

Modèles de Markov Cachés et Modèle de Longueur pour la Reconnaissance de l'Écriture Arabe à Basse Résolution

Fouad Slimane^{1,2}, Slim Kanoun², Jean Hennebert^{1,3}, Adel M. Alimi² et Rolf Ingold¹

1 : Université de Fribourg, Faculté des sciences, Bd. de Pérolles 90, 1700 Fribourg – Suisse.

2 : Université de Sfax, Ecole Nationale d'Ingénieurs de Sfax, BP. 1173, 3038 - Sfax – Tunisie.

3 : Haute école spécialisée de Suisse occidentale, HES-SO, TechnoArk 3, 3960 Sierre – Suisse.

Contact : Fouad.Slimane@unifr.ch, Slim.Kanoun@yahoo.fr,
Jean.Hennebert@hevs.ch, Adel.Alimi@ieee.org, Rolf.Ingold@unifr.ch

Résumé

Nous présentons dans ce papier un système de reconnaissance automatique de l'écriture arabe à vocabulaire ouvert, basse résolution, basé sur les Modèles de Markov Cachés. De tels modèles sont très performants lorsqu'il s'agit de résoudre le double problème de segmentation et de reconnaissance pour des signaux correspondant à des séquences d'états différents, par exemple en reconnaissance de la parole ou de l'écriture cursive. La spécificité de notre approche est dans l'introduction des modèles de longueurs pour la reconnaissance de l'Arabe imprimé. Ces derniers sont inférés automatiquement pendant la phase d'entraînement et leur implémentation est réalisée par une simple altération des modèles de chaque caractère composant les mots. Dans notre approche, chaque mot est représenté par une séquence des sous modèles, ces derniers étant représentés par des états dont le nombre est proportionnel à la longueur de chaque caractère. Cette amélioration, nous a permis d'augmenter de façon significative les performances de reconnaissance et de développer un système de reconnaissance à vocabulaire ouvert. L'évaluation du système a été effectuée en utilisant la boîte à outils HTK sur une base de données d'images synthétique à basse résolution.

Abstract

We present a system for automatic recognition of printed Arabic text in open vocabulary mode, low resolution. This system is based on Hidden Markov Models. Such models have shown to be particularly successful when it comes to solve the double problem of segmenting and recognizing signals corresponding to sequences of different states, such as recognition of speech or cursive writing. The specificity of our approach is the introduction of minimum duration models for printed Arabic text recognition. The minimum duration models are automatically inferred during the training phase and their implementation is achieved by a simple alteration of character models of each component words. In our approach, each word is represented by a sequence of sub models, which are represented by states whose number is proportional to the length of each character. This improvement allows us to increase significantly the recognition performance and to build a system working in open vocabulary mode. The evaluation of system is performed using HTK on a database of word images that are synthetically generated in low resolution.

Mots-clés : OCR, arabe, MMC, segmentation, reconnaissance.

Keywords: OCR, Arabic, HMM, segmentation, recognition.

1. Introduction

La reconnaissance de l'écriture arabe remonte aux années 70. La majorité des solutions proposées a été testée sur l'écriture latine puis appliquée telle quelle pour la reconnaissance de l'écriture arabe imprimée [18]. Ces méthodes supposent généralement que les caractères peu-

vent être isolés par une étape de segmentation. Cette étape de segmentation est possible dans le cas d'un texte latin imprimé, mais très difficile dans le cas de l'écriture cursive ou semi-cursive, le cas de l'écriture arabe.

Les deux principales méthodes qui ont été utilisées sont l'approche globale et l'approche analytique. La première modélise le mot dans sa globalité. Elle présente l'avantage de garder le caractère dans son contexte de voisinage mais son inconvénient est qu'elle est limitée à la reconnaissance de vocabulaire très réduit et statique [1,4]. L'approche analytique quand à elle se base sur la modélisation de l'alphabet de la langue et la segmentation du mot en entités représentant un caractère ou un pseudo-caractère. La reconnaissance du mot consiste à identifier ces entités et à proposer des hypothèses de mots. Cette approche est très liée aux résultats de la segmentation [12] et elle présente l'avantage de pouvoir manipuler un vocabulaire ouvert.

