Un panaroma des méthodes syntaxiques pour la segmentation d'images de document manuscrits

Stéphane Nicolas[1] – Thierry Paquet[1] – Laurent Heutte[1]

Laboratoire PSI
CNRS FRE 2645 - Université de Rouen
Place E. Blondel
UFR des Sciences et Techniques
F-76 821 Mont-Saint-Aignan cedex

{Stephane.Nicolas, Thierry.Paquet, Laurent.Heutte}@univ-rouen.fr

Résumé : *La problèmatique de la segmentation d'images de documents qui consiste à en extraire les entités informatives et à déterminer les relations spatiales et hiérarichques entre ces composantes peut être vue comme une procédure inverse à la phase de production du document. Si on considère que le document a été généré à partir d'un modèle de structuration donné, la segmentation consiste à déterminer la meilleure correspondance entre le modèle et l'image du document. Il s'agit alors d'un problème d'analyse syntaxique. Nous dressons dans ce papier un panaroma des principaux modèles et méthodes d'analyses syntaxiques utilisés traditionnellement dans les domaines du traitement des langages naturels et de la compilation des langages machines, pour l'analyse de structures monodimensionnelles, et discutons de l'adaptation de ces techniques à des problèmes de segmentation d'images de documents manuscrits.*

Mots-clés : segmentation de documents manuscrits non contraints, analyse syntaxique

1 Introduction

Nous nous intéressons à la segmentation de documents manuscrits non contraints en entités informatives élémentaires. Cette étude est motivée par le constat que les principaux travaux envisagés dans la littérarture ce sont jusqu'à présent principalement focalisés sur la reconnaissance de mots isolés ou de messages très contraints dans leur format et leur disposition spatiale. L'extension de ces approches à des documents au format et au contenu moins contraints pose des difficultés spécifiques que les approches utilisées pour l'analyse des documents imprimés ne permettent pas de résoudre. Dans cet article nous faisons une synthèse non exhaustive des approches qui pourraient être envisagées pour dépasser les limitation actuelles. L'une des problématiques récurrentes en analyse de document est de parvenir à intégrer des connaissances formelles, dont on peut disposer sur un corpus particulier, dans le processus d'analyse qui doit prendre en compte la variabilité des instanciations physiques des entités. Ainsi par exemple, dans le cadre de la reconnaissance de mots manuscrits on peut expliquer le succès des approches Markoviennes parce qu'elles parviennent à faire une adéquation efficace entre un modèle de connaissances (lexicales dans ce cas particulier) et l'ambiguïté dans l'interprétation des entités physique de l'image. Cette brève analyse nous a conduit à nous intéresser principalement aux modèles de connaissannces qui permettent d'exprimer la structure d'un document, en privilégiant les approches modélisant les ambiguïtés. Enfin, un point important, est l'aptitude de ces modèles à représenter les descriptions bi-dimensionnelles.

Le but du processus de segmentation est d'extraire la structure physique du document. Formellement la structure d'un objet complexe permet de le décrire en termes d'objets plus simples et d'exprimer les relations qui existent entre ces objets. La description de la structure d'un objet est alors faite par un modèle particulier. Un document quel qu'il soit, imprimé ou manuscrit, peut être vu comme le résultat d'opérations de formatage et de mise en page du contenu informatif qu'il véhicule. Segmenter consiste à partionner le document en régions homogènes. Dans le cas des documents textuels cela revient à identifier et localiser les blocs de textes, les paragraphes, les lignes de texte, les mots... etc. Cette structuration est hiérarchique et peut être représentée par un arbre. Deux approches sont possibles pour déterminer cette structure, une approche ascendante ou une approche descendante. La première consiste à extraire les primitives du documents, puis à regrouper itérativment ces primitives en région homogènes. La seconde cherche à diviser récursivement l'image en sous-régions homogènes. Enfin une troisième solution consiste à combiner les deux approches. De nombreuses méthodes basées sur l'une de ces stratégies sont proposées dans la litérature pour la segmentation des documents imprimés. Parmi les plus populaires on peut notamment citer la méthode de Kise [KIS 98] basée sur l'utilisation de diagrammes de Voronoi, la méthode Docstrum de O'Gorman [O'G 93] basée sur un regroupement en voisins, ou encore la méthode de découpage X-Y de Nagy [NAG 92] utilisant les profils de projection. Cependant ces méthodes ne sont en général appliquables qu'à des documents structurés présentant peu de variabilité dans la disposition. Ce n'est pas le cas des documents manuscrits qui présentent en général une structure globale relativement simple (arrangements de lignes de textes et éventuellement de paragraphes) mais localement une variabilité importante : lignes de texte inclinées et fluctuantes, recouvrement entre les lignes, paragraphes non alignés. De ce

