Modélisation de HMMs en contexte avec des arbres de décision pour la reconnaissance de mots manuscrits

Anne-Laure Bianne, Christopher Kermorvant, Laurence Likforman-Sulem

Modélisation de HMMs en contexte avec des arbres de décision pour la reconnaissance de mots manuscrits

Anne-Laure Bianne[1,2], **Christopher Kermorvant**[1], **Laurence Likforman-Sulem**[2]

(1) *A2iA SA, Artificial Intelligence and Image Analysis*
40 bis rue Fabert, 75007 Paris, France
alb, ck @a2ia.com
(2) *Telecom ParisTech/TSI and CNRS LTCI*
46 rue Barrault, 75013 Paris, France
likforman@telecom-paristech.fr

RÉSUMÉ. *Cet article présente un système à base de HMMs pour la reconnaissance hors-ligne de mots manuscrits où les modèles de mots sont la concaténation des modèles des caratères les composant. Afin de prendre en compte les liaisons entre caractères, leurs modèles sont considérés dépendants de leur contexte et sont appelés trigraphes. Or, une telle modélisation augmente de manière considérable le nombre de paramètres à calculer. Ainsi, nous effectuons un partage des paramètres par un clustering sur chaque position d'état. Ce clustering est basé sur des arbres de décision qui ont l'avantage, en phase de test, de pouvoir associer un modèle connu à un trigraphe non appris. Nous avons testé notre système sur la base publique Rimes et l'avons comparé à un système à base de monographes. Notre système atteint 74% de mots correctement reconnus et dépasse ainsi les performances de l'état de l'art sur cette base.*

ABSTRACT. *This paper presents an HMM-based recognizer for the off-line recognition of handwritten words. Word models are the concatenation of context-dependent character models: the trigraphs. Due to the large number of possible context-dependent models to compute, a clustering is applied on each state position, based on decision trees. Our system is shown to perform better than a baseline context independent system, and reaches an accuracy higher than 74% on the publicly available Rimes database.*

MOTS-CLÉS : *reconnaissance d'écriture manuscrite, clustering d'états, arbres de décision.*

KEYWORDS: *off-line handwriting recognition, state position clustering, decision trees.*

1. Introduction

Nous présentons dans cet article un système à base de HMMs pour la reconnaissance de mots manuscrits. Une application directe de ce système est la lecture de document manuscrits numérisés : archives, courriers envoyés aux entreprises ou aux administrations publiques, etc. Etant donné la variabilité de tels documents, le système utilisé doit être robuste à tout type de scripteur, sans contrainte sur la forme des mots, ni sur le lexique. Les systèmes HMMs avec stratégie analytique sont particulièrement intéressants pour ce type de problème, car n'importe quel modèle de mot peut être construit par la concaténation des modèles des caractères le composant, sans avoir besoin d'exemples supplémentaires pour apprendre le mot.

Plusieurs systèmes HMMs ont été proposés pour cette tâche. La plupart d'entre eux utilisent l'approche par fenêtre glissante pour extraire des caractéristiques d'une image de mot (Vinciarelli *et al.*, 2001, Toselli, 2004, El-Hajj *et al.*, 2009, Rodriguez *et al.*, 2008). Ces systèmes diffèrent cependant dans le choix des caractéristiques à extraire et la forme de la fenêtre glissante. Les résultats obtenus pour l'alphabet Latin sont représentés en général sur deux grandes bases de données publiques, la base IAM (Marti *et al.*, 1999) et la base (plus récente) Rimes (Augustin *et al.*, 2006).

La structure HMM, appliquée originellement à la reconnaissance de la parole, peut être améliorée de plusieurs façons. La méthode la plus fréquente de raffinement consiste à modéliser des phonèmes dépendants de leur contexte pour illustrer l'effet de co-articulation. Certains travaux appliquent cette approche à la reconnaissance d'écriture manuscrite (Schüßler *et al.*, 1998, Natarajan *et al.*, 2006, Fink *et al.*, 2007) en remplaçant les phonèmes par des lettres. Les modèles de lettres augmentent alors leur degré de précision car les déformations possibles dues à l'écriture d'un même caractère dans des mots différents sont prises en compte. Cependant, un tel système voit son nombre de paramètres à calculer augmenter considérablement, et nécessite donc un très grand nombre de données d'apprentissage afin d'estimer correctement tous les contextes de tous les caractères. C'est pourquoi les systèmes à modèles contextuels utilisent des méthodes de clustering (de modèle, ou d'état, ou autre) afin de regrouper les contextes similaires. Notre approche diffère des méthodes usuelles à base de data-driven clustering car nous utilisons des arbres de décision modélisés à partir de connaissances humaines sur les propriétés morphologiques de caractères pour grouper nos paramètres. Cette approche permet en outre de traiter des contextes non appris. Ceci est très utile dans notre tâche de reconnaissance étant donné que de nombreux contextes du lexique de test ne sont pas présents dans la base d'apprentissage.