Cependant, la difficulté majeure est la segmentation à priori du mot en fragments surtout lorsque il s'agit des données à basse résolution. Plusieurs approches ont été testées, mais toujours avec un succès limité [23]. Pour s'affranchir des problèmes de segmentation dus à la complexité de l'écriture arabe, d'autres approches spécifiques ont été proposées à savoir l'approche pseudo-globale fondée sur la notion de pseudo-mot [9], les approches analytiques qui intègrent la structure morphologique du vocabulaire en post-traitement pour valider les hypothèses de mots et proposer des scénarios de correction [2, 22], l'approche affixale modélisant les segments linguistiques du mot à préfixe, suffixe, infixes et racine [21] et l'approche neuro-linguistique, modélisant les concepts linguistiques du vocabulaire en terme de réseaux de neurones [10].

Face aux limites des approches existantes pour la reconnaissance de l'écriture arabe et aux difficultés de segmentation des mots arabes surtout à basse résolution, de nombreuses recherches ([7,8,11,12,13,14,15,16,17,18]) se sont orientées vers l'utilisation des méthodes stochastiques et en particulier vers les Modèles de Markov Cachés (MMCs) [19] pour la reconnaissance de mots et de textes.

Les MMCs, par nature, modélisent un double processus stochastique : l'émission des observations et la transition entre états. Chaque image de mot est transformée en une séquence de vecteurs de caractéristiques qui seront les observations à passer au MMC. Ainsi le problème de reconnaissance revient à un problème d'analyse de chaînes d'observations et le MMC modélise le mot en affectant les états aux caractères ou pseudo-caractères. Les probabilités de transitions sont calculées pour modéliser les probabilités de passage d'un caractère à un autre dans le mot.

Donc, il y a deux principaux avantages de l'utilisation des MMCs pour la reconnaissance de l'écriture arabe. Premièrement, la densité des fonctions de probabilités d'émissions des MMCs permet de modéliser les variations des formes de caractères arabes. Deuxièmement, la procédure de décodage résout en même temps la segmentation et la reconnaissance des mots.

Nos travaux présentent différentes améliorations par rapport aux systèmes basés sur les MMCs pour la reconnaissance de l'écriture arabe, comme celui présenté en [11]. Principalement, nous analysons l'apport de l'introduction des modèles de longueurs dans les topologies des MMCs afin de modéliser les variations des caractères arabes¹. Des systèmes à modèles de longueur ont déjà été testé pour la reconnaissance du texte latin à basse résolution, voir par exemple [5]. Une autre amélioration visée est la reconnaissance à vocabulaire ouvert.

Ce papier est organisé comme suit. Dans la section 2, nous présentons un aperçu général sur les caractéristiques de l'écriture arabe. Dans la section 3, nous décrivons les détails du système proposé. La section 4 présente la base de données utilisée pour l'évaluation. Pour conclure, nous discutons les résultats obtenus dans la section 5.

2. Caractéristiques générales de l'écriture arabe

Comme illustré sur la figure 1, l'arabe s'écrit de la droite vers la gauche. L'écriture est semi-cursive soit sous forme imprimée ou manuscrite. Chaque caractère a un point de connexion droite et/ou gauche lié sur la *ligne de base*. Le concept de majuscule et minuscule en écriture arabe n'existe pas. L'alphabet arabe est plus riche que son équivalent latin, Il contient 28 lettres dont la plupart changent de formes selon leur apparition au début, au milieu ou à la fin du mot.

¹ Une autre différence en se comparant à [7] est l'introduction des estimateurs continus pour les probabilités d'émission plutôt que des estimateurs discrets.

Par exemple : la lettre « خ (Xaa) » a quatre formes d'apparitions : isolée « خ » comme dans « صرخ (crier) », au début « خ » comme dans « خبرة (expérience) », au milieu « خ » comme dans « مخبر (laboratoire) », à la fin « خ » comme dans « طبخ (cuisiner) ». La table 1 présente les 28 lettres arabes avec leurs différentes formes d'apparitions dans un mot. Les lettres qui ont juste deux formes d'apparitions ne peuvent pas être liées à la lettre suivante, leur forme de début est simplement leur forme isolée et leur forme de milieu est exactement celle de fin.