fait les méthodes d'analyse proposées dans la littérature se limitent généralement à localiser et extraire les lignes de texte, et sont généralement basées sur une stratégie ascendante. Ces méthodes reposent sur une analyse locale. La méthode proposée dans [LIK 94] consiste en regroupement itératif de composantes connexes basé sur des critères de similarité, de continuité et de proximité. Cette méthode intègre également une procédure de résolution de conflits basée sur une combinaison d'une analyse locale et d'une analyse globale. Une méthode consistant à rechercher l'arbre recouvrant minimum du document est proposée dans [ABU 95] pour l'extraction de ligne dans les documents arabes. Dans [LIK 95] une stratégie "émission d'hypothèse - validation" utilisant la transformée de Hough est utilisée. On trouvera un aperçu des principales autres méthodes dans [BRU 99]. Toutes ces méthodes ont pour inconvénients d'une part de nécessiter le réglage de nombreux paramètres, qui ne peuvent être déterminés que de manière empirique, ce qui ne permet pas un apprentissage automatique, et donc limite l'adaptation de ces méthodes à différentes structures, et d'autre part, elles reposent pour la plupart sur une analyse trop locale pour permettre de gérer certaines ambiguités. Enfin ces méthodes formulent des hypothèses a priori sur la structure du document à traiter, mais ces connaissances ne sont pas exprimée explicitement, ce qui est un inconvénient supplémentaire pour l'adaptation à différents types de structure. En considérant que segmenter revient à déterminer la structure syntaxique du document, c'est à dire identifier les objets du documents, et déterminer les relations entre ces objets, c'est à dire déterminer les règles sous-jacentes de mise en forme de ces objets, nous pensons qu'il y a un intérêt à exprimer explicitement les connaissances a priori sur la structure du document à l'aide d'un formalisme intuitif et intelligible permettant de décrire simplement les règles de structuration du document. Pour cela on peut s'inspirer des modèles grammaticaux et techniques d'analyse syntaxique utilisés dans le domaine du traitement de la langue. De plus étant donnée la forte variabilité dans la structure des documents manuscrits non contraints, il est nécessaire de ce placer dans un cadre probabiliste. Un tel formalisme présente l'avantage de permettre un apprentissage des paramètres du modèle d'une part tout en garantissant l'explicitation et l'externalisation des connaissances descriptives du problème traité.

Dans la section 1 nous passons en revue les principaux modèles grammaticaux utilisées pour modéliser des structures monodimensionnelles, à savoir les grammaires de chaînes, ainsi que les techniques d'analyses associées à ces modèles, en commençant par les modèles formels, puis en discutant des extensions probabilistes de ces modèles. Dans la 2ème section nous discutons de l'extension au cas 2D des techniques d'analyses utilisées dans le cas des modèles probabilistes 1D. Nous terminerons par des perspectives sur l'utilisation d'une telle extension dans le cas d'un problème de segmentation physique de documents manuscrits non contraints.

2 Grammaires de chaînes

Les grammaires de chaînes ont été intialement introduites par les linguistes (notamment Chomsky) pour modéliser les langages naturels. On distingue les grammaires formelles de leurs extensions probabilistes.

2.1 Grammaires formelles

Une grammaire est un modèle génératif pouvant être utilisé pour des problèmes de reconnaissance. Une grammaire est constituée d'un ensemble de règles de structuration qui permettent de réécrire une chaîne en une autre (de taille supérieure), selon un principe de dérivation. Formellement un modèle grammatical est défini par un quadruplet $G = (T, N, P, S)$ où T désigne le vocabulaire des symboles terminaux (unités lexicales), N un vocabulaire auxiliaire de symboles non-terminaux (correspondant aux catégories syntaxiques), P un ensemble de règles de réécriture de la forme $\alpha \rightarrow \beta$ où est α et β sont des chaînes de symboles terminaux et non-terminaux, et S un symbole initial à partir duquel sont dérivées des chaînes de symboles terminaux. On appelle langage engendré (ou reconnu) par une grammaire, l'ensemble des chaînes (séquences de symboles) que la grammaire permet de générer (ou de reconnaître). Une grammaire formelle constitue donc un moyen de définir un langage, c'est à dire un ensemble de chaînes partageant certaines caractéristiques structurelles (définies par les productions). Une hiérarchie des grammaires a été établie en fonction de leur pouvoir d'expression. On distingue les grammaires de type 3 (grammaires régulières), de type 2 (grammaires non-contextuelles), de type 1 (contextuelles) et de type 0. Ces différentes classe de grammaires sont obtenues en posant des resctrictions sur la forme des règles de réécriture. Les grammaires de type 3 sont les plus restrictives, et celles de type 0 ne posent aucune restriction.