Afin de prouver l'efficacité de notre méthode, nous comparons deux systèmes : un premier système où les modèles de caractères sont indépendants du contexte (les monographes), et un second système basé sur des modèles en contexte, les trigraphes.

L'article s'organise ainsi : la Section 2.1 présente les étapes de prétraitement des données : nettoyage des images puis extraction de caractéristiques par approche à base de fenêtre glissante. La Section 2.2 présente le système classique avec des modèles hors contexte et la Section 3 introduit le système amélioré avec des modèles de carac-

tères dépendants de leur contexte, dont les états sont groupés en clusters grâce à des arbres de décision. Nous comparons ces deux systèmes à base de fenêtres glissantes à des travaux existants dans le domaine en Section 4. Les expériences ont été produites sur la base Rimes 2008 (Grosicki *et al.*, 2009).

2. Système de reconnaissance d'écriture à base de HMMs hors contexte

2.1. *Pré-traitements et extraction de caractéristiques*

Les premières étapes d'un système de reconnaissance consistent à préparer les données : normaliser les images en entrée, et en extraire les caractéristiques qui permettront de calculer les modèles. Nous limitons l'étape de la normalisation à la correction de l'inclinaison des caractères au niveau du mot. Les images ne sont pas normalisées en taille car les caractéristiques extraites par la suite sont indépendantes de la hauteur de l'image, et la modélisation HMM est robuste à un nombre variable de vecteurs de caractéristiques correspondant à un même modèle. Afin de calculer des caractéristiques propres aux ascendants et descendants, les lignes de base des mots sont calculées. Nous nous sommes inspirés du travail de Vinciarelli (Vinciarelli *et al.*, 2001) pour notre correction d'inclinaison et notre calcul de lignes de base.

L'extraction de caractéristiques s'inspire des travaux de El-Hajj (El-Hajj *et al.*, 2005, El-Hajj *et al.*, 2009) et utilise des fenêtres glissantes parcourant l'image de gauche à droite. Les fenêtres sont divisées verticalement en un nombre fixe de cellules. Dans chaque fenêtre sont extraites N_f caractéristiques : 2 sont reliées au nombre de transitions caractère/arrière-plan, 12 aux configurations de pixels (calcul de concavités), une caractéristique est dérivative et les autres sont reliées aux densités de pixels. Certaines de ces caractéristiques dépendent de la position des lignes de base. Les images en niveau de gris sont binarisées avec la méthode de Otsu (Otsu, 1979) afin de calculer les caractéristiques de transition caractère/arrière-plan ainsi que celles de configuration de pixels.

Nous avons optimisé la largeur w des fenêtres glissantes, le décalage δ entre deux fenêtres consécutives et le nombre de cellules c dans chaque fenêtre sur une base de validation indépendante. Les paramètres obtenus sont $w = 8$ pixels, $\delta = 4$ pixels et $c = 20$ cellules. Ce choix entraîne donc un nombre total de caractéristiques $N_f = 28$ à calculer dans chaque fenêtre.

2.2. *Modélisation de caractères indépendamment de leur contexte*

Notre système de base est générique et ne prend pas en compte les contextes. Il utilise une stratégie analytique et procède sans segmentation. Un mot est modélisé par la concaténation des caractères le composant. Tous les modèles de caractères partagent la même topologie : 8 états émetteurs, transitions gauche-droite et saut d'état autorisé. La densité de probabilité des observations est un mélange de 20 distribu-

tions gaussiennes. Les modèles HMM sont appris avec l'algorithme Baum-Welch et le décodage se fait avec l'algorithme Viterbi. Nous utilisons le Hidden Markov Model Toolkit (HTK (Young *et al.*, 2006)) pour l'apprentissage et le décodage.

Etant donné que les modèles de notre système de base sont indépendants des contextes, chaque caractère a un unique modèle lui correspondant. Nous appelons ces modèles des 'monographes'. Nous définissons un total de 78 monographes différents (chiffres, lettres, ou symboles), en prenant en compte la casse (a est différent de A) et les accents (*é* est différent de e).

3. Système de reconnaissance d'écriture à base de modèles HMMs dépendants du contexte

Les HMMs ont été largement utilisés pour la reconnaissance de la parole, et cet usage a permis d'élaborer des modèles de plus en plus précis. Etant donné que le parallèle parole/écrit a souvent été intéressant, nous avons essayé d'appliquer un outil spécifique à la parole à notre problème de reconnaissance d'écriture manuscrite : les modèles en contexte (Lee, 1990). En parole, un même son a une prononciation différente selon les sons précédents et suivants. C'est le phénomène de co-articulation. Afin de modéliser ces changements, des modèles spécifiques sont créés pour les phonèmes en contexte. La transposition de ce modèle à la reconnaissance de l'écriture manuscrite nous conduit à modéliser chaque caractère en fonction de son contexte gauche et droit. Un inconvénient majeur de la construction de modèles en contexte est que le nombre de paramètres des HMMs à calculer peut devenir très important, chaque caractère pouvant avoir une multitude de contextes différents, et le nombre de données d'apprentissage devient rapidement trop faible. Une solution pour réduire ce nombre est le partage de paramètres entre différents modèles.

