Survey of Document Segmentation and Coding Techniques". Computer Vision, Graphics, and Image Processing, vol. 28, 1984. pp. 240-262.

[68] G. Nagy, J. Kanai, M. Krishnamoorthy, M. Thomas, and M. Viswanathan. “Two Complementary Techniques for Digitized Document Analysis”. Proceedings ACM Conference on Document Processing Systems, Santa Fe, New Mexico, USA, December 1988.

[69] G. Nagy and S. Seth. “Hierarchical representation of optically scanned documents”. Proceedings of ICPR, 1984, pp. 347-349.

[70] G. Nagy, S. Seth and M. Viswanathan. “A prototype document image analysis system for technical journals”. Computer, vol. 25, No. 7, July 1992, pp. 10-22.

[71] H. E. Nielson and W. A. Barrett. “Consensus-Based Table Form Recognition”. Proceedings of the 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, August 2003, pp. 906-910

[72] D. Niyogi and S. Srihari. “Knowledge-Based Derivation of Document Logical Structure”. Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, Canada, 1995, pp. 472-475.

[73] L. O'Gorman. “The Document Spectrum for Page Layout Analysis”. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, issue 11, November 1993, pp. 1162-1173.

[74] G. I. Palermo and Y. A. Dimitriadis. “Structured document labeling and rule extraction using a new recurrent fuzzy-neural system”. Proceedings of the 5th International Conference on Document Analysis and Recognition, Bangalore, India, September 1999, pp. 181-184.

[75] T. Pavlidis and J. Yhou. “Page Segmentation and Classification”. CVGIP Vol 54, No. 6, 1992, pp. 482-469.

[76] T. Pavlidis and J. Zhou.“Page Segmentation by White Streams”, Proceedings of the 1st International Conference on Document Analysis and Recognition, St-Malo, France, September 1991, pp. 945-953.

[77] M. Pechwitz. “HMM Based Approach for Handwritten Arabic Word Recognition Using the *IFN/ENIT* - Database”. Proceedings of the 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, August 2003, pp. 890-894.

[78] M. Pfister, S. Behnke and R. Rojas. “Recognition of Handwritten ZIP Codes in a Real – World Non-Standard-Letter Sorting System”. Journal of Applied Intelligence, 2, 1998, pp. 1-25

[79] I.T. Philips, S. Chen, J. Ha and R.M. Haralick. “English Document Database Design and Implementation Methodology”. Proceedings of the 2nd Annual Symp. Doc. Analysis and Retrieval, UNLV, USA, 1993, pp. 65-104.

[80] I.T. Philips, S. Chen and R.M. Haralick. “CD-ROM Document Database Standard”. Proceedings of the 2nd International Conference on Document Analysis and Recognition, Tsukuba, Japan, 1993, pp. 478-483.

[81] RAF Technology. “DAFS Library, Programmer's Guide and Reference”. August 1995.

[82] M. Rigamonti, K. Hadjar, D. Lalanne et R. Ingold. “Xed : un outil pour l'extraction et l'analyse de documents PDF”, Proceedings of the Colloque International Francophone sur l’Ecrit et le Document, CIFED’2004, La Rochelle, France, June 2004, pp. 85-90.

[83] L. Robadey. “2(CREM) Une méthode de reconnaissance structurelle de documents complexes basée sur des patterns bidimensionnels”. PhD thesis, DIUF-University of Fribourg, Fribourg, Switzerland 2001.

[84] N. Roussel, O. Hitz and R. Ingold. “Web-based Cooperative Document Understanding”. Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, September 2001, pp. 368-373.

[85] T. Saitoh and T. Pavlidis. “Page Segmentation without Rectangle Assumption”. Proceedings of the 11th International Conference on Pattern Recognition, The Hague, USA, 1992, pp. 277-280.

[86] H. Sako, N. Furukawa, M. Fujio and S. Watanabe. “Document-Form Identification Using Constellation Matching of Keywords Abstracted by Character Recognition”, Proceedings of the 5th International Workshop on Document Analysis Systems, DAS2002, Princeton, USA, August 2002, pp. 261-271

[87] S. Souafi-Bensafi, M. Parizeau, F. Lebourgeois and H. Emptoz. “Logical Labeling using Bayesian Networks”. Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, September 2001, pp. 832-836.