2.1.1 Grammaires régulières

Les productions sont de la forme $A \rightarrow aB$ ou $A \rightarrow a$ (grammaire régulière à droite) avec $A, B \in N$ et $a \in T$. Les langages engendrés par des grammaires régulières sont des langages réguliers. Les grammaires régulières permettent de modéliser des structures non récursives et sont particulièrement adaptées pour modéliser des phénomènes de répétition. L'analyseur permettant de reconnaître des langages réguliers est un automate fini déterministe [KUN 00]. Dans la pratique une telle méthode d'analyse est facilement implémentable.

2.1.2 Grammaires non-contextuelles

Les règles de réécriture sont moins restrictives. Elles sont de la forme $A \rightarrow \beta$ avec $A \in N$ et $\beta \in \{N \cup T\}^+$, (soit une chaîne sur la réunion des alphabets terminal et non-terminal). La réécriture d'un symbole non-terminal donné ne dépend pas de ce qu'il y a autour (contexte) dans la chaîne. Dans le cas des grammaires "hors-contexte", il est plus pratique de représenter les dérivations d'une manière canonique, c'est pourquoi on utilise couramment une structure d'arbre, appelée arbre de dérivation ou arbre d'analyse. Généralement dans une grammaire "hors-contexte" il y a plusieurs règles de réécriture possibles pour un symbole non-terminal donné, ce qui peut aboutir à générer une même phrase avec des dérivations différentes. Dans ce cas la grammaire est ambigüe. Etant donnée une grammaire hors-contexte et une chaîne d'entrée, le principe de l'analyse consiste à déterminer le ou les arbres de dérivation de cette chaîne selon la grammaire. Deux principaux algorithmes sont utilisés pour réali-

ser cette analyse : l'algorithme CYK et l'algorithme de Earley [KUN 00]. L'algorithme CYK est un algorithme ascendant utilisant le principe de la programmation dynamique. Il nécessite que la grammaire soit exprimée dans la forme normale de Chomsky, c'est à dire que ses règles de production soient de la forme $A = BC$ ou $A = a$ avec $(A, B, C \in N)$ et $(a \in T)$. On suppose donc qu'un non terminal donné puisse se décomposer en deux non-terminaux ou se réécrire en un symbole terminal, ce qui permet d'utiliser le principe de la programmation dynamique (un problème se décompose en deux sous-problèmes plus simples). Cet algorithme utilise une matrice triangulaire pour mémoriser les résultats partiels de l'analyse. Chaque case (i, j) de cette matrice mémorise l'ensemble $V_{i,j}$ des non-terminaux qui dérive la sous-chaîne $w_i...w_j$. La chaîne analysée est grammaticale si l'axiome S appartient à l'ensemble $V_{1,m}$, c'est à dire s'il existe au moins une dérivation $S \stackrel{*}{=>} w_1...w_m$. Si la chaîne est grammaticale, les différentes analyses peuvent ensuite être retrouvées en parcourant la matrice dans le sens inverse de sa construction. La complexité de cet algorithme est en $O(m^3)$. L'algorithme de Earley est une méthode d'analyse descendante qui ne nécessite aucune forme particulière de la grammaire. Il consiste à partir de l'axiome de la grammaire et à dériver systématiquement en profondeur le terminal le plus à gauche dans la chaîne de manière à générer des prédictions de dérivation. Ces prédictions sont ensuite vérifiées dans la chaîne à analyser. La fin de l'analyse se produit lorsque toute la chaîne a été parcourue.

2.1.3 Grammaires de type 1 et de type 0

Malgré leur capacité à modéliser des structures plus complexes, les grammaires de type 1 et 0 ne sont pas ou peu utilisées dans la pratique, car il n'existe pas de techniques d'analyse performantes pour ce type de grammaires. Nous ne détaillerons donc pas ici ces grammaires.