Dans cet article, nous proposons de construire des modèles de trigraphes (un monographe avec ses contextes gauche et droit), et de partager certains paramètres entre ces trigraphes. Deux types de paramètres sont partagés : les matrices de transition sont liées, et des clusters d'états sont construits avec une méthode originale à base d'arbres de décision.

(a) "i" et "n" dans le mot "Monsieur"

(b) "i" et "n" dans le mot "distinguée"

Figure 1. *Illustration des différentes formes que peut prendre un caractère selon son contexte : "i" et "n" ont une forme différente pour chaque apparition dans les mots ci-dessus.*

3.1. *Modélisation de caractères en contexte*

Le principe de modélisation de caractères en contexte est illustré en Figure 1 : notre façon d'écrire un caractère évolue avec les lettres adjacentes à écrire. De plus, étant donné que les deux mots de la Figure 1 on été écrits par le même scripteur, on peut imaginer la variabilité des contextes qui peut exister dans une base de données multi-scripteur de grande taille, telle que la base Rimes.

Cette modélisation consiste à définir un mot non pas comme une succession de caractères, mais comme une succession de caractères avec leurs contextes. Les monographes de la Section 2.2 sont remplacés par des trigraphes. Etant donné que nous avons utilisé HTK (Young *et al.,* 2006) pour nos expériences, nous conserverons la syntaxe de cet outil. Ainsi, un contexte gauche est suivi d'un signe moins '-', et un contexte droit est précédé d'un signe plus '+'. La lettre i de la Figure 1.a, entourée d'un s et d'un e s'écrit donc, en contexte : $s - i + e$.

La procédure d'apprentissage des monographes pourrait s'appliquer directement aux trigraphes, mais pour les 4500 trigraphes présents dans la base d'apprentissage, certains, avec peu d'exemples, seront mal appris. De plus, si nous choisissions une topologie finale de 8 états émetteurs et un mélange de 20 gaussiennes par état, il y aurait presque 1 million de distributions à calculer, ce qui est prohibitif. C'est pourquoi nous considérons à présent le clustering de paramètres, afin de pallier le problème du manque de données d'apprentissage ainsi que celui du trop grand nombre de modèles à calculer.

3.2. *Clustering de paramètres HMM*

3.2.1. *Organisation*

Il existe plusieurs moyens de lier ou regrouper les paramètres des modèles de trigraphes. Dans cette Section, nous présentons notre système à trois étapes, illustré sur la Figure 2.

Les premiers paramètres HMMs partagés sont les matrices de transition. Nous avons observé qu'une fois la topologie d'un modèle choisi (gauche-droit, saut d'état, etc), des modifications sur les coefficients de la matrice de transition ont une très faible influence sur la phase de reconnaissance. Considérant cette information, nous avons alors imposé que les trigraphes ayant la même racine ($* - a + *$ ou $* - b + *$ ou $* - c + *$, etc.) partageront la même matrice de transition. Ainsi, le nombre de matrices à calculer se réduit au nombre initial de monographes, soit 78. La fin de cette première étape consiste alors à lister tous les trigraphes présents dans le lexique de la base d'apprentissage, copier les monographes initialisés (avec 8 états et 1 distribution gaussienne par état) dans les trigraphes correspondants, et passer l'algorithme Baum-Welch sur tous ces trigraphes pour les initialiser, tout en conservant la contrainte sur leur matrice de transition. Cette première étape est illustrée dans la partie *"duplication*

et contrainte sur les matrices de transition" de la Figure 2 pour le caractère b et tous les trigraphes $* - b + *$ centrés en b présents dans l'ensemble d'apprentissage.