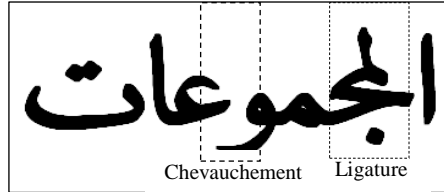


FIG. 1 - Exemple de mot arabe présentant une ligature verticale et un chevauchement

Un mot arabe peut être composé d'une ou plusieurs composantes connexes (pseudo-mot). Les caractères d'une même composante connexe peuvent être ligaturés horizontalement ou verticalement. Dans certaines fontes, on peut aller jusqu'à quatre caractères ligaturés verticalement. Ceci rend la segmentation à priori en caractères quasi-impossible. La figure 1 montre une ligature verticale de trois caractères Laam, Miim et Jiim.

Finalement, les chevauchements verticaux peuvent se produire par l'intersection des composantes connexes (pseudo-mots, voir figure 1) ou des mots pour quelques combinaisons de caractères. Les chevauchements et ligatures dépendent de la fonte utilisée. Plus de détails sur les caractéristiques de l'écriture arabe sont disponibles dans [18].

Etiquettes des lettres	isolé	début	milieu	fin	Etiquettes des lettres	isolé	début	milieu	fin
Alif		ا		آ	Daad	ض	ضد	ضد	ض
Baa	ب	ب	ب	ب	Thaaa	ط	ط	ط	ط
Taaa	ت	ت	ت	ت	Taa	ظ	ظ	ظ	ظ
Thaa	ث	ث	ث	ث	Ayn	ع	ع	ع	ع
Jiim	ج	ج	ج	ج	Ghayn	غ	غ	غ	غ
Haaa	ح	ح	ح	ح	Faa	ف	ف	ف	ف
Xaa	خ	خ	خ	خ	Gaaf	ق	ق	ق	ق
Daal		د		د	Kaaf	ك	ك	ك	ك
Thaal		ذ		ذ	Laam	ل	ل	ل	ل
Raa		ر		ر	Miim	م	م	م	م
Zaay		ز		ز	Nuun	ن	ن	ن	ن
Siin	س	س	س	س	Haa	ه	ه	ه	ه
Shiin	ش	ش	ش	ش	Waaw	و	و	و	و
Saad	ص	ص	ص	ص	Yaa	ي	ي	ي	ي

TAB. 1 - Les lettres arabes

3. Description du Système proposé

Le système proposé est basé sur la boîte à outils HTK. Comme illustré dans la figure 2, le système fonctionne en deux phases : l'apprentissage et la reconnaissance. Pour les deux phases, nous extrayons les mêmes caractéristiques sur les images de mots arabe à basse résolution.

3.1. Prétraitement et caractéristiques extraites

Chaque image du mot est transformée en une séquence de vecteurs de caractéristiques calculée à partir d'une fenêtre glissante de taille N pixels. Dans notre cas, la fenêtre d'analyse est de taille uniforme et se déplace d'un pixel de droite vers la gauche. Nous avons effectué plusieurs tests afin de déterminer la taille optimale de la fenêtre glissante. La largeur $N=14$ optimise le taux de reconnaissance sur notre base de test. Aucune procédure de segmentation en caractère n'est faite à priori et chaque mot est transformé en une matrice dont le nombre des lignes correspond au nombre de fenêtres et le nombre des colonnes au nombre de caractéristiques.