2.2 Extensions probabilistes

Les modèles formels vus précédemment ne permettent pas de faire face à une certaine variabilité dans les structures analysées. Des modèles stochastiques permettant de prendre en compte des ambiguités syntaxiques ont donc été proposés, parmi lesquels on peut distinguer des extensions probabilistes de grammaires formelles, et des modèles entièrement statistiques comme les modèles markoviens. A la différence des grammaires formelles qui déterminent si une phrase donnée est grammaticale ou non, les grammaires probabilistes déterminent la vraisemblance que la phrase soit générée par la grammaire. Une grammaire probabiliste est définie simplement en associant des probabilités aux productions. Ces probabilités traduisent la probabilité conditionnelle d'application d'une règle étant donné un certain non-terminal $A \rightarrow \lambda[p]$ où $p = P(A \rightarrow \lambda|A)$

2.2.1 Grammaires régulières probabilistes

Comme pour toutes les grammaires probabilistes, les grammaires régulières probabilistes (GRP) sont définies par une grammaire caractéristique formelle, qui décrit l'ensemble des règles de structuration du modèle, et d'un ensemble de probabilités associées à chacune de ces règles de structuration. Ici, la grammaire caractéristique est une grammaire régulière. L'algorithme d'analyse pour une grammaire régulière probabiliste est un automate probabiliste à nombre fini d'états. A la différence d'un automate fini classique, dans le cas des automates probabilistes des probabilités pondèrent les arcs de l'automate. Ces pondérations traduisent la probabilité de passer d'un état donné à un autre. A l'aide d'un automate probabiliste, on peut déterminer la probabilité qu'une chaîne donnée ait été générée par le modèle.

2.2.2 Modèles de Markov Cachés

Les modèles de markov cachés (MMC) sont les modèles statistiques les plus utilisés dans le domaine de la reconnaissance de la parole, que ce soit pour la reconnaissance de mots isolés ou par la reconnaissance de la parole continue [BOI 00]. Un modèle de markov caché est une fonction probabiliste d'un processus markovien [MAN 02]. Un modèle Markovien est un processus stochastique, c'est à dire un processus aléatoire qui peut changer d'état au cours du temps, chaque état correspondant à la réalisation d'une variable aléatoire $X(t) = o_i$. L'état d'un processus Markovien au temps t ne dépend que de son état à l'instant $t - 1$, et les probabilités de transitions entre les états ne dépendent pas du temps. Les modèles markoviens permettent d'analyser les séquences possibles de symboles dans une chaîne et de modéliser des structures régulières, comme les grammaires régulières probabilistes. Les modèles de markov cachés se placent à un niveau d'abstraction supérieur en considérant qu'il existe une structure "cachée" derrière les observations. L'avantage des MMC par rapport aux grammaires régulières probabilistes est qu'ils permettent de travailler sur des structures régulières non observables tout en bénéficiant d'algorithmes d'analyses et d'apprentissage efficaces comme nous allons le voir. Ils sont donc bien appropriés pour décrire des données bruitées mono-dimensionnelles régies par une structure régulière.

Formellement un modèle de Markov caché est défini par :

- un ensemble d'états internes cachés $S = \{s_1, ..., s_N\}$ où $|S| = N$
- la matrice des probabilités de transition entre états $A = \{a_{ij}\} \forall (i, j) \in [1, N]$
- les probabilités initiales des états $\Pi = \{\pi_i = P(s_i)\}$
- dans le cas d'un modèle discret, l'alphabet des symboles émis par les états $\Sigma = \{v_1, ...v_k\}$
- les probabilités d'émission des symboles par les états $B = \{b_j(v_k) = P(v_k/s_i\}$

Nous rappelons brièvement ici les problèmes classiques auxquels on peut être confronté lorsque l'on choisit de travailler avec de tels modèles :

- calculer la probabilité qu'une séquence observée soit générée par un modèle
- déterminer la séquence d'états cachés la plus probable qui puisse être associée à une séquence d'observation
- estimer les paramètres du modèle qui maximisent la vraisemblance d'un ensemble de séquences d'apprentissage (apprentissage des paramètres du modèle)

La réponse à la première question permet de discriminer un ensemble de modèles pour les classer dans l'ordre des vrai-

semblances décroissantes. C'est le cas par exemple en reconnaisance de mots manuscrits dirigée par le lexique. La réponse à la seconde question permet d'accéder à la séquence d'état cachés et elle est utile si l'on souhaite associer une sémantique aux données. En se référant à la grammaire régulière sous-jacente au modèle Markovien on retrouve alors la meilleure séquence de règles qu'on l'on puisse associer aux données. L'accès à cette connaissance permet de segmenter les données observées ; en caractères dans le cas de la reconnaissanec de mots manuscrits, et de manière générale en fonction des règles ou des états. Enfin le dernier point apporte une réponse importante au problème de l'apprentissage des paramètres du modèle. Nous ne développerons pas ici plus avant les calculs relatifs au traitement complet de ces différents points. Notre ambition se limitant à montrer de manière unifiée les points communs à l'ensemble des approches