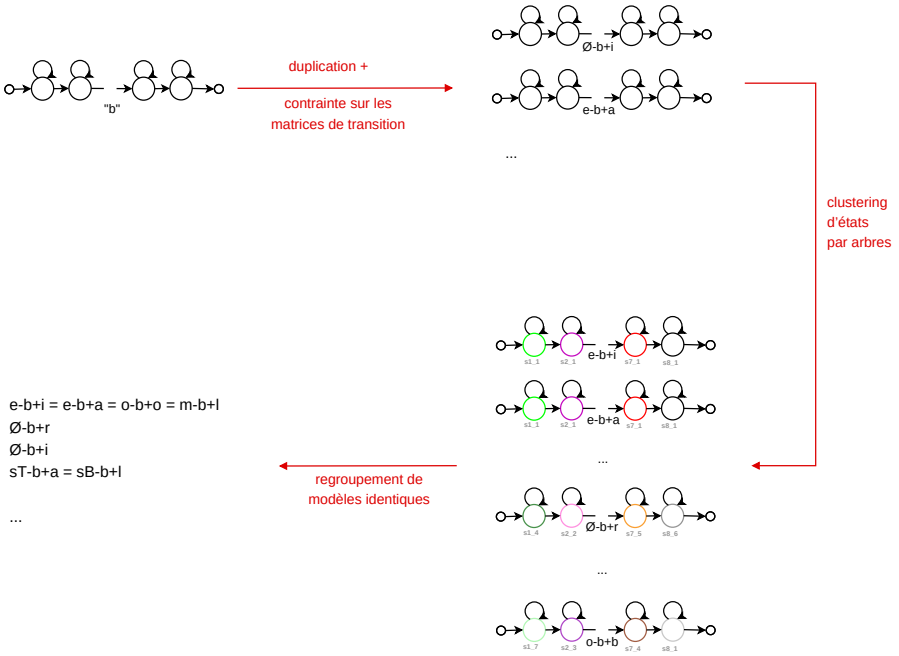

Figure 2. *Illustration des étapes du système de clustering des paramètres des modèles en contexte pour la lettre b.*

La seconde étape de notre système est le clustering d'états appliqué aux trigraphes. Fink et al.(Fink *et al.*, 2007) et Natarajan et al. (Natarajan *et al.*, 2006) ont montré l'efficacité de ce type de groupement. Nous différons de ce qu'il proposent dans l'utilisation d'arbres de décision pour calculer nos clusters. Nous expliquons en détail notre méthode en Section 3.2.2.

Le principe du clustering état par état est que, pour les trigraphes associés à une lettre centrale donnée, tous les états correspondant à une même position dans le modèle HMM sont soumis à un clustering. Soit c la lettre centrale considérée. N_c trigraphes existent, centrés sur c. Donc pour chaque position d'état i des trigraphes $* - c + *$, $1 \leq i \leq 8$, N_c différents états devraient être calculés. L'utilisation de clusters d'états permet la réduction du nombre de modèles à calculer. Ainsi pour la position d'état numéro i, $n_i \leq N_c$ clusters sont sélectionnés pour représenter tous les états de la position i, définissant n_i différent modèles d'états $s_{i,j}$, $1 \leq j \leq n_i$. Un état en position i prend alors sa valeur dans l'un des $s_{i,j}$ modèles. Nous rappelons que pour des raisons de complexité, un modèle d'état est représenté par une seule distribution gaussienne durant la phase d'initialisation des trigraphes.

Cette étape est illustrée dans la partie *"clustering par arbres"* de la Figure 2 pour la lettre b. Elle nous a permis de réduire le nombre de paramètres d'un facteur 10.

Finalement, étant donné que le clustering d'états entraîne une réduction importante du nombre de paramètres différents, certains trigraphes se retrouvent partager les mêmes états, et donc être identiques. Nous groupons donc dans une dernière étape les trigraphes identiques, ce qui permet de réduire encore le nombre de modèles, le divisant par trois.

3.2.2. *Arbres de décision pour clustering d'états de trigraphes*

Le clustering à base d'arbres de décision est une alternative au clustering guidé par les données. Il se base sur une autre mesure de distance entre modèles (maximisation de la vraisemblance) et permet une découpe de clusters plus contrôlée. Ce type de clustering a d'abord été proposé pour la parole (Young *et al.*, 1994) où il donne des résultats au niveau de l'état de l'art avec un système moins complexe que d'autres avec modèles en contexte. Le groupement et la séparation de clusters d'états sont conduits par un arbre binaire dont chaque noeud correspond à une question rhétorique sur les caractéristiques morphologiques des caractères. Dans notre cas, les questions sont définies sur le comportement des contextes gauche et droit. Un arbre est construit pour chaque position d'état de tous les trigraphes correspondant à une même lettre centrale. Nous définissons ainsi 78 *lettres centrales* * 8 *états par caractère* = 624 arbres différents.

Pour construire un arbre, tous les états correspondant à une même position pour une lettre centrale donnée sont d'abord regroupés au noeud racine. Ensuite, la question binaire qui sépare le groupe en deux sous-groupes de vraisemblance maximale est choisie, et la séparation est effectuée, créant deux nouveaux noeuds. Il en va ainsi de la scission de chaque noeud jusqu'à ce que l'augmentation de la vraisemblance descende sous un seuil, ou bien jusqu'à ce que plus aucune question ne soit disponible qui crée deux sous-groupes avec un taux d'occupation d'états suffisant.

On peut formuler mathématiquement le clustering par arbres (Zen *et al.*, 2003, Young *et al.*, 1994). Si $\boldsymbol{S}$ est l'ensemble d'états présent dans le noeud courant, alors la séparation de $\boldsymbol{S}$ en deux sous-groupes $\boldsymbol{S_{q+}}$ et $\boldsymbol{S_{q-}}$ est faite par la question q qui maximise :

$$\begin{aligned}\Delta L_q &= -\frac{1}{2}\{\Gamma(\boldsymbol{S}_{q+})log(|\Sigma(\boldsymbol{S}_{q+})|) + \Gamma(\boldsymbol{S}_{q-})log(|\Sigma(\boldsymbol{S}_{q-})|) \\ &\quad -\Gamma(\boldsymbol{S})log(|\Sigma(\boldsymbol{S})|)\} - n\Gamma(\boldsymbol{S_0}) \qquad [1]\end{aligned}$$