Les caractéristiques extraites sont divisées en deux ensembles. Le premier est calculé directement sur la fenêtre d'analyse, il s'agit du :

- nombre N_1 de composantes connexes noires (CCNs),
- nombre N_2 de composantes connexes blanches (CCBs),
- rapport N_1/N_2 des composantes connexes noires et blanches,
- l'ordonnée de la plus petite composante connexe noire divisée par la hauteur de la fenêtre,
- périmètre de toutes les composantes dans la fenêtre divisée par celui de la fenêtre,
- compacité $((\text{PerimetreCCNs})^2 / (4\pi \text{SurfaceCCNs}))$,
- centre de gravité de la fenêtre, du demi droite, du demi gauche, du 1^{er}, 2^{ème} et 3^{ème} tiers de la fenêtre divisée verticalement.

$$G_x = \frac{\sum_{i=1}^n x_i}{n \times \text{Largeur}} \quad G_y = \frac{\sum_{i=1}^n y_i}{n \times \text{Hauteur}}$$

Le deuxième ensemble de caractéristiques consiste à calculer les valeurs des deux projections verticales et horizontales en normalisant la hauteur de la fenêtre à 20 pixels. L'ensemble des caractéristiques utilisées est 53 coefficients. Le tout est implémenté en Java.

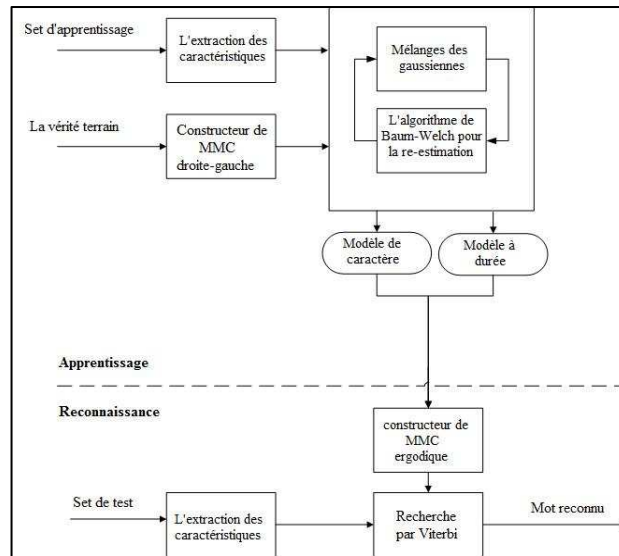


FIG. 2 - Mode de fonctionnement du système de reconnaissance basé sur HTK.

3.2. MMC et HTK

Nous utilisons la boîte à outils HTK [24] pour l'évaluation de notre approche. Cette boîte à outils dédiée aux MMCs est principalement utilisée pour la reconnaissance de la parole. HTK se compose d'un ensemble de modules exécutés en ligne de commande. L'utilisation de HTK comprend trois phases : la préparation des données, l'apprentissage et la reconnaissance.

La phase de préparation des données vise à extraire les séquences de vecteurs de caractéristiques pour chaque mot.

Durant la phase d'apprentissage, les fichiers sont d'abord utilisés pour l'initialisation des paramètres des sous-modèles MMCs en utilisant l'outil HCompV. Dans notre approche, les sous-modèles représentent une lettre ou une combinaison des lettres et sont combinés de droite à gauche selon le contenu de l'image représentant un mot de l'apprentissage. La procédure d'apprentissage utilise une topologie simple avec un état par modèle comme le montre la figure 3-B. Une fois les paramètres initialisés, un entraînement itératif des paramètres du modèle est effectué avec l'algorithme de Baum-Welch (outil HTK HERest). La procédure d'apprentissage se déroule en fait en deux étapes. Dans la première étape, une procédure est appelée pour augmenter le nombre de gaussiennes utilisées dans les fonctions d'estimation. Dans la deuxième étape, l'algorithme de Baum-Welch est utilisé pour estimer les paramètres des gaussiennes.

A la fin de cette phase d'apprentissage, la longueur de chaque sous-modèle est calculée selon une procédure d'alignement forcé appliquée sur les données d'entraînement. Ces longueurs sont ensuite utilisées pour modifier la topologie de chaque sous-modèle (voir section 3.3).