Etant donné un MMC λ et une séquence d'observations $O = o_1...o_T$ de longueur T, une manière directe de résoudre le premier problème consiste à sommer sur tous les chemins (séquences d'états internes $Q = q_1...q_T$) possibles les probabilités jointes de la séquence observée O et de Q :

$$P(O/\lambda) = \sum_{Q} P(O/Q,\lambda)P(Q/\lambda)$$

L'inconvénient de cette méthode est qu'elle présente une complexité non-linéaire en $O(2TN^T)$ (où T est le nombre d'observations, et N le nombre d'états internes). L'algorithme Forward-Backward, basé sur le principe de la programmation dynamique permet de résoudre plus efficacement ce problème, en factorisant les probabilités de sous-séquences communes à plusieurs séquences. Selon cet algorithme, le calcul de la probabilité $P(O/\lambda)$ utilise des relations de récurrence, basées soit sur les probabilités Forward $\alpha(i,t) = P(o_1...o_{t-1}, s_t = i/\lambda)$, soit sur les probabilités Backward $\beta(i,t) = P(o_{t+1}...o_n/s_t = i,\lambda)$.

La procédure Forward (en avant) est basée sur un parcours du treillis d'observation de la gauche vers la droite .Les relations de récurrence utilisées sont les suivantes :

$$\begin{aligned} \alpha(i,1) &= \pi_i b_i(o_1) \\ \alpha(j,t+1) &= [\sum_{i=1}^{N} \alpha(i,t)a_{ij}]b_j(o_t+1) \\ P(O/\lambda) &= \sum_{i=1}^{N} \alpha(i,T) \end{aligned}$$

La complexité de ce calcul est en $O(TN^2)$ au lieu de $O(2TN^T)$ pour le calcul direct. La procédure Backard est basée sur un parcours inverse du treillis avec le même type de factorisation. En pratique elle n'est utile en conjonction avec les variables forward que pour le problème de l'apprentissage (question 3) et nous ne nous étendrons pas d'avantage sur ce point ici.

La détermination de la meilleure séquence de règles (d'états) est également mise en oeuvre selon le principe de la programmaiton dynamique (algorithme de viterbi). La différence essentielle avec le calcul des variables forward est la transformation de la sommme sur tous les chemins possibles en une opération de recherche du meilleur état précédent et sa mémorisation pour permettre à l'issue des calculs de retrouver la meilleure séquence d'états par une procédure de recherche arrière (backtrack). Notons ici une extension naturelle de cet algorithme à la recherhe des N meilleures séquences d'états (N meileures possibilités de segmentation) qui consiste à mémoriser à chaque étape les N meilleures solutions partielles.

2.2.3 Grammaires probabilistes hors-contexte

Les grammaires probabilistes les plus couramment employées sont les grammaires probabilistes hors-contexte. Avec ce type de grammaire, l'application d'une règle ne dépend que de la présence du non-terminal qui la déclenche. En conséquence, la probabilité d'une dérivation donnée est simplement déterminée par le produit des probabilités des règles induites par la dérivation. Comme pour les MMC, étant donnée une grammaire probabiliste hors-contexte, on peut être amené chercher à à résoudre les problèmes suivants :

- quelle est la probabilité qu'une chaîne donnée ait été générée par la grammaire
- quel est l'arbre de dérivation le plus probable pour une chaîne donnée (ou les n arbres les plus probables)
- quels sont les paramètres du modèle (probabilités des règles) qui maximisent la probabilité qu'une chaîne donnée soit générée par la grammaire (apprentissage des probabilités des règles)

Une résolution directe du premier problème consiste à déterminer toutes les analyses possibles de la chaîne considérée selon la grammaire et à sommer les probabilités sur chacune de ces analyses. Etant donné une chaîne d'entrée w et une grammaire G, cela se traduit par la relation suivante, où où t est un arbre d'analyse possible pour la chaîne w :