$\Gamma(\boldsymbol{S})$ est le taux d'occupation d'états cumulé du noeud contenant $\boldsymbol{S}$, $|\Sigma(\boldsymbol{S})|$ est le déterminant de la matrice de covariance partagée par tous les états contenus dans $\boldsymbol{S}$, et n est la dimension des vecteurs de caractéristiques. $\boldsymbol{S_0}$ représente l'ensemble initial d'états présents au noeud racine. La découpe en deux sous-groupes se fait à condition que $\Gamma(\boldsymbol{S}_{q+})$ et $\Gamma(\boldsymbol{S}_{q-})$ soient au-dessus du taux minimal d'occupation, et que ΔL_q

soit plus grand qu'un seuil minimal choisi. Les seuils minimaux pour $\Gamma(\boldsymbol{S})$ et ΔL_q ont été validés et fixés sur une base de validation indépendante.

Un exemple d'arbre de décision est donnée en Figure 3. Il représente l'arbre calculé pour la position d'état numéro deux des trigraphes centrés sur la lettre b.

Les questions définissant les arbres de décision pour la reconnaissance de la parole on été réalisées par des experts (Chelba *et al.*, 2002). A notre connaissance, personne n'utilise de tels arbres pour l'écrit. Nous avons donc proposé des questions en rassemblant des caractères semblables dans différents contextes. L'ensemble des questions que nous avons créées contient des questions générales, pour permettre des clusters larges, mais aussi des questions précises, au cas où des clusters très fins doivent être créés, et des questions intermédiaires. Les questions sont uniquement fonctions des contextes (gauche et droit). Par exemple :

– (question générale) est-ce que le contexte droit (gauche) est une majuscule ? une minuscule ? Contient-il un ascendant ?

– (question intermédiaire) le contexte contient-il une boucle ("o", "b", "d", etc.) ? Ou une barre verticale ("t", "p", etc.) ?

– (question précise) est-ce que le lien avec la lettre suivante (précédente) se situe sur la ligne de base basse ("a", "c", etc.) ou sur la ligne de base haute ("v", "w", etc.) ?

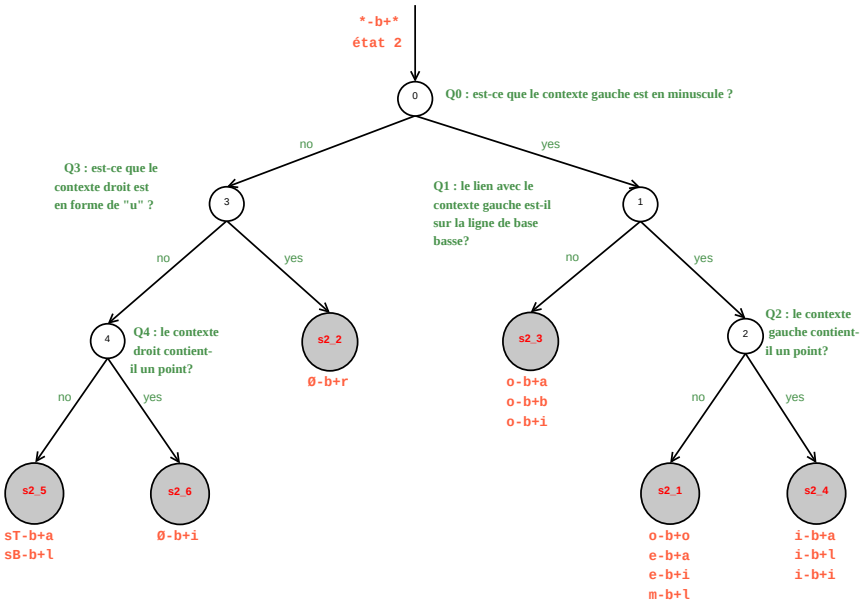

Figure 3. *Exemple d'arbre de décision pour le clustering d'états : l'ordre des questions et les clusters sont associés à un état donné (ici l'état numéro 2) de tous les trigraphes* $- * b + *$.

3.2.3. *Apprentissage et décodage avec modèles en contexte*

La première étape du processus d'apprentissage consiste à établir les modèles de trigraphes comme présenté dans la section précédente. Les modèles des trigraphes sont ensuite ré-estimés en utilisant l'algorithme Baum-Welch. Le nombre de gaussiennes dans le mélange est augmenté graduellement pour arriver à 20 distributions gaussiennes par état. Cette topologie permet de comparer directement ce nouveau système au système de base présenté en Section 2.2.

L'utilisation d'arbres de décision pour constituer les clusters d'états pour modèles en contexte apporte une fonctionnalité supplémentaire très utile lors du décodage : elle permet de sélectionner les clusters qui serviront à modéliser les états des trigraphes non vus lors de l'apprentissage. Par exemple, le lexique de test de la base Rimes 2008 est différent du lexique d'apprentissage. Ainsi, plusieurs nouveaux mots, et donc plusieurs nouveaux trigraphes apparaissent, qui n'ont pas été appris. Le clustering par arbres permet de les modéliser car il peut adapter les modèles appris à n'importe quel vocabulaire (basé sur les mêmes monographes de départ).