[88] A. L. Spitz. “Recognition Processing for Multilingual Documents”, In R. Furuta (ed.), Proceedings of the 1990 International Conference on Electronic Publishing, Document Manipulation and Typography, Gaithersburg, Maryland, USA, September 1990, pp. 193-205.

[89] S. N. Srihari and G. W. Zack. “Document Image Analysis”. Proceedings of the 8th International Conference on Pattern Recognition, Paris, France, October 1986, pp. 434-436.

[90] C. Strouthopoulos and N. Papamarkos. “Text identification for document image analysis using a neural network”. Image and Vision Computing, vol. 16, no 12/13, 1998, pp. 879-896.

[91] Y. Y. Tang, M. Cheriet, J. Liu, J. N. Said, C. Y. Suen. “Document Analysis and Recognition by Computers”. Handbook of Pattern Recognition and Computer Vision.

[92] C. I. Tomai, K. M. Allen and S. N. Srihari. “Recognition of Handwritten Foreign Mail”. Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, September 2001, pp. 882-886

[93] S. Tsujimoto and H. Asada. “Understanding multi-articled documents”. Proceedings of the 10th International Conference on Pattern Recognition, Atlantic City, USA, June 1990, pp. 551-556

[94] M. Viswanathan. “Analysis of scanned documents a syntactic approach”. in Structured Document Image Analysis, Springer-Verlag, 1992, pp. 115-136.

[95] F. Wahl, K. Wong and R. Casey. “Block Segmentation and Text Extraction in Mixed Text/Image Documents”. Computer Vision Graphics, and Image Processing, vol. 20, 1982, pp. 375-390.

[96] Y. Wang, R. Haralick and I.T. Philips. “Zone Content Classification and Its Performance Evaluation”. Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, September 2001, pp. 540-544.

[97] Y. Wang, I.T. Philips and R.M. Haralick. “A Study on the Document Zone Content Classification Problem”, Proceedings of the 5th International Workshop on Document Analysis Systems, DAS2002, Princeton, USA, August 2002, pp. 212-223.

[98] Y. Wang, I.T. Philips and R.M. Haralick. “Automatic Table Ground Truth Generation and A Background-analysis-based Table Structure Extraction Method”. Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, September 2001, pp. 528-532.

[99] D. Wang and S. N. Srihari. “Classification of newspaper image blocks using texture analysis”. Computer Vision Graphics and Image Processing, vol. 47, no. 3, September 1989, pp. 327-352.

[100] D. Wasserman. “Neural Computing: Theory and Practice”. Van Nostrand Reinhold 1989.

[101] K. Y. Wong, R. G. Casey and F. M. Wahl. “Document Analysis System”. IBM Journal of Research and Development, vol. 26, No 6, November 1982, pp. 647-656.

[102] World Wide Web Consortium (W3C). “Extensible Markup Language (XML) 1.0 (Third Edition)”. http://www.w3c.org/TR/REC-xml, February 2004.

[103] Q. Xu, L. Lam and C. Suen. “Automatic Segmentation and Recognition System for Handwritten Dates on Canadian Bank Cheques”. Proceedings of the 7th International Conference on Document Analysis and Recognition, Edinburgh, Scotland, August 2003, pp. 704-708

[104] A. Yamashita, T. Amano, Y. Hirayama, N. Itoh, S. Katoh, T. Mano and K, Tokokawa. “A Document Recognition System and Its Applications”. IBM Journal of Research and Development, vol. 40, 1996, pp. 341-352.

[105] B.A. Zanikoglu and L. Vincent. “Pink Panther: A Complete Environment for Ground-Truthing and Benchmarking Document Page Segmentation”. Pattern Recognition, vol. 31, no 9, January 1998, pp. 1191-1204.