$$P(w/G) = \sum_{t} P(w,t/G)$$

Une manière plus efficace pour résoudre ce problème consiste à utiliser le principe de la programmation dynamique d'une manière analogue aux développements effectués dans le cas des MMC. L'idée est de décomposer la chaîne considérée en sous-chaînes et de déterminer les probabilité de générer ces sous-chaînes, ceci de manière récursive jusqu'à obtenir des sous-chaînes élémentaires composées d'un seul symbole, dont la probabilité est triviale. Pour cela on définit les probabilités Inside $\alpha_j(p,q) = P(w_{p,q}/N^j_{p,q}, G)$ et Outside $\beta_j(p,q) = P(w_{1,p-1}, N^j_{p,q}, w_{q+1,m}/G)$ (analogues aux probabilités forward et backward dans le cas des MMC). La probabilité inside du symbole non-terminal N^j couvrant les symboles termimaux $w_p...w_q$, est la probabilité de générer la séquence de symboles terminaux $w_p...w_q$ à partir du non terminal N^j. La probabilité Outside est la probabilité de générer le début de la séquence de symboles $w_1...w_{p-1}$, le non-terminal N^j couvrant les symboles w_p à w_q et la fin de séquence de symboles terminaux $w_{q+1}...w_m$.

L'algorithme inside, basé sur l'utilisation des probabilités inside, permet de calculer la probabilité d'une chaîne en utilisant les relations de récurrence suivantes :

$$\alpha_j(k,k) = P(N^j \rightarrow w_k/G)\ 1 \leq k \leq m$$

$$\alpha_j(p,q) = \sum_{r,s}\sum_{d=p}^{q-1} P(N^j \rightarrow N^r N^s \alpha_r(p,d)\alpha_s(d+1,q)$$
$$P(w_{1m}/G) = \alpha_1(1,m)$$

Comme pour l'algorithme CYK, une matrice triangulaire est utilisée pour mémoriser les résultats partiels. Dans chaque case (p,q) de cette matrice, on mémorise les probabilités inside $\alpha_j(p,q)$ non nulles. L'algorithme utilisée pour déterminer l'arbre d'analyse le plus problable (second problème), est une adaptation de l'algorithme inside. Cet algorithme est très proche de l'algorithme de Viterbi. Au lieu d'effectuer la somme des probabilités de toutes les analyses possibles, on ne mémorise que la probabilité de la meilleure analyse $\delta_j(p,q)$. On mémorise également le chaînage des meilleurs constituants $\psi_j(p,q)$. Le meilleur arbre d'analyse est ensuite obtenu par recherche arrière (backtracking) à l'aide des variables $\psi_j(p,q)$. Enfin le troisième problème, c'est à dire l'apprentissage des probabilités des règles de la grammaire, peut être résolu en utilisant également les probabilités inside et outside. Nous n'aborderons pas ici ce problème.

2.2.4 Discussion

A la vue de cette brève présentation on remarque la grande similitude qui peut exister entre d'une part les GPR et les MMC, et d'autre part les MMC et les GPHC. En effet, les règles de transisition entre les états d'un modèle de Markov peuvent être représentées par un automate à états finis de la même manière qu'une Grammaire Régulière. On fait alors l'analogie entre les états cachés du modèle de Markov et les règles de la grammaire d'une part et les observations discrètes et les symboles terminaux d'autre part. La différence essentielle réside alors dans la prise en compte de deux processus stochastiques, le premier lié aux règles de transision entre états et le second lié au processus d'émision des observations à partir des états cachés. Pour achever l'analogie entre les deux modèles, on doit préciser le rôle analogue joué par les probabilités initiales d'un MMC et l'axiome d'une grammaire régulière qui permet de débuter une dérivation. De même un état de fin de chaîne doit être pris en compte. L'analogie entre les MMC et les GPHC quant à elle est principalement liée aux techniques de calcul mises en oeuvre plutot qu'à la structure des règles. Ce qui est tout à fait cohérent et complémentaire avec la première analogie que nous venons de mettre en évidence. Dans les deux cas l'ambiguïté dans l'application des règles est prise en compte de manière analogue en procédant selon les principes de la programmation dynamique. De plus pour ces deux modèles stochastiques il existe des procédures d'apprentissage des paramètres similaires (que nous n'avons pas présenté ici par manque de place). De manière générale on dispose donc de procédures efficaces pour répondre aux trois types de problèmes suivants : probabilité qu'une séquence (de mots ou d'observations) ait été générée par le modèle, détermination de la séquence de règles la plus probable et apprentissage des paramètres du modèle. Si les développements algorithmiques des deux modèles sont donc très voisins, il faut cependant souligner que la complexité des algorithme traitant les grammaires stochastiques hors contexte est d'un ordre supérieur à la complexité des algorithmes utilisés pour les MMC (complexité en $O(TN^2)$ N2 pour les MMC et complexité en $O(TN^3)$ pour les grammaires hors-contexte).