La modélisation d'un trigraphe inconnu se fait de la manière suivante : chaque état du trigraphe est positionné à la racine de l'arbre correspondant au même numéro d'état et à la même lettre centrale. Ensuite, chaque état parcourt son arbre, répondant aux questions sur les contextes du nouveau trigraphe, jusqu'à atteindre un noeud où est positionné un cluster. Le modèle d'état représentant le cluster est alors le modèle assigné au numéro d'état considéré. La capacité des arbres de traiter des trigraphes non vus nous a permis de modéliser plusieurs centaines de nouveaux trigraphes dans la base de test de Rimes.

4. Expériences

4.1. *La base de données Rimes*

La base de données Rimes a été rendue disponible en 2006 (Augustin *et al.*, 2006). Des exemples sont donnés en Figure 4. Elle rassemble plus de 12500 documents manuscrits écrits par environ 1300 volontaires. Elle a été créée pour répondre au besoin récurrent de bases complètes d'images propres et de taille suffisante. Aucune contrainte n'a été imposée aux scripteurs, les documents sont donc très variables, ce qui rend la base réaliste.

Des campagnes d'évaluation ont eu lieu depuis 2007, ce qui nous permet de nous comparer à l'état de l'art. Nous utilisons dans cet article les données et la procédure de la campagne Rimes 2008 (Grosicki *et al.*, 2009). L'ensemble d'apprentissage est composé de 44196 images de mots isolés pour un lexique de 4000 mots différents. L'ensemble de test est quant à lui composé de 7542 images de mots isolés pour un lexique de 1636 mots. Plus de 200 mots du lexique de test n'existent pas dans le lexique d'apprentissage.

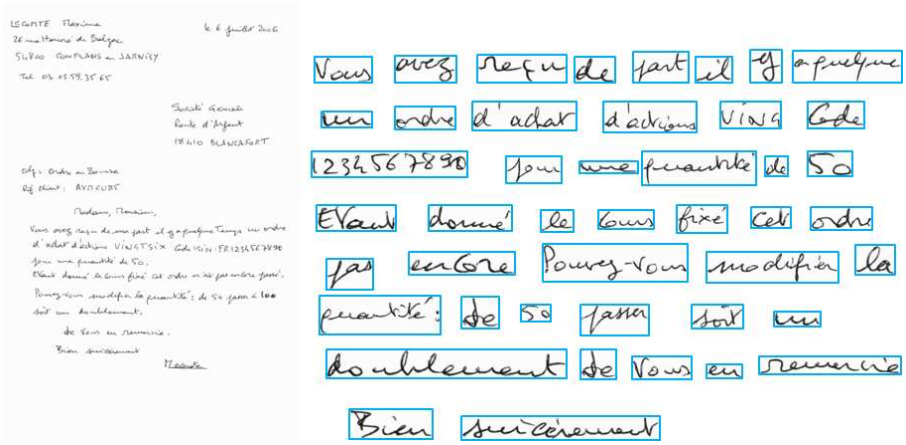

Figure 4. *Quelques exemples de la base Rimes*

4.2. *Résultats*

Nous présentons dans cet article deux systèmes différents :

système 1 : CL (classique)

Ce système correspond à celui présenté en Section 2. Le nombre de modèles de lettres est de 78. Les modèles du silence, de l'apostrophe, de la barre oblique et du trait d'union sont simplifiés à deux états émetteurs et une distribution gaussienne par état. Tous les autres modèles (lettres et chiffres) ont une topologie de 8 états émetteurs et un mélange de 20 distributions gaussiennes par état.

système 2 : MDC (modèles dépendants du contexte)

Ce système correspond à celui présenté en Section 3. Le nombre de trigraphes différents dans la base d'apprentissage est de 4784. Après le clustering d'états, ce nombre est réduit à 2196. Les modèles HMMs de trigraphes sont calculés à partir de 3615 modèles d'états différents. Chaque état contient un mélange de 20 distributions gaussiennes, sauf ceux correspondant aux lettres centrales {apostrophe, barre oblique, trait d'union}, qui n'en contiennent qu'une. Le modèle du silence n'est pas contextualisé. Les topologies finales sont donc les mêmes que celles du système CL. Lors de la phase de test, 364 nouveaux trigraphes on été modélisés avec les arbres de décision.

Les résultats des systèmes sont calculés avec la procédure de Rimes 2008. Une erreur de casse ou d'accent est comptée. Les taux d'erreurs sont donnés avec un intervalle de confiance binomial à 95% dans la Table 1.