Durant la phase de reconnaissance, un MMC ergodique est créé à partir de tous les sous-modèles. Dans ce MMC, toutes les transitions entre sous-modèle sont autorisées ce qui lui permet de reconnaître n'importe quel mot dans un vocabulaire ouvert. L'inconvénient est que le système peut proposer des hypothèses de mots qui ne sont pas valides. Cependant, les résultats obtenus montrent que le recours aux modèles de longueurs réduit de façon significative les erreurs du système. L'utilisation d'une topologie ergodique offre également d'autres avantages comme une utilisation moins intensive de la mémoire et du processeur par rapport aux autres approches globales ou utilisant des grammaires stochastiques. La reconnaissance se fait en appliquant l'algorithme de Viterbi qui cherche la meilleure séquence des sous-modèles dans le MMC (outil HVite de HTK). Les performances du système sont évaluées en termes de taux de reconnaissance des mots sur un set indépendant de test (outil HResult de HTK).

3.3. Topologies des sous-modèles

En phase de reconnaissance, le modèle proposé est ergodique et un sous-modèle représente un ou plusieurs caractères (cas de ligature). Le tableau 2 montre les 60 sous-modèles utilisés. La procédure de sélection des sous-modèles dérive d'un regroupement des caractères de formes similaires. Notre hypothèse est que l'estimateur des probabilités d'émission basées sur le mélange des gaussiennes offrira la flexibilité nécessaire pour modéliser les parties communes et les variations intérieures de chaque catégorie de sous-modèle.

Selon la terminologie introduite pour la reconnaissance de la parole [20], nos modèles sont censés être indépendants du contexte, c à d indépendant des caractères précédents et suivants.

Etiquette du sous-modèle	Caractères	Etiquette du sous-modèle	Caractères	Etiquette du sous-modèle	Caractères	Etiquette du sous-modèle	Caractères
Baa_B	ب	TaaaClosed_E	ة	Saad_I	ص	HaaChadda_E	ه
Taaa_M	ت	YaaChadda_I	ي	Xaa_B	خ	HamzaAboveAlif	أ
Raa_I	ر	YaaChadda_M	م	Siin_I	س	NuunChadda_E	ن
Ghayn_B	غ	Hamza	ء	Siin_B	س	Shiin_M	ش
Alif_E	ا	Laam_I	ل	Daal_I	د	Taa_I	ظ
Nuun_B	ن	Laam_M	ل	Haa_I	ه	Baa_E	ب
Taaa_E	ت	LaamAlif_I	لا	Haa_B	ه	Shiin_I	ش
Miim_B	م	Thaa_M	ث	Thaal_I	ذ	Saad_B	ص
Miim_I	م	Gaaf_I	ق	Xaa_I	خ	Ayn_M	ع
Thaaa_I	ط	Gaaf_B	ق	AlifBroken_E	ي	Ayn_B	ع
Waaaw	و	Jiim_M	ج	Haaa_I	ح	Ghayn_M	غ
Nuun_I	ن	Haaa_B	ح	Zaay_I	ز	Daad_I	ض
Yaa_M	ي	Thaa_E	ث	Kaaf_I	ك	Daad_M	ض
Yaa_I	ي	HamzaUnderAlif_I	ا	Kaaf_B	ك	Jiim_I	ج
Ghayn_I	غ	Ghayn_E	غ	Ayn_E	ع	Faa_M	ف

TAB. 2 - les étiquettes des sous-modèles et les caractères correspondants

Nous avons testés trois topologies de sous-modèles :

Catégorie 1. Modèles à longueur égale: la même topologie est utilisée pour tous les sous-modèles. Par exemple, chaque sous-modèle contient 1 état (Figure 3-B) ou 3 états (Figure 3-C).

Catégorie 2. Modèle de longueur dépendant de la largeur à priori de la forme: le nombre d'états dépend de la largeur du caractère associé: par exemple pour le caractère *Alif_E*, on affecte un état parce qu'il est étroit alors que pour le caractère *Saad_I*, plus large, on lui affecte 5 états.

Catégorie 3. Modèle à longueur minimale inférée: durant la phase d'apprentissage, nous affectons un seul état par modèle et en phase de reconnaissance, nous répétons cet état un nombre de fois proportionnel à la longueur inférée par alignement forcé sur les données d'entraînement. Ce modèle de longueur est dit *minimal* car nous forçons le processus de décodage à affecter au modèle *i* un minimum de D_i vecteurs de caractéristiques.