3 Extension au structures bidimensionnelles

Les modèles grammaticaux que nous venons de passer en revue, permettent de représenter des structures monodimensionnelles, telles que des chaînes. De même, les méthodes d'analyses associées à ces modèles sont fondamentalement monodimensionnelles. Dans les structures monodimensionnelles, les entités sont parfaitement ordonnées dans une direction unique. Dans les structures bidimensionnelles, les entités sont disposées spatialement, et une composante donnée ne possède donc pas un seul prédécesseur et un seul successeur (ce qui est le cas dans les chaînes), mais un certain nombres de voisins. Si l'espace est discrétisé (ce qui est le cas d'une image numérique) et que l'on considère un voisinage 4-connexe, un élément donné aura 4 voisins. Les grammaires de chaînes telles que nous venons de les voir ne permettent pas de modéliser des structures bidimensionnelles. Des formalismes bidimensionnels ont donc été proposés. Il s'agit essentiellement de grammaires de graphes, ou de modèles dérivés des grammaires de graphes (grammaires d'arbres, grammaires de web, grammaires plex)[FU 82][BLO 95]. Ces modèles ont été utilisés pour la reconnaissance de documents structurés, tels que formules mathématiques, diagrammes, schémas, ou comme formalisme de modélisation de la syntaxe dans les langages visuels [REK 95]. Cependant comme le souligne Couäsnon [COü 96], ces formalismes sont complexes et plutôt dédiés à l'analyse d'objets connexes. Une généralisation des grammaires de chaînes à l'aide d'opérateurs de position a également été proposée [COü 96]. De tels opérateurs de position permettent de modéliser une plus grande variété de relations entre les entités. L'avantage de ce type de formalisme est de permettre l'utilisation des techniques classiques d'analyse utilisées dans le cas des grammaires de chaînes, moyennant une certaine adaptation due au fait que l'on analyse des structures 2D. Dans le cas des modèles stochastiques, l'extension au 2D des modèles de Markov cachés est réalisée à l'aide de champs de Markov. Le principe de la programmation dynamique sur lequel est basé l'algorithme de Viterbi, est fondamentalement considéré comme monodimensionnel. Cependant, comme le montre Geoffrois, ce principe peut être naturellement étendu au cas 2D. Dans [GEO 04], cet extension du principe de programmation dynamique est appliqué à un problème de reconnaissance de chiffres manuscrits à l'aide de champs de Markov. Le principe de la programmation dynamique peut être appliqué dès lors que le problème à résoudre peut se décomposer récursivement en deux problèmes moins complexes. Le principe de programmation dynamique est également utilisé dans l'algorithme Inside pour l'analyse à l'aide de grammaires hors-contexte. Cette méthode peut donc théoriquement se généraliser au cas de l'analyse de structures bidimensionnelles. Dans le cas des chaînes, cet algorithme consiste à déterminer récursivement toutes les manières de factoriser une chaîne donnée de longueur n, à partir des factorisa-

tion des sous-chaînes de longueurs 1 à n, on considérant qu'une chaîne donnée peut se décomposer en deux sous-chaînes. Si on considère une chaîne de longueur 3, il y a 2 configurations possibles de décomposition de la chaîne en 2 sous chaînes. Dans le cas 2D, on ne considère pas des sous-chaînes, mais des sous-espaces du plan. L'algorithme fait l'hypothèse qu'un sous-espace peut se décomposer en 2. Seulement dans ce cas le nombre de configurations est beaucoup plus important que dans le cas des chaînes. Si on considère un sous-espace de dimension 2×2, il y a 6 façons différentes de décomposer cet espace en deux sous-espaces. Plus les sous-espaces considérés deviennent grand, plus le nombre de configurations devient important, et on est alors confronté à un problème d'explosion combinatoire.