Il est clair que le système basé sur des modèles en contexte donne de meilleurs résultats que le système HMM sans contexte (p-value $< 2.2 \times 10^{-16}$ pour un test binomial bilatéral à 95%). L'amélioration est de plus de 6% en absolu, ce qui correspond

système	taux de reconnaissance
CL	$67.5\% \pm 1.1$
MDC	$74.1\% \pm 1.0$

Tableau 1. *Comparaison des deux systèmes présentés sur la base test de Rimes 2008*

à une différence de plus de 450 images bien décodées. Il est intéressant de noter qu'à moins de 10 images près, les mots correctement décodés par le système CL sont également bien décodés par le système MDC. Le système avec contextes ne crée donc pas de nouvelles faiblesses par rapport au système de base, et résoud en plus certaines difficultés.

Il est intéressant de comparer nos résultats à l'état de l'art, reporté par Grosicki et al. (Grosicki *et al.,* 2009). La table 2 montre la comparaison avec le meilleur système à base de HMMs présenté à la compétition 2008. Proposé par le Litis (Kessentini *et al.,* 2008), il est basé sur des HMMs multi flux et utilise deux séquences de vecteurs de caractéristiques. Notre système avec trigraphes et clustering d'états par arbres de décision dépasse de 2.2% en absolu ce système. La différence est statistiquement significative, avec un test binomial bilatéral à 95% (p-value $= 0.002$).

système	taux de reconnaissance
MDC	$\mathbf{74.1\%}$
Litis	72.5%
CL	67.5%

Tableau 2. *Comparaison des systèmes proposés à l'état de l'art de la base de données Rimes test 2008*

4.3. *Travaux relatifs aux modèles en contexte*

Les modèles dépendants des contextes ont d'abord été proposés en reconnaissance de la parole. Cette amélioration dans la modélisation HMM a ensuite été appliquée à l'écrit pour la reconnaissance en ligne d'écriture (Kosmala *et al.*, 1997, Tokuno *et al.,* 2002, Rigoll *et al.,* 1998). A notre connaissance, l'utilisation de modèles en contexte pour la reconnaissance hors ligne d'écriture n'a été citée que dans 4 travaux (Schüßler *et al.*, 1998, Natarajan *et al.,* 2006, Fink *et al.,* 2007, El-Hajj *et al.,* 2008). Les principaux problèmes évoqués dans la littérature sont le nombre de données d'entraînement requis, et le très quand nombre de modèles à créer. Ces problèmes peuvent être résolus en utilisant des modèles semi-continus, ou en faisant du clustering de paramètres.

Schussler et al. (Schüßler *et al.,* 1998) décrivent un système avec modèles en contextes à base de HMMs. Toutes les sous-unités de mots (allant du caractère au mot entier) sont modélisées et hiérarchisées. Les modèles qui n'ont pas suffisamment

d'exemples sont éliminés. Le système est testé sur un lexique dynamique de taille réduite.

Natarajan et al. (Natarajan *et al.*, 2006) proposent un groupement d'états en fonction de leur position. Un ensemble de 128 distributions gaussiennes est défini pour chaque position d'état de chaque caractère. Leur procédure est relativement similaire à notre système, mais l'utilisation du clustering guidé par les données rend leur système instable face aux mots hors du vocabulaire d'apprentissage.

Plus récemment, Fink et al. (Fink *et al.*, 2007) ont aussi proposé un système avec des modèles en contexte et un data-driven clustering état par état. Ils montrent qu'ils peuvent améliorer leur résultats (taux d'erreur au niveau mot de 24.0% à 22.7%). Cependant, ils jugent que temps et la puissance de calcul additionnels requis ne valent pas les gains en performance obtenus.

Finalement, El Hajj et al. (El-Hajj *et al.*, 2008) font du clustering de modèles pour la reconnaissance de l'arabe : les trigraphes précédés (ou suivis) par un caractère contenant un descendant (ou un ascendant) sont regroupés.

Notre système avec modèles en contextes propose un regroupement de paramètres, tout comme le proposent Fink et al.(Fink *et al.*, 2007) et Natarajan et al.(Natarajan *et al.*, 2006). Mais nous effectuons notre clustering avec des arbres de décision, dont les questions ont été définies a priori en se basant sur la morphologie des caractères. Notre approche améliore les performances dans une plus grande proportion que les approches présentées ci-dessus : la différence entre un système classique et celui avec contextes est de 6.6% en absolu, ce qui représente une réduction relative de plus de 20% du taux d'erreur.

5. Conclusion

Nous avons présenté un système basé sur des modèles HMMs dépendants de leur contexte ainsi qu'un clustering d'états pour la reconnaissance hors-ligne de mots manuscrits. Les HMMs ne modélisent plus des caractères isolés mais des trigraphes, ce qui augmente considérablement le nombre de modèles à calculer. Ce nombre est réduit par un clustering d'états à l'aide d'arbres de décision. Ce type de clustering permet en outre de gérer des lexiques de test contenant des mots et des trigraphes non vus à l'entraînement, et de les associer à un modèle connu.