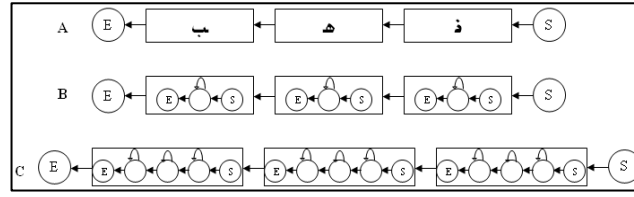


FIG. 3 - Exemple de topologie : Modèle de longueur à égalité (B : 1-état) (C : 3-États)

4. Evaluation du système

Pour l'évaluation de notre système, nous avons utilisé une base de données semblable à celle présentées dans [6]. Elle est composée de 20630 images de mots arabes imprimés à basse résolution. Ces images sont synthétiques, générées en utilisant une taille de 24 *points* et avec la fonte *Times*. Elles sont normalisées à une hauteur de 56 pixels et encodées en format PNG. La résolution des images est de 72 pixels/pouce.

Une génération synthétique des données permet de générer une très grande base d'images où les mots sont déjà isolés. Cette grande quantité de données est très bénéfique pour l'évaluation et permet de converger vers des valeurs précises des probabilités d'émissions en phase d'apprentissage. Il faut néanmoins noter que les données ainsi générées ne contiennent pas les artefacts normalement présents lors de numérisation par scanner. Les résultats obtenus seront donc optimistes par rapport à un contexte de numérisation par scanner. Ils seront néanmoins réalistes dans un contexte d'applications « screen-based OCR » et les comparaisons entre types de modèles de longueur resteront pertinentes.

La base de donnée est divisée en deux ensembles disjoints : le premier contient 19630 images utilisées pour l'apprentissage et le deuxième contient 1000 images utilisées pour le test.

5. Résultats expérimentaux

Les résultats sont résumés dans le tableau 3. Ils sont obtenus avec les mêmes paramètres du système : même complexité des modèles (64 gaussiennes par états), même nombre d'itération et l'application des mêmes algorithmes. Les catégories listées dans la table correspondent aux trois catégories de modèle de longueur décrites en section 3.3.

Catégorie	Modèle d'apprentissage	Modèle de test	% (mot)
1	1-état	1- état	7.0
1	3- états	3- états	70.4
1	5- états	5- états	92.9
1	6- états	6- états	93.1
2	[1,3]- états	[1,3]- états	18.1
2	[3,5]- états	[3,5]- états	85.9
2	[1,3,5]- états	[1,3,5]- états	78.5
2	[3,5,6]- états	[3,5,6]- états	88.6
3	1- état	longueur minimale calculé manuellement	81.5
3	1- état	longueur minimale $d = \mu_d/2$	91.9

TAB. 3 - Résultats du système

Les résultats de la catégorie 1 montrent que les performances s'améliorent avec le nombre d'états par modèle. Avec 6 états par sous-modèle, nous obtenons le meilleur résultat avec un taux de 93.1%. Ces bonnes performances obtenues sont néanmoins au détriment du temps cpu et de la mémoire utilisée. Nous observons que l'augmentation de performance de 1 à 6 états est très significative. Une analyse des erreurs a révélé un nombre important d'erreurs d'insertions et de suppression pour la configuration à 1 état. Le modèle ergodique avec un seul état n'est effectivement pas performant car il permet des sauts d'un sous-modèle à un autre pratiquement de vecteur en vecteur. Pour un état, le taux de reconnaissance par mot est très faible (7%) mais

le taux de reconnaissance par caractères est élevé (99.4%). Cela montre que la majorité des lettres sont reconnues alors que certaines sont insérées ou supprimées dans la plupart des mots.

Dans la catégorie 2, le nombre d'états dépend de la largeur de chaque caractère. Pour la configuration des états [1,3,5], nous avons affecté 1 seul état pour les lettres courtes, 3 états pour les lettres de largeur moyenne et 5 états pour les lettres larges. L'affectation d'une lettre à une catégorie (courte, moyenne ou large) est manuelle en utilisant nos connaissances à priori sur la forme des lettres arabe. L'idée est d'affecter plus d'états pour les lettres larges afin de mieux les modéliser et moins d'états pour les courtes. Les résultats montrent que les performances ne sont pas loin de celles de la première catégorie avec une réduction du temps cpu et mémoire.