4 Conclusion et perspectives

Nous nous intéressons à la segmentation des documents manuscrits non contraints. Les approches d'analyse de documents imprimés sont mises en échec sur ce type de documents du fait de la variabilité dans la disposition spatiale des entités physiques et parce que ces méthodes sont fondées sur des décisions locales non contextualisées. Si le formalisme des modèles génératifs (grammaires) semble bien adapté pour expliciter les connaissances d'organisation spatiales des documents, il apparaît en revanche insuffisant pour modéliser la variabilité des productions manuscrites. L'extension de ces formalismes dans leur version stochastique conduit soit aux Modèles de Markov Cachés (extension des grammaires régulières), soit aux grammaires stochastiques hors contexte (extension des grammaires hors contexte). Les grammaires stochastiques hors contexte semblent naturellement aptes pour modéliser les structures hiérarchiques présentes dans les documents. Néanmoins leur extension au cas bi-dimensionnel est d'un niveau de complexité très important qui nécessiterait des heuristiques d'élagage des solutions développées au cours du processus d'analyse. Si cette option reste envisageable on peut néanmoins se demander si elle est pertinente au regard de la difficulté à laquelle on est principalement confronté dans l'analyse des documents manuscrits. Ces documents ont en général une structure peu complexe et il y a assez peu d'ambiguïté dans la disposition spatiale des entités de haut niveau (paragraphes, annotations marginales, en bas de pages,...). L'ambiguïté qui constitue la difficulté principale dans l'analyse de ces documents semble résider principalement dans la localisation et la détection des lignes d'écriture, qu'il s'agisse des lignes d'écriture dans les marges, dans le corps du texte ou tout autre zone informative. Cette analyse nous conduit actuellement à privilégier un formalisme de grammaire régulière stochastique 2D tel que les champs de Markov pour la segmentation des documents manuscrits en lignes d'écriture. Ce choix permet de limiter la combinatoire en focalisant l'analyse sur les entités les plus difficiles à extraire en laissant toutefois la possibilité de soumettre l'analyse de la structure des meilleures solutions de segmentation en lignes d'écritures à une segmentation en zones informatives de plus haut niveau grace à un modèle de grammaire hors contexte. Dans une telle stratégie, la complexité de l'analyse grammaticale serait naturellement limitée puisque focalisée sur les meilleures hypothèses de segmentation en lignes.

Références

[ABU 95] ABUHAIBA I., HOLT M., DATTA S., Line Extraction and Stroke Ordering of Text Pages, *ICDAR95*, 1995, pp. 390-393.

[BLO 95] BLOSTEIN D., FAHMY H., GRBAVEC A., Practical use of graph rewriting, rapport n° 95-373, 1995, CDN.

[BOI 00] BOITE R., BOURLARD H., DUTOIT T., HANCQ J., LEICH H., *Traitement de la parole*, Presses Polytechniques et Universitaires Romandes, 2000.

[BRU 99] BRUZZONE E., COFFETTI M., An Algorithm for Extracting Cursive Text Lines, *ICDAR99*, 1999, pp. 749-752.

[COü 96] COÜASNON B., Segmentation et reconnaissance de documents guidees par la connaissance a priori : application aux partitions musicales, Thèse de Doctorat, Université de Rennes I, Janvier 1996.

[FU 82] FU K., *Syntactic Pattern Recognition and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[GEO 04] GEOFFROIS E., CHEVALIER S., PRÊTEUX F., Programmation dynamique 2D pour la reconnaissance de caractères manuscrits par champs de Markov, *RFIA, Reconnaissance de Formes et Intelligence Artificielle*, Toulouse, France, Janvier 2004.

[KIS 98] KISE K., SATO A., IWATA M., Segmentation of page images using the area voronoi diagram, *Computer Vision and Image Understanding*, vol. 70, 1998, pp. 370-382.

[KUN 00] KUNT M., CORAY G., GRANLUND G., HATON J., INGOLD R., KOCHER M., *Reconnaissance des formes et analyse de scènes*, Presses Polytechniques et Universitaires Romandes, 2000.

[LIK 94] LIKFORMAN-SULEM L., FAURE C., Extracting text lines in handwritten documents by perceptual grouping, FAURE C., KEUSS P., LORETTE G., WINTER A., Eds., *Advances in handwriting and drawing : a multidisciplinary approach*, Europia, Paris, 1994, pp. 117-135.

[LIK 95] LIKFORMAN-SULEM L., HANIMYAN A., FAURE C., A Hough based algorithm for extracting text lines in handwritten documents, *ICDAR95*, 1995, pp. 774-777.

[MAN 02] MANNING C. D., SCHÜTZE H., *Foundations of Statistical Natural Language Processing*, MIT Press, 2002.

[NAG 92] NAGY G., SETH S., VISWANATHAN M., A Prototype Document Image Analysis System for Technical Journals, *Computer*, vol. 25, n° 7, 1992, pp. 10-22.

[O'G 93] O'GORMAN L., The document spectrum for page layout analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, 1993, pp. 1162-1173.

[REK 95] REKERS J., SCHÜRR A., A Graph Grammar Approach to Graphical Parsing, *11th IEEE Symposium on Visual Languages*, may 1995, pp. 195-202.