Comparé à un système HMM classique, la performance est améliorée de 6.6% en absolu, ce qui correspond à réduction relative de 20% du taux d'erreur. De plus, notre système en contexte donne de meilleurs résultats que l'état de l'art publié sur la base Rimes 2008 (Grosicki *et al.*, 2009).

6. Bibliographie

Augustin E., Carre M., Grosicki E., Brodin J.-M., Geoffrois E., Preteux F., « Rimes evaluation campaign for handwritten mail processing », *Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition - IWFHR06*, p. 231-235, October, 2006.

Chelba C., Morton R., « Mutual Information Phone Clustering for Decision Tree Induction », *Proceedings of the International Conference on Spoken Language Processing -ICSLP02*, 2002.

El-Hajj R., Likforman-Sulem L., Mokbel C., « Arabic Handwriting Recognition Using Baseline Dependant Features and Hidden Markov Modeling », *Proceedings of the Eighth International Conference on Document Analysis and Recognition - ICDAR05*, p. 893-897, 2005.

El-Hajj R., Likforman-Sulem L., Mokbel C., « Combining Slanted-Frame Classifiers for Improved HMM-Based Arabic Handwriting Recognition », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

El-Hajj R., Mokbel C., Likforman-Sulem L., « Recognition of Arabic handwritten words using contextual character models », *Proceedings of the IST & SPIE Conference on Document Recognition and Retrieval XV*, January, 2008.

Fink G., Plotz T., « On the Use of Context-Dependent Modeling Units for HMM-Based Offline Handwriting Recognition », *International Conference on Document Analysis and Recognition - ICDAR07*, vol. 2, p. 729-733, 2007.

Grosicki E., Carre M., Brodin J.-M., Geoffrois E., « Results of the second Rimes evaluation campaign for handwritten mail processing », *Proceedings of the 10th International Conference on Document Analysis and Recognition ICDAR09*, 2009.

Kessentini Y., Paquet T., Benhamadou A., « A Multi-Stream HMM-based Approach for Offline Multi-Script Handwritten Word Recognition », *Proceedings of the 11th International Conference on Frontiers in Handwriting Recognition - ICFHR08*, 2008.

Kosmala A., Rottland J., Rigoll G., « Improved On-Line Handwriting Recognition Using Context Dependent Hidden Markov Models », *Proc. Int. Conference on Document Analysis and Recognition - ICDAR97*, p. 641-644, 1997.

Lee K.-F., « Context-dependent phonetic hidden Markov models for speaker-independent continuous speech recognition », *Readings in speech recognition*, vol. 1, p. 347-366, 1990.

Marti U., Bunke. H., « A full English sentence database for off-line handwriting recognition », *Proceedings of the 5th International Conference on Document Analysis and Recognition - ICDAR99*, p. 705-708, 1999.

Natarajan P., Saleem S., Prasad R., MacRostie E., Subramanian K., « Multi-lingual Offline Handwriting Recognition Using Hidden Markov Models : A Script-Independent Approach », *Summit on Arabic and Chinese Hanwriting - SACH06*, 2006.

Otsu N., « A Threshold Selection Method from Grey-Level Histograms », *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, n° 1, p. 62-66, January, 1979.

Rigoll G., Kosmala A., Willett D., « An Investigation Of Context-Dependent And Hybrid Modeling Techniques For Very Large Vocabulary On-Line Cursive Handwriting Recognition », *Proceedings of Sixth Int. Workshop on Frontiers in Handwriting Recognition - IWFHR98*, 1998.

Rodriguez J. A., Perronnin F., « Local gradient histogram features for word spotting in unconstrained handwritten documents », *Proceedings of the 1st International Conference on Handwriting Recognition - ICFHR08*, August, 2008.

Schüßler M., Niemann H., « A HMM-based System for Recognition of Handwritten Address Words », *Proceedings of Sixth Int. Workshop on Frontiers in Handwriting Recognition - IWFHR98*, p. 505-514, 1998.

Tokuno J., Inami N., Matsuda S., Nakai M., Shimodaira H., Sagayama S., « Context-Dependent Substroke Model for HMM-Based On-Line Handwriting Recognition », *Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition - IWFHR02*, 2002.

Toselli A. H., Reconocimiento de Texto Manuscrito Continuo, PhD thesis, Departamento de Sistemas Informaticos y Computacion, Universidad Politecnica de Valencia, 2004.

Vinciarelli A., Luettin J., « A new normalization technique for cursive handwritten words », *Pattern Recognition Letters*, vol. 22, n° 9, p. 1043-1050, 2001.

Young S., al., *The HTK Book V3.4*, Cambridge University Press, Cambridge, UK, 2006.

Young S. J., Odell J. J., Woodland P. C., « Tree-based state tying for high accuracy acoustic modelling », *Proceedings of the workshop on Human Language Technology (HLT94)*, p. 307-312, 1994.

Zen H., Tokuda K., Kitamura T., « Decision tree based simultaneous clustering of phonetic contexts, dimensions, and state positions for acoustic modeling », *Proceedings of the 8th European Conference on Speech, Communication and Technology*, p. 3189-3192, 2003.