[106] J. Zhou, C. Y. Suen and K. Liu. “A feedback-based Approach for Segmenting Handwritten Legal Amounts on Bank Cheques”. Proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, September 2001, pp. 887-891

Nom : HADJAR
Prénom : Karim
Date et lieu de naissance : 30 Janvier 1968 à Tunis
Statut Familial : Marié
E-MAIL: karim.hadjar@unifr.ch

Etudes

- Primaires **:**1973 / 1980 Ecole primaire Rue d'ANGOLA (Jeanne d'ARC), Tunis, Tunisie
- Secondaires : 1980 / 1987 Collège El Menzah VI, Tunis, Tunisie
- Universitaires :
  - 1987 / 1989 **Institut Supérieur de Gestion de Tunis, Tunisie** (Maîtrise Informatique de gestion) (I.S.G)
  - 1991 / 1993 **Institut Supérieur de Gestion de Tunis, Tunisie** (Maîtrise Informatique de gestion) (I.S.G)
  - 1993 / 1995 **Institut Supérieur de Gestion de Tunis, Tunisie** (Mastère en Modélisation et Informatique de gestion) (I.S.G)
  - 2001 / 2005 **département d'informatique de l'université de Fribourg, Suisse** (Doctorat en informatique) (DIUF Université de Fribourg)

Diplômes obtenues

- **Baccalauréat** (Section : Math Sciences Juin 1987)
- **Diplôme de technicien supérieur en informatique de gestion** Juin 1989
- **Maîtrise en informatique de gestion** Juin 1993 (Mention très bien)
- **Mastère en modélisation et informatique de gestion** Novembre 1995 (Mention très bien)
- **Doctorat en informatique**

Publications Conférences

1- **"Newspaper Page Decomposition using a Split and Merge Approach",** Karim HADJAR, Oliver Hitz and Rolf Ingold, ICDAR'01 Seattle (USA) 10-13 September 2001, pp. 1186-1189
2- **"Arabic Newspaper Page Segmentation"**, Karim HADJAR and Rolf Ingold, ICDAR'2003 Edinburgh (Scotland) 03-06 August 2003, pp. 895-899
3- **"Logical Labeling of Arabic Newspapers using Artificial Neural Nets "**, Karim HADJAR and Rolf Ingold, ICDAR'05 Seoul (Korea) 29 Aout -1 Septembre 2005, pp. 426-430

Publications Revues

1. **"Physical Layout Analysis of Complex Structured Arabic Documents using Artificial Neural Nets"**, Karim HADJAR and Rolf Ingold, DAS'2004 Florence (Italy) 08-10 September 2004, pp. 170-178

Logiciels Multimédia

- MACROMEDIA DIRECTOR 8

- MACROMEDIA DREAMWEAVER 3
- ADOBE PHOTOSHOP 5.5
- ADOBE ILLUSTRATOR 8
- 3D STUDIO MAX RELEASE 3.1
- ADOBE PREMIERE 5.1
- SOUND FORGE 4.5

Langages de script

- BASH SHELL (LINUX)
- JavaScript
- JAVA SERVER PAGES

Méthodes de conception

- Castellani
- Merise
- OMT
- UML

Internet

- HTML
- XML, XSLT, XPATH, XML SCHEMA, SVG, JDOM

Langages informatiques

- C
- Visual Basic Pro Version 3.0, 4.0, 5.0, 6.0 and .NET
- Visual C++ 1.0 and 4.0
- JAVA J2SE

Bases de données relationnelles

- PostGreSQL version 6.5.2 (on LINUX)
- ORACLE 8.0.5 (on LINUX)

Systèmes d'exploitation et interfaces graphiques

- Red Hat LINUX 5.2, 6.0, 6.1, 6.2, 7.0, 7.1
- LINUX MANDRAKE 7.0, 7.1, 7.2, 8.0, 8.1, 9, 10
- MS-DOS
- SunOS (UNIX SYSTEM V Release 4.1.1), SOLARIS 1.0
- Windows 3.0, 3.1, 3.11, 95, 98, NT, 2000 & XP
- OpenLOOK
- Xwindows, AfterStep, KDE, Gnome

Langues parlées et écrites

- Arabe, Français, Anglais, Italien