La 3^{ème} catégorie présente deux expériences. Dans la première, les valeurs de longueur minimale sont récupérées manuellement en utilisant nos connaissances à priori sur la forme des caractères arabes. Alors que pour la deuxième, les valeurs de longueur minimale sont inférées automatiquement par un alignement forcé Viterbi pendant la phase d'apprentissage. Pendant l'alignement, on accumule dans un histogramme le nombre de vecteurs consécutifs affectés au même sous-modèle. La durée moyenne dans un sous-modèle est ensuite calculée sur base de l'histogramme. La longueur minimale est ensuite fixée à la moitié de la valeur moyenne, $D_i = \mu_i / 2$ [20]. Selon les résultats observés, nous pouvons conclure que :

- Les modèles de longueur inférés automatiquement donnent de meilleurs résultats que ceux basés sur nos connaissances à priori des longueurs. Ce résultat montre que les longueurs mesurées automatiquement sont plus précises ou en tout cas correspondent mieux au processus de décodage en phase de reconnaissance.
- En comparant les résultats obtenus avec un seul état par modèle (7%) et avec les modèles de longueur minimale (91.9%), nous observons tous les avantages de l'introduction des longueurs minimales. La mémoire utilisée reste plus ou moins la même (60 octets de plus que la reconnaissance avec un seul état par modèle), puisqu'on utilise le même set de paramètres qui est simplement référencé en mémoire plusieurs fois à travers les répétitions. Le décodage se fait également plus rapidement puisque le taux de branchement entre états est réduit dans les sous-modèles.

L'observation la plus intéressante est que l'on peut avoir des performances similaires à celles d'un système complexe (catégorie 1, 6 états par sous-modèle) en utilisant moins de ressource mémoire et temps de calcul juste en introduisant la longueur minimale dans les sous-modèles (catégorie 3, 1 état).

Un problème qui persiste dans le système présenté est la reconnaissance de mots qui ont deux mêmes lettres consécutives. Il est en effet équivalent pour le système de rester plus longtemps dans le même sous-modèle que de passer dans deux sous-modèles successifs. Ce point sera traité prochainement en utilisant des modèles de longueur minimale et maximale.

6. Conclusion

Nous avons présenté un système de reconnaissance de l'écriture arabe imprimée à basse résolution et à vocabulaire ouvert basé sur les MMCs. Tel que cela a déjà été montré dans de nombreux travaux, le principal avantage des MMCs est la résolution simultanée de la segmentation en caractère et de la reconnaissance, ce qui est particulièrement important dans le cas de l'écriture arabe imprimée qui est semi-cursive. La principale nouveauté rapportée dans ce travail est le développement d'un système de reconnaissance de l'écriture arabe à basse résolution avec l'introduction de modèles de longueurs minimale qui permettent d'améliorer de façon significative les performances du système tout en gardant une architecture légère en termes de mémoire et temps de calcul. Nous envisageons prochainement étendre le système pour reconnaître des mots imprimés omni-fontes en intégrant les concepts linguistiques arabe dans le moteur markovien (approche marko-linguistique). Nous envisageons également introduire des modèles de longueur maximale qui permettraient de réduire le nombre d'erreurs sur les mots comprenant des doubles caractères.

Bibliographie

1. A. Amin, H.B. Al-Sadoun, S. Fisher: HandPrinted Arabic Character Recognition system

- using an artificial network. *Pattern Recognition*, 29(4) :663-675, 1996.
2. A. Amin, S. Al-Fedaghi, Machine recognition of printed Arabic text utilizing natural language morphology, *Int. Journal Man-Machine Studies*, IJMMS'91, 35 :769-788, 1991.
3. A. Schlapbach, H. Bunke, Using HMM-based recognizers for writer identification and verification, *Proc. 9th Int. Work. on Frontiers in Handwriting Recognition*, 167-172, 2004.
4. B. Al-Badr, S.A. Mahmoud, Survey and bibliography of Arabic Optical Text Recognition, *Signal processing*, 41 :49-77, 1995.
5. F. Einsele, R. Ingold, J. Hennebert, A Language-Independent, Open-Vocabulary System Based on HMMs for Recognition of Ultra Low Resolution Words, In *proc. of 23rd Annual ACM Symposium on Applied Computing*, Fortaleza (Brasil), March 16 - 21 2008.
6. F. Slimane, R. Ingold, S. Kanoun, M. A. Alimi and J. Hennebert, A New Arabic Printed Text Image Database and Evaluation Protocols, *ICDAR'09*, pp. 946-950, 2009.
7. H. Miled, C. Olivier, M. Cheriet, Y. Lecourtier, Coupling observation/letter for a Markovian modelisation applied to the recognition of arabic handwriting, *IEEE Proc. 4th Int. conference on document analysis and recognition*, Ulm, Germany, 580-583, 1997.
8. H. Miled, M. Cheriet, C. Olivier, Y. Lecourtier, Modélisation markovienne de l'écriture arabe manuscrite: une approche analytique, *Proc. 1er Colloque international francophone sur l'écrit et le document*, Québec, Canada, 50-59, 1998.
9. H. Miled, N. Ben Amara, Planar Markov Modeling for Arabic Writing Recognition: Advancement State, *Proc. of ICDAR'01*, 69-73, 2001.
10. I. Ben Cheikh, A. Belaïd, and A. Kacem, A Novel Approach for the Recognition of a Wide Arabic Handwritten Word Lexicon, *ICPR'2008*, 2008.
11. M. S. Khorsheed, Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK), *Pattern Recognition Letters*, 28(12) :1563-1571, 2007.
12. M. S. Khorsheed, Off-line arabic character recognition- a review, *Pattern Anal, Appl.* 5, 31-45, 2002.
13. M. S. Khorsheed, Recognising handwritten Arabic manuscripts using a single hidden Markov model, *Pattern Recognition Letters*, 24(14) :2235-2242, October 2003.
14. M. S. Khorsheed, Clocksin, W.F., Multi-font arabic word recognition using spectral features, *15th Int. Conf. on Pattern Recognition*, Barcelona, Spain, 4 : 543-546, 2000.
15. M.C. Fehri, Reconnaissance de textes arabes mutifonte à l'aide d'une approche hybride neuro-markoviennes. Thèse de doctorat, Université des sciences, des techniques et de médecine de Tunis II, Tunisie, 1999.
16. M.C. Fehri, M. Ben Ahmed, Off-line arabic handwriting recognition, *Computational engineering in systems applications*, Nabeul-Hammamet, Tunisie, 1-3, 1998.
17. N. Ben Amara : Utilisation des modèles de Markov cachés planaires en reconnaissance de l'écriture arabe imprimée. Thèse de doctorat, Université des sciences, des techniques et de médecine de Tunis II, Tunisie, 1999.
18. N. Ben Amara, A. Belaïd and N. Ellouze, Utilisation des modèles markoviens en reconnaissance de l'écriture arabe :État de l'art, *CIFED*, 2000.
19. Rabiner, L., A tutorial on HMM and selected applications in speech recognition. *Proc. IEEE* 77(2) :257-286, 1989.
20. Rabiner, L., Juang, B., *Fundamentals of Speech Recognition*, Prentice Hall, 1993.
21. S. Kanoun, A. M. Alimi, Y. Lecourtier, Affixal Approach for Arabic Decomposable Vocabulary Recognition: A Validation on Printed Word in Only One Font, *ICDAR'2005*, Seoul, Korea, 1025-1029, 2005.
22. T. Sari, M. Sellami, MOOrpho-LEXical analysis for correcting OCR-generated arabic words (MOLEX), *Int. Work. on Frontiers in Handwriting Recognition*, 461-466, 2002.
23. U-V. Marti and H. Bunke, Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system, *Journal of Pattern Recognition and Art. Intelligence* 15, 65-90, 2001.
24. Young, S., Evermann, G., Kershaw, D., Moore, D., Odell, J., Ollason, D., Valtchev, V., Woodland, P., *The HTK Book*, Cambridge University Engineering Dept., 